

**Marcelo Augusto Leal Alves**  
(Engenheiro Mecânico, EPUSP, 1993)

**UTILIZAÇÃO DE FREQUÊNCIAS NATURAIS E REDES NEURAIAS  
ARTIFICIAIS PARA DETECÇÃO DE DEFEITOS EM ESTRUTURAS**

Dissertação de mestrado apresentada à Escola  
Politécnica da Universidade de São Paulo para a  
obtenção do título de mestre em Engenharia

**São Paulo**

**1997**

OK

**Marcelo Augusto Leal Alves**  
(Engenheiro Mecânico, EPUSP, 1993)

**UTILIZAÇÃO DE FREQUÊNCIAS NATURAIS E REDES NEURAIAS  
ARTIFICIAIS PARA DETECÇÃO DE DEFEITOS EM ESTRUTURAS**

Dissertação de mestrado apresentada à Escola  
Politécnica da Universidade de São Paulo para a  
obtenção do título de mestre em Engenharia

Área de concentração:  
Engenharia Mecânica

Orientador:  
Prof. Dr. Paulo Carlos Kaminski

**São Paulo**

**1997**

DEDALUS - Acervo - EPMN



31600010105

**À minha família**

## **AGRADECIMENTOS**

Ao CNPq, pelo apoio nos estágios iniciais desta pesquisa.

Ao Prof. Dr. Paulo C. Kaminski, pela orientação e incentivo desde o curso de graduação.

## ÍNDICE

<b>TÓPICO</b>	<b>pag.</b>
NOTAÇÃO	
RESUMO	1
ABSTRACT	2
1. INTRODUÇÃO	3
2. REDES NEURAIIS ARTIFICIAIS	7
2.1. Introdução	7
2.2. Redes neurais biológicas	8
2.3. Unidades de processamento	9
2.4. Redes neurais	13
2.4.1 O problema da representação	15
2.4.1.1 Representação com uma camada de neurônios	15
2.4.1.2 Representação com duas ou mais camadas de neurônios	16
2.5. Processos de treinamento	17
2.5.1 Regra do delta	21
2.6. Algoritmo de retro-propagação do erro	24
2.7. Diretrizes para a aplicação de RNAs	33
2.7.1 Adequação da rede ao problema a ser resolvido	33
2.7.2 Geração do conjunto de dados de treinamento	34
3. REVISÃO BIBLIOGRÁFICA	36
3.1 Redes neurais artificiais e suas aplicações.	36
3.2 Redes neurais artificiais aplicadas à detecção de defeitos em estruturas	40
3.3 Ensaios não destrutivos baseados em modificações na resposta dinâmica de estruturas.	41
4. IDENTIFICAÇÃO DE DEFEITOS EM ESTRUTURAS	47
4.1 Ensaios não-destrutivos baseados na análise dinâmica de estruturas	48
4.2 ENDS empregando variações na resposta dinâmica e RNAs	52
4.2.1 Pré-processamento das entradas.	54

5. EXEMPLOS DE APLICAÇÃO	56
5.1 Exemplo - 1: Viga livre-livre	57
5.1.1 Modificações na resposta dinâmica	58
5.1.2 Rede neural adotada.	77
5.1.2.1 Conjunto de dados de treinamento e teste	78
5.1.2.2 Determinação da RNA mais adequada	79
5.1.3 Resultados do exemplo - 1	81
5.2 Exemplo - 2: Torre	83
5.2.1 Rede neural adotada.	84
5.2.2 Resultados do exemplo - 2	86
6. CONCLUSÕES E TRABALHOS FUTUROS	89
7. BIBLIOGRAFIA	92
APÊNDICE - I	96
APÊNDICE - II	111

## ÍNDICE PARA AS FIGURAS

FIGURA	pag.
2.2.1 Neurônio biológico e seu componentes principais.	8
2.3.1 Neurônio de McCulloch-Pitts	9
2.3.2 Funções de transferência	11
2.4.1 Rede neural multicamada com duas camadas ocultas	13
2.4.1.1 Neurônio com duas entradas	15
2.4.1.2 Separação em duas categorias	15
2.4.1.3 (a) Classes linearmente separáveis; (b) Classes não separáveis linearmente	16
2.4.1.4 Rede neural com duas camadas	17
2.4.1.5 Superfícies convexas. (a) e (b) fechadas; (c) aberta	17
2.4.1.6 (a) Classes separáveis por curva convexa; (b) classes não-separáveis por curva convexa	18
2.4.1.7 Rede neural com três camadas	18
2.4.1.8 União de duas superfícies de representação convexas resultando numa não-convexa	19
2.6.2 Fluxograma para a retropropagação do erro	28
5.1.1 Viga livre-livre com entalhe	57
5.2.1 Exemplo - 2: Torre aperticada	83

## NOTAÇÃO

$a$  = entrada líquida de um neurônio genérico

$a_i$  = entrada líquida do neurônio  $i$

$E$  = função erro

$k$  = índice para camada da rede neural

$K$  = número total de camadas de uma rede neural artificial

$m$  = índice para elemento do conjunto de treinamento

$n$  = índice para iteração

$nfr_i$  = diferença entre as frequências naturais da estrutura com e sem defeitos para o modo  $i$ .

$nfm_i$  = normalização para o parâmetro  $nfr_i$ .

$t$  = índice para transposto

$t$  = saída desejada de um neurônio genérico

$t_i$  = saída desejada do neurônio  $i$

$x$  = entrada líquida de um neurônio genérico

$x_j$  = entrada  $j$  de um neurônio

$y$  = saída de um neurônio genérico

$y_i$  = saída do neurônio  $i$

$z_i$  = saída do neurônio  $i$  (notação específica para dedução do algoritmo de retro-propagação)

$\alpha$  = coeficiente do termo de momento

$\delta_i$  = erro no neurônio  $i$

$\Delta w_{ij}$  = atualização de um peso  $w_{ij}$

$\eta$  = taxa de aprendizado

$\theta_i$  = limiar da função de transferência tipo degrau no neurônio  $i$

$\omega_i$  = frequência natural referente ao  $i$ -ésimo modo de vibrar da estrutura com defeitos

$\omega_{oi}$  = frequência natural referente ao  $i$ -ésimo modo de vibrar da estrutura sem defeitos

### Vetores e Matrizes

$D$  = matriz de amortecimento da estrutura

$K$  = matriz de rigidez da estrutura

$M$  = matriz de massa da estrutura

$H$  = matriz hessiana

$t_m$  = m-ésimo vetor de saídas desejadas do conjunto de treinamento

$u$  = vetor de deslocamentos

$w$  = vetor de pesos

$w_i$  = vetor de pesos das conexões de entrada de um neurônio  $i$

$w_{(n)}$  = vetor de pesos da iteração  $n$

$x$  = vetor de entradas

$x_m$  = m-ésimo vetor de entradas do conjunto de treinamento

$y_m$  = vetor de saídas referentes as entradas  $x_m$

$\Delta w$  = atualização de um vetor de pesos  $w$

$\Phi_i$  = i-ésimo modo de vibrar

## RESUMO

É proposto o uso de frequências naturais e redes neurais artificiais para a detecção de defeitos em estruturas. As frequências naturais de uma estrutura são afetadas pela presença de defeitos. Entretanto, a detecção e localização direta destes últimos a partir apenas das variações das frequências só foi conseguida para modelos simples de estruturas. O uso das redes neurais artificiais é indicado como meio para obter-se a localização e avaliação da extensão destes defeitos tendo apenas algumas frequências naturais como informação.

Foi feita uma revisão bibliográfica que cobre os tópicos: Detecção de defeitos baseando-se em modificações na resposta dinâmica da estrutura, redes neurais artificiais e suas aplicações e ainda mais especificamente, redes neurais artificiais aplicadas à detecção de defeitos. Também é feita uma introdução às redes neurais.

São apresentados dois exemplos de aplicação que cobrem diferentes aspectos da metodologia proposta. São vistas duas estruturas bastante distintas bem como maneiras diferentes de modelagem dos defeitos a serem detectados.

Os resultados mostram a viabilidade da metodologia proposta e ainda fornecem diretrizes para desenvolvimentos futuros.

## **ABSTRACT**

The use of natural frequencies and artificial neural networks to damage detection in structures is proposed. The natural frequencies of a structure are modified by the presence of damages. However, the detection and location of damages only from the changes on natural frequencies, it was only achieved in simple models for the structures. The use of neural networks is aimed as a way to obtain the location and extent of the damages having only a few natural frequencies as information.

A bibliographical review was done and it covers the following topics: Damage detection using the modification in the dynamic response of a structure, artificial neural networks and their applications, and more specifically, artificial neural networks applied do damage detection. It was also done an introduction to the artificial neural networks.

Two examples are presented and they cover many aspects of the proposed methodology. Two very different structures are viewed as well as distinct ways to model the damages to be detected.

The results show the feasibility of the proposed methodology and they also provide guidelines to future developments.

## 1. INTRODUÇÃO

Estruturas e componentes mecânicos acumulam falhas durante suas vidas úteis. A detecção destas falhas ou monitoração das mesmas é essencial do ponto de vista da eficiência e segurança.

Por afetarem basicamente a rigidez das estruturas, as falhas provocam mudanças na resposta dinâmica das mesmas. Por resposta dinâmica entende-se o conjunto de frequências naturais e modos de vibrar. Entendendo a maneira pela qual estas falhas modificam a resposta dinâmica, seria possível detectá-las a partir da comparação das respostas dinâmicas da estrutura sem defeitos com aquelas de uma estrutura defeituosa.

Entretanto, por razões práticas não é possível obter por completo a resposta dinâmica precisa de uma estrutura com os procedimentos de ensaio disponíveis. A resposta completa só seria obtida caso fossem feitas medições em todos os graus de liberdade de todos os pontos da estrutura. Caso se restrinja a comparação da resposta dinâmica a uma comparação entre frequências naturais, já se aproxima de um procedimento mais prático de ensaio, uma vez que estas são obtidas com razoável precisão sem que seja necessária uma aparelhagem complexa de ensaio.

Uma maneira de fazer a comparação entre as respostas dinâmicas é através de redes neurais artificiais (RNAs). Surgidas na década de 40 como um modelo para o funcionamento do cérebro humano, as RNAs possuem como característica principal a capacidade de *aprendizado*. Neste caso, uma rede neural artificial é *treinada* para aprender a relação que existe entre as variações na resposta dinâmica de uma estrutura e a presença de defeitos, bem como a localização e extensão dos mesmos.

Uma rede neural artificial é composta por uma série de *unidades de processamento* ou *neurônios artificiais* dispostos em *camadas* e ligados entre si. A cada ligação entre um neurônio  $i$  e outro  $j$  é associado um peso,  $w_{ij}$ . Cada neurônio possui uma função de transferência que calcula a saída do mesmo, que é única, a partir das entradas ponderadas pelos pesos.

As camadas de uma RNA podem ser divididas em três grupos. O primeiro contém os neurônios que recebem as entradas das redes e as distribuem para os neurônios das camadas

seguintes. Este primeiro grupo constitui o que se denomina de *camada de entrada*, cujos neurônios não realizam qualquer tipo de processamento, isto é, acabam não possuindo função de transferência nem pesos associados as suas conexões de entrada. Existem ainda a *camada de saída* onde são calculadas as saídas das redes neurais. Entre estas duas camadas, estão as *camadas ocultas*, responsáveis por maior parte do processamento que ocorre numa RNA. A camada de entrada não é computada no total de camadas de uma RNA justamente pelas características peculiares de seus neurônios. Logo, uma RNA com três camadas de neurônios possui, além da camada de saída, duas camadas ocultas.

O processo de *treinamento* de uma rede neural artificial consiste em determinar os valores dos pesos  $w_{ij}$ , de forma que a RNA consiga implementar a relação correta entre os dados de entrada e saída. Existem basicamente dois tipos de processos de treinamento. O primeiro é denominado *treinamento supervisionado* e consiste em apresentar à RNA um conjunto de dados de entrada característicos do problema e seus respectivos valores esperados de saída. A partir dos erros entre saídas esperadas e aquelas calculadas pela rede é possível ir ajustando o valor dos pesos  $w_{ij}$  de maneira a minimizar estes erros.

O outro tipo de processo de treinamento é denominado *não supervisionado*. Neste tipo de treinamento, não são apresentadas a RNA as saídas esperadas. No treinamento não supervisionado, os pesos são ajustados para que entradas similares produzam saídas similares. O algoritmo de treinamento extrai regularidades estatísticas do conjunto de dados de treinamento. O tipo de treinamento mais empregado na prática é o supervisionado, sendo que o algoritmo mais difundido é o de *retro-propagação do erro*.

O uso de redes neurais faz-se necessário ante a necessidade de resolução de um *problema inverso*. A obtenção da resposta dinâmica de uma estrutura dadas as modificações em suas características de massa e rigidez já foi extensivamente estudada [21]. Este problema é conhecido como *problema direto*, isto é, dada a estrutura e os esforços que agem sobre a mesma obtém-se a resposta da estrutura. Já para o problema inverso pode-se fazer uma distinção. Num primeiro caso dispõe-se da estrutura e da resposta da mesma e o que se procura são as entradas que provocaram a resposta disponível. Numa segunda possibilidade são disponíveis as entradas e a

resposta da estrutura às mesmas e com as relações entre estes dois parâmetros ajusta-se um modelo para a estrutura. Este segundo caso é conhecido como problema de *identificação*.

A detecção de defeitos em estruturas com base nas alterações na resposta dinâmica é um caso especial de um problema de identificação. Vários autores propõem este tipo de ensaio não-destrutivo, sendo que alguns deles também propõem o emprego de redes neurais.

Em comparação com outros métodos para a resolução do problema de identificação as redes neurais artificiais apresentam algumas vantagens. Entre elas pode-se destacar:

- Não há necessidade de um algoritmo complexo para a solução do problema.
- Outros métodos necessitam que a resposta seja medida com razoável precisão e envolvendo um número elevado de graus de liberdade, o que não é necessariamente requerido quando são usadas as RNAs.
- As redes neurais artificiais podem ser treinadas para compensar alguns erros oriundos dos processos de ensaio.

Entretanto, há desvantagens quando são usadas RNAs, entre elas destacam-se:

- O processo de treinamento de uma rede neural apresenta uma série de complicadores, entre eles há uma necessidade de muitos casos de treinamento, isto é, respostas conhecidas dadas as falhas. Este conjunto de treinamento também deve ser representativo do problema a ser resolvido uma vez que as RNAs não são indicadas para a extrapolação.
- Redes neurais artificiais não fornecem resultados com alta precisão, isto é, caso se esteja procurando resultados exatos, não são indicadas as redes neurais.
- Os fundamentos teóricos das RNAs ainda estão sendo estabelecidos, sendo que as mesmas ainda estão no início de seu desenvolvimento. Não se constituindo, portanto, numa ferramenta totalmente dominada.

Os objetivos deste trabalho são: Primeiramente apresentar os ensaios não-destrutivos baseados na análise da resposta dinâmica de uma estrutura. Em seguida será proposto o uso de redes neurais como método alternativo a resolução do problema inverso, referente a localização e

avaliação dos defeitos em estruturas, tendo como entradas para as redes neurais algumas das frequências naturais da estrutura sob ensaio.

Serão apresentadas duas simulações da metodologia proposta. Num primeiro caso é analisada uma viga com apenas um defeito presente. São variadas as posições e severidades deste defeito. O segundo exemplo estudado consiste num caso mais complexo, onde uma estrutura aproticada é analisada. Neste último caso há mais de um defeito presente sendo que os mesmos são representados como uma perda de rigidez de alguns dos componentes da estrutura. A rede neural indica se houve perda de rigidez num dos componentes da treliça e qual é o valor desta perda.

Como objetivo secundário destaca-se a apresentação das redes neurais artificiais, ainda pouco conhecidas e aplicadas em engenharia mecânica.

É feita ainda uma revisão bibliográfica que engloba três categorias de trabalhos:

- Redes neurais e suas aplicações.
- Redes neurais aplicadas à detecção de defeitos em estruturas
- Ensaio não destrutivo baseado em modificações na resposta dinâmica de estruturas.

Recomenda-se para aqueles que não conhecem as RNAs, a leitura do capítulo 2 deste trabalho - Redes Neurais Artificiais ou a referência [20]. Do contrário a leitura de outros capítulos ficará dificultada pois a terminologia e os conceitos a respeito das RNAs são essenciais para a compreensão deste trabalho.

## 2. REDES NEURAS ARTIFICIAIS

### 2.1. Apresentação

É cada vez maior o número de aplicações em engenharia mecânica de uma ferramenta computacional denominada rede neural artificial. Algumas descrições desta ferramenta e características atribuídas fazem com que as redes neurais artificiais apareçam como uma panacéia, solução para todos os problemas. É objetivo deste capítulo, resumir os aspectos principais a respeito do tema encontrado na bibliografia, de forma que o engenheiro mecânico possa obter algumas informações relevantes a respeito de redes neurais artificiais, além de apresentar o tema para aqueles que ainda o desconhecem.

A bibliografia encontrada não é dirigida a engenheiros mecânicos. O material disponível vai desde trabalhos desenvolvidos por pesquisadores da área de ciências cognitivas até da área de ciência da computação. Com isto, há uma profusão de notações distintas, jargões e até mesmo divergências a respeito de alguns conceitos.

As redes neurais artificiais (RNAs) surgiram como modelos para o cérebro humano, e tem como principal característica a capacidade de aprender a partir de um determinado conjunto de informações. A possibilidade de resolver problemas complexos sem a necessidade de um algoritmo igualmente complexo, ou ainda onde não é possível aplicar outra ferramenta de análise, é bastante atrativa e justamente por este motivo foi feito todo um investimento em pesquisa e aplicação de RNAs.

Alguns tópicos foram deixados à margem deste capítulo ou porque ainda estão na fronteira de desenvolvimento ou não são tão empregados na prática. Segue-se uma linha de desenvolvimento que levou a obtenção do algoritmo de retro-propagação de erros (back-propagation) que é o mais difundido.

## 2.2. Redes neurais biológicas

Está fora do escopo deste capítulo a descrição detalhada das redes neurais biológicas. Entretanto, as redes neurais artificiais foram inicialmente inspiradas nos sistemas biológicos, em particular no sistema nervoso humano, sendo que até parte da terminologia usada nas redes neurais artificiais provém do estudo dos sistemas biológicos.

A unidade básica de uma rede neural biológica é a *célula nervosa*, ou *neurônio*. Os componentes principais de um neurônio são os *dendritos*, o *soma* ou *núcleo* e o *axônio*. As ligações entre neurônios são chamadas de *sinapses*.



Fig. 2.2.1: Neurônio biológico e seu componentes principais.

Os dendritos recolhem os sinais dos outros neurônios, via sinapses, e conduzem estes sinais para o núcleo do neurônio.

O núcleo é responsável pelo processamento que ocorre no neurônio. Os sinais provenientes dos dendritos são somados e caso o total ultrapasse um determinado limiar, a célula dispara e envia um sinal pelo axônio em direção a outras células. Alguns sinais tendem a inibir o disparo da célula enquanto outros tem caráter excitatório. Estas características são as mais básicas das redes neurais biológicas e são repetidas nas redes artificiais.

Além das características básicas já citadas, existem muitas outras que não são repetidas nas redes artificiais. Isto se deve ao distanciamento cada vez maior entre as pesquisas com RNAs e o estudo dos sistemas biológicos.

Uma diferença importante entre redes neurais artificiais e biológicas, diz respeito ao tamanho e complexidade. As redes neurais biológicas possuem, em média, 100 bilhões de neurônios formando  $10^{14}$  sinapses [29], enquanto as redes neurais artificiais apresentam no

máximo algumas centenas de neurônios. Esta diferença de tamanho, entre outros fatores, condiciona a diferença de capacidade e desempenho entre sistemas biológicos e artificiais.

Devido a grande diferença entre os tipos de rede neural, fica cada vez mais vazia a analogia entre sistemas biológicos e artificiais. Uma ênfase nesta analogia levou a criação de grandes expectativas em relação ao desempenho das redes neurais artificiais, que não foram concretizadas e quase puseram em questão a capacidade e relevância destas últimas.

### 2.3. Unidades de processamento

Os componentes básicos de uma rede neural artificial são as unidades de processamento, também chamadas de *neurônios* ou *neurônios artificiais* em analogia aos sistemas biológicos.

O primeiro modelo de unidade de processamento foi proposto por MCCULLOCH e PITTS [36] em 1943. O objetivo era modelar o comportamento das células nervosas e para a época, o modelo de McCulloch - Pitts baseava-se no conhecimento disponível acerca dos sistemas biológicos. Hoje sabe-se que o modelo proposto em 1943 está bastante distante da realidade destes últimos. No entanto, o modelo de McCulloch - Pitts norteou o desenvolvimento dos modelos subsequentes para neurônios artificiais e a configuração dos modelos mais recentes não difere muito daquela do modelo de 1943.

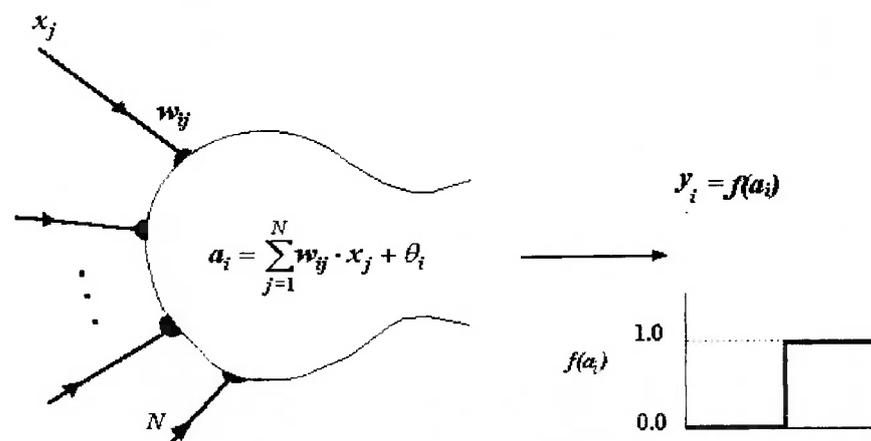


Fig. 2.3.1 Neurônio de McCulloch-Pitts

O neurônio de McCulloch-Pitts pode ser dividido em duas partes. A primeira parte engloba as conexões de entrada do neurônio. Segundo este modelo, a cada entrada,  $x_j$ , está associado um *peso*,  $w_{ij}$ . Os valores de entrada são multiplicados por estes pesos e então somados.

O total desta soma é a *entrada líquida* do neurônio. O termo  $-\theta_i$  é o valor do limiar do neurônio. Este termo pode ser considerado para fim de deduções matemáticas futuras como um dos pesos que é sempre multiplicado por uma entrada unitária.

Esta primeira parte do neurônio artificial consiste numa boa aproximação do que ocorre num neurônio biológico e é adotada em quase todos os tipos de redes neurais artificiais. Matematicamente, diz-se que a esta parte do neurônio está associada uma função, a *função de rede* ou *função de base*. No neurônio de McCulloch - Pitts esta função é uma combinação linear das entradas segundo os pesos das conexões, logo pode-se representar esta função, para um neurônio  $i$  com  $N$  entradas, como sendo:

$$a_i = \sum_{j=1}^N w_{ij} \cdot x_j \quad (2.3.1)$$

Ou na notação vetorial:

$$a_i = \mathbf{w}_i^t \mathbf{x} \quad (2.3.2)$$

Embora seja a mais aplicada, a combinação linear não é a única função de rede empregada. Pode-se citar ainda as funções *produtório* (2.3.3), *máximo ou mínimo* (2.3.4) e *soma cumulativa* (2.3.5), onde o cálculo da entrada líquida leva em conta não apenas as entradas e os pesos como também o valor da entrada líquida calculada numa iteração anterior.

$$a_i = \prod_{j=1}^N w_{ij} \cdot x_j \quad (2.3.3)$$

$$a_i = \max\{w_{ij} \cdot x_j\} \quad j = 1, 2, \dots, N \quad (2.3.4)$$

$$a_i(t) = a_i(t-1) + \sum_{j=1}^N w_{ij} \cdot x_j \quad (2.3.5)$$

Há ainda a função de rede radial citada por KUNG [29]. No entanto, esta função é de emprego bem mais restrito que a linear. Existe um tipo específico de rede neural, as *redes com função de base radial*, que empregam este tipo de função. Estas redes são recentes e até há pouco tempo eram muito pouco empregadas apesar de apresentarem algumas vantagens em relação a outros tipos de RNA [51].

A função de rede representa as conexões entre neurônios. Os pesos indicam o efeito da conexão entre dois neurônios. Valores elevados para o peso associado a uma conexão indicam que esta é bastante importante. Valores próximos a zero fazem com que esta conexão seja

relativamente pouco importante para o neurônio. Na função de rede radial, procura-se verificar o quanto uma determinada entrada está distante de um determinado valor, sendo este valor o peso da conexão.

A segunda parte do neurônio artificial recebe a entrada líquida, calculada pela função de rede e calcula a *ativação* do neurônio segundo uma função. Esta função é denominada *função de transferência* do neurônio, *função de ativação* ou, simplesmente, *função do neurônio*.

No neurônio de McCulloch-Pitts esta função de transferência consiste num degrau (Threshold). O desenvolvimento das redes neurais levou a adoção de outros tipos de função de transferência. As principais são: Degrau, rampa, logística, tangente hiperbólica e gaussiana.

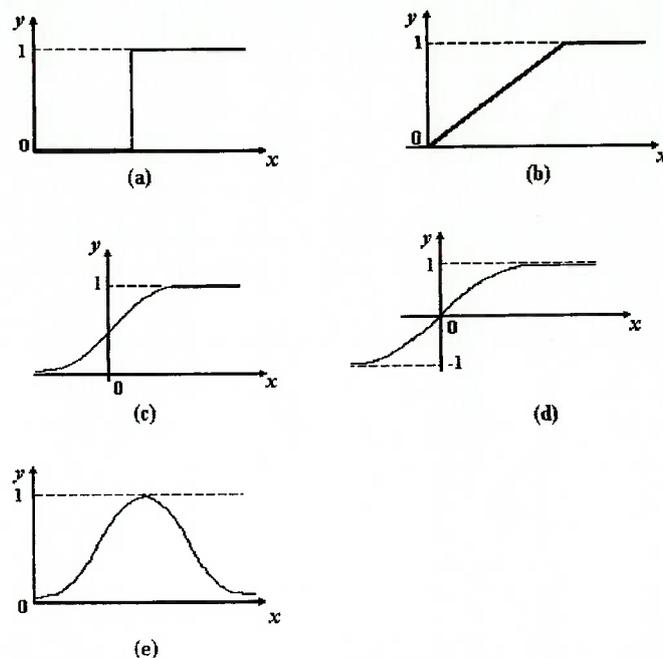


Fig 2.3.2. Funções de transferência: (a) Degrau, (b) Rampa, (c) Logística, (d) Tan. Hiperbólica, (e) Gaussiana.

O emprego de funções de transferência como a logística vem de encontro a um problema comum quando se usa redes neurais: Como uma mesma rede pode manipular entradas de grande magnitude e pequena magnitude ao mesmo tempo. Este problema foi levantado por GROSSBERG [16] que o resolveu propondo o uso de funções de transferência onde o ganho (Entendendo o ganho como sendo a derivada da função de transferência calculada para o valor da entrada) entre entrada e saída é variável, de tal forma que entradas pequenas fiquem sujeitas a ganhos maiores do que as entradas de grande magnitude.

As funções de transferência do tipo logística ou tangente hiperbólica fazem parte de uma classe de funções chamada sigmóide. Esta denominação provém do formato do gráfico destas funções, isto é, uma forma de "S". As funções do tipo sigmóide são ideais para resolver o problema apresentado no parágrafo anterior.

A função de transferência do tipo rampa com saturação possui características similares as das sigmóides no que diz respeito a variação do ganho. No entanto, ela não é largamente empregada pois não é diferenciável em todos os pontos e esta característica é necessária para certos tipos de processo de treinamento.

Uma alternativa a função logística é a tangente hiperbólica, cuja curva apresenta o mesmo formato da função logística e no entanto os limites desta função são diferentes. Ressalta-se que em alguns casos uma das funções pode levar a uma convergência mais rápida do processo de treinamento.

A propriedade de variação do ganho pode ser facilmente verificada observando-se a derivada da função logística, uma vez que o ganho é dado justamente por esta.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.3.6)$$

$$f'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = f(x) \cdot (1 - f(x)) \quad (2.3.7)$$

Para valores de entrada  $x$  positivos e elevados,  $f(x) \cong 1$ . Logo,  $f'(x) \cong 0$  e por conseguinte o ganho é mínimo. O mesmo pode ser dito para valores de  $x$  menores que zero mas de valor absoluto bastante elevado. Para este caso  $f(x) \cong 0$  e  $f'(x) \cong 0$ . Para valores de  $x$  próximos a zero,  $f'(x) \cong 1/4$  e este será o ganho máximo.

Devido a função de transferência do tipo degrau unitário, o neurônio de McCulloch-Pitts [36] é considerado um dispositivo binário, isto é, se  $a_i$  for maior que um certo valor a saída será igual a um, se for menor, então a saída será nula. Com os valores apropriados para os pesos das conexões, neurônios como o da figura 2.3.2 podem implementar as funções E (AND) ou OU (OR) da álgebra booleana. Sabe-se que com estas duas funções é possível implementar qualquer outra função booleana mais complexa [27]. No entanto, McCulloch e Pitts não propuseram nenhum algoritmo para a determinação ou cálculo dos pesos.

A saída do neurônio, calculada pela função de transferência, é única. Assim, um neurônio envia para as outras unidades de processamento sempre o mesmo valor de saída, que no entanto poderá vir a ser ponderado por pesos diferentes para cada conexão.

#### 2.4. Redes neurais

Após a proposição do neurônio artificial, o passo seguinte no desenvolvimento das redes neurais foi o agrupamento de neurônios, formando uma rede e o desenvolvimento de um algoritmo para determinação dos pesos relativos às conexões entre neurônios.

As primeiras redes neurais foram chamadas de *Perceptrons*. Propostas no final da década de 50 por ROSENBLATT [43], os perceptrons consistiam numa rede composta por neurônios similares ao de McCulloch-Pitts.

Nas redes neurais, os neurônios artificiais são dispostos em *camadas* (ver figura 2.4.1). A primeira camada, composta pelos neurônios que recebem as entradas, é denominada *camada de entrada*. Os neurônios desta camada não realizam qualquer processamento. Os neurônios que produzem os valores finais da rede neural formam a camada de saída. As camadas entre a entrada e a saída são chamadas de *ocultas* (hidden layers). Os perceptrons eram redes com apenas uma camada de neurônios. A camada de entrada não é computada no total de camadas da rede neural pois só devem ser consideradas as camadas que possuam neurônios com capacidade de processamento. Uma rede com uma camada de neurônios possui além desta, uma outra camada, a de entrada, cuja função é apenas distribuir as entradas para os neurônios da camada seguinte. Assim, a rede da figura 2.4.1 possui três camadas de neurônios.

Uma outra maneira de contar as camadas de uma RNA é considerar as camadas de pesos  $w_{ij}$ . Desta forma evita-se uma eventual confusão na contagem de camadas devido às características especiais dos neurônios da camada de entrada.

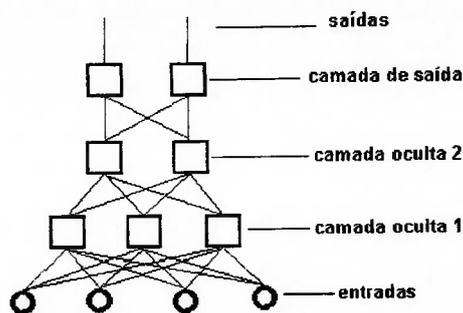


Fig. 2.4.1 Rede neural multicamada com duas camadas ocultas

A outra característica dos perceptrons era a existência de um algoritmo para determinação dos pesos. No modelo de McCulloch-Pitts não havia nenhuma previsão de como seriam calculados estes pesos. Em problemas simples como implementação de funções E (AND) e OU (OR), a determinação dos pesos é imediata. No entanto, para funções mais complexas, a determinação dos pesos é tarefa bastante complicada. Este algoritmo é considerado hoje como sendo o primeiro método de *treinamento* de redes neurais artificiais.

Foi provado por ROSENBLATT [43] que um perceptron poderia aprender tudo aquilo que pudesse representar, entendendo por representação a capacidade de uma rede neural em implementar uma determinada função.

Este último teorema e o algoritmo de treinamento dos perceptrons foram provados somente para redes com apenas uma camada de neurônios [43]. Como será visto mais adiante esta configuração é bastante limitada. O exemplo mais usado para apresentar estas limitações é o da função booleana OU Exclusivo (XOR), que não pode ser implementada por um perceptron [52]. Tal fato foi notado por MINSKY e PAPPERT [37] em 1969. Como Rosenblatt não conseguia uma extensão de seu teorema para redes com mais camadas de neurônios, Minsky e Pappert colocaram em xeque as possibilidades de emprego e desenvolvimento dos perceptrons. Após a publicação do trabalho de Minsky e Pappert as pesquisas com redes neurais ficaram estagnadas por aproximadamente 15 anos, havendo apenas desenvolvimentos pontuais e esporádicos.

### 2.4.1 O problema da representação

Conforme já foi mencionado anteriormente, representação pode ser entendida como a capacidade que uma rede neural tem em implementar uma determinada função. Como exemplo, considera-se uma rede que consegue identificar alguns tipos diferentes de flores tendo como informação para esta identificação o número de pétalas e sépalas. Será possível representar esta função usando uma rede neural? Será possível ajustar os pesos desta rede neural de tal forma que a classificação possa ser feita? Se for possível, então diz-se que a rede neural representa a função desejada.

#### 2.4.1.1 Representação com uma camada de neurônios

Seja um neurônio de McCulloch-Pitts com duas entradas binárias.

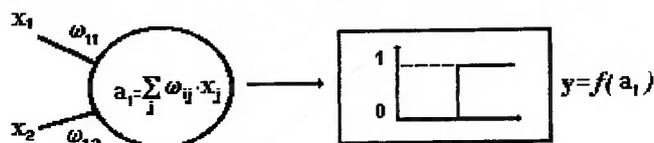


Fig. 2.4.1.1 Neurônio com duas entradas

$$a_1 = w_{11}x_1 + w_{12}x_2 \quad (2.4.1.1)$$

$$y = 1 \text{ se } a_1 > 0,5$$

$$y = 0 \text{ se } a_1 < 0,5$$

Percebe-se que o neurônio acima divide o espaço em duas regiões. Uma onde as entradas contidas no vetor  $\mathbf{x} = \{x_1, x_2\}$  provocam saída igual a 1 e outra onde as entradas provocam saída igual a 0 conforme é mostrado na figura 2.4.1.2.

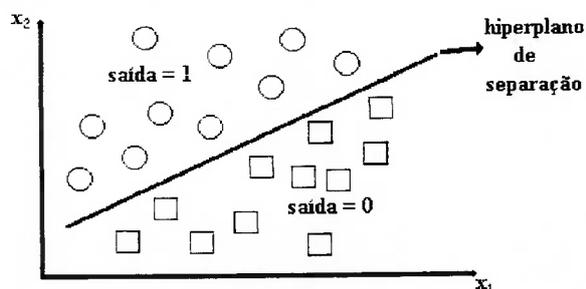


Fig. 2.4.1.2 Separação em duas categorias

A separação entre as duas regiões é feita por uma reta cuja equação é dada por:

$$x_1w_1 + x_2w_2 = \text{cte} \quad (2.4.1.2)$$

Genericamente pode-se dizer que um neurônio de McCulloch-Pitts funciona como um discriminador linear, isto é, consegue distinguir se conjuntos de dados pertencem a classes distintas desde que estas classes sejam linearmente separáveis.

Duas classes são ditas linearmente separáveis se for possível fazer tal separação por meio de um hiperplano linear [27]. A seguir são apresentadas duas situações. Numa é possível fazer uma separação linear, no caso por uma reta (caso especial de hiperplano) e outra onde tal separação é impossível.

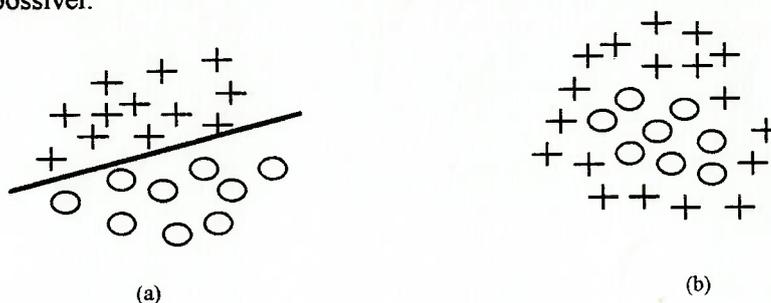


Fig. 2.4.1.3 (a) Classes linearmente separáveis: (b) Classes não separáveis linearmente

No exemplo apresentado anteriormente (Fig. 2.4.1.2), o hiperplano de separação é uma reta. Caso houvesse três entradas no neurônio, o hiperplano seria um plano no espaço euclidiano.

A separação linear é característica de perceptrons com uma camada de neurônios quando estes tem função de rede linear e função de transferência do tipo degrau.

#### 2.4.1.2 Representação com duas ou mais camadas de neurônios

Desde que se determinaram as limitações, quanto a capacidade de representação, das redes neurais com uma camada de neurônios, ficou claro que tais limitações podiam ser superadas pelas redes com mais de uma camada de neurônios.

Pode-se entender uma rede com mais de uma camada de neurônios como uma combinação de redes de uma camada, como pode ser visto na figura 2.4.1.4 .

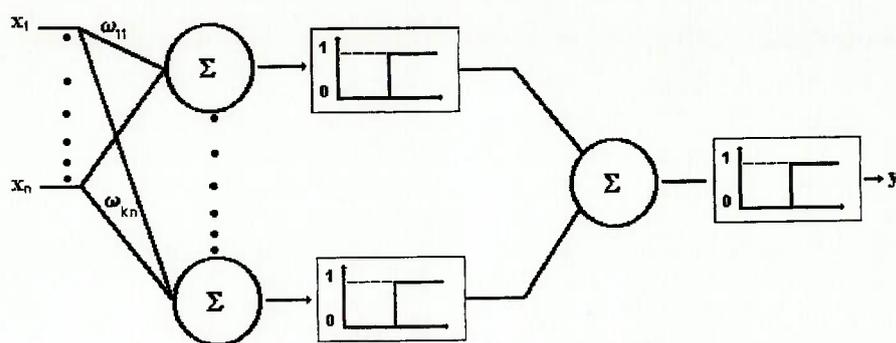


Fig. 2.4.1.4 Rede neural com duas camadas

A capacidade de representação de uma rede como a da figura também pode ser entendida como uma combinação das capacidades de representação de cada rede individualmente. Esta combinação dependerá dos pesos associados às conexões entre a primeira e a segunda camada, bem como da função de rede e de transferência dos neurônios na segunda camada.

Uma RNA com unidades de processamento binárias e duas camadas de neurônios consegue, por exemplo, separar pontos contidos em regiões convexas abertas ou fechadas. Entende-se por região convexa toda aquela onde dois pontos quaisquer podem ser unidos por uma reta que não deixa a região. Uma região fechada tem todos os seus pontos contidos por um contorno. Na figura 2.4.1.5 vê-se exemplos de regiões abertas e fechadas.

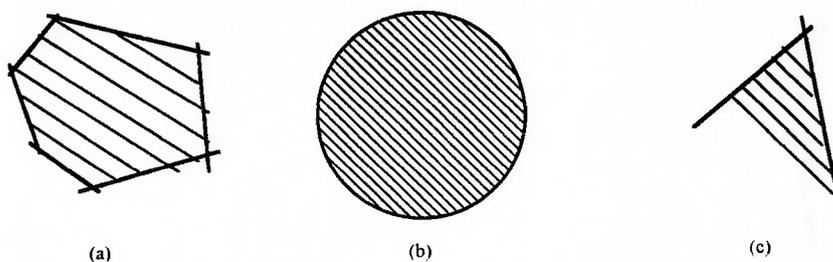


Fig. 2.4.1.5 Superfícies convexas. (a) e (b) fechadas; (c) aberta

No caso de uma rede com duas camadas de neurônios e duas entradas distintas, só é possível separar uma categoria de dados de outra, se uma delas for representada por uma região convexa [52]. Esta é uma limitação considerável e que ainda deixa restrita a aplicação destas redes com duas camadas. A figura 2.4.1.6 mostra esta restrição.

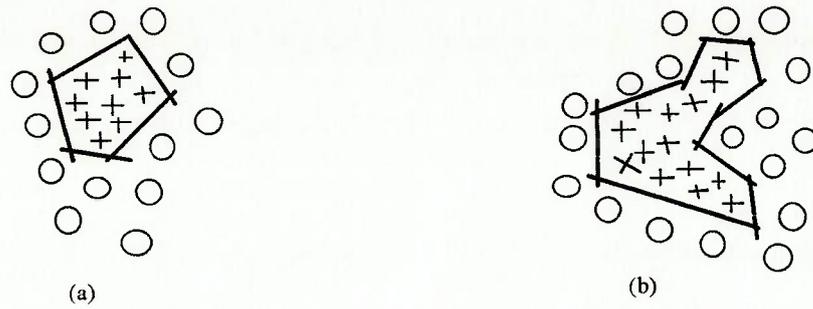


Fig. 2.4.1.6 (a) Classes separáveis por curva convexa; (b) classes não-separáveis por curva convexa

Uma rede com três camadas de neurônios consegue superar as restrições apresentadas pelas redes com uma e duas camadas. As limitações de uma rede com três camadas de dados ficam restritas ao número total de neurônios e, por conseguinte, ao número de pesos.

A rede com três camadas pode ser entendida como sendo uma combinação de duas redes de duas camadas (ver figura 2.4.1.7). Esta combinação depende apenas dos pesos entre a penúltima e última camada de neurônios. No entanto, esta combinação não leva a restrição alguma como uma região de representação convexa. Na verdade, esta combinação de duas superfícies convexas pode levar a uma superfície não convexa.

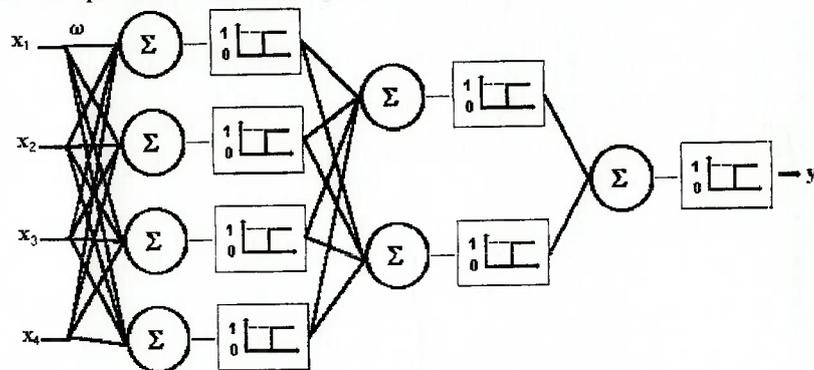


Fig. 2.4.1.7 Rede neural com três camadas

Na figura 2.4.1.8 está representada uma combinação de duas superfícies convexas. A combinação entre as duas superfícies consiste, neste caso, numa união das regiões resultando numa região não-convexa. Pode-se desta maneira, isto é, combinando regiões de representação convexas, obter regiões de representação quaisquer.

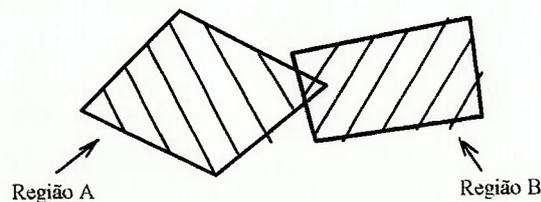


Fig. 2.4.1.8 União de duas superfícies de representação convexas resultando numa não-convexa.

Deve ser ressaltado que as vantagens das redes com mais de uma camada de neurônios só são válidas se, pelo menos, a função de transferência dos neurônios for não-linear. Caso a função de rede e a função de transferência sejam lineares não há vantagem alguma em ter uma rede com mais de uma camada de neurônios. Neste último caso, as camadas podem ser reduzidas a uma única, que é combinação linear das outras.

As conclusões acerca da capacidade de representação das redes multicamadas foram obtidas tendo como premissa a função de rede ser linear. Do contrário, a análise da capacidade de representação precisará ser refeita, como no caso de RNAs com função de base radial.

## 2.5. Processos de treinamento

Uma característica dos sistemas biológicos é a capacidade de aprendizado. Nas redes neurais artificiais considera-se como o aprendizado o processo de determinação dos pesos, de maneira que a rede possa implementar a função desejada.

Os métodos de treinamento de uma rede neural podem ser divididos em duas categorias:

- Aprendizado supervisionado
- Aprendizado não supervisionado.

O primeiro tipo de aprendizado é o mais comum e empregado nos diversos tipos de redes neurais existentes. O aprendizado supervisionado consiste em apresentar a rede neural uma série de dados de entrada e as respectivas saídas esperadas. Um algoritmo será empregado para ajustar os pesos das conexões entre os neurônios baseando-se no *erro* entre as saídas que a rede neural calcula e aquelas que seriam esperadas. Entre os algoritmos para o cálculo destaca-se o algoritmo de *retro-propagação* do erro que será detalhado mais adiante.

Este primeiro método de treinamento é denominado de supervisionado pois é necessário apresentar a rede as respostas esperadas. Uma desvantagem deste tipo de aprendizado é a necessidade de se dispor do conjunto de dados de treinamento. Dependendo do tamanho da rede, serão necessários vários casos de treinamento para que todos os pesos sejam obtidos de maneira ideal.

A escolha deste conjunto de treinamento deve ser criteriosa. Os dados devem ser representativos de todos os casos que eventualmente serão apresentados a rede neural.

O outro tipo de treinamento é denominado não-supervisionado. Neste caso, não é necessário apresentar a rede as respostas esperadas.

No treinamento não supervisionado os pesos são ajustados para que entradas similares produzam saídas similares. O algoritmo de treinamento extrai regularidades estatísticas do conjunto de dados de treinamento.

O aprendizado não supervisionado ainda é objeto de muitas pesquisas e poucas são as aplicações práticas onde este método foi empregado.

Fundamentos teóricos dos processos de treinamento de redes neurais datam de fins da década de 40, quando HEBB [17] publicou um trabalho sobre aprendizado de uma forma geral. O ponto central do trabalho de Hebb era que o aprendizado consistia em modificações nas sinapses motivadas por fenômenos restritos a estas, portanto, o aprendizado pode ser estudado, como um fenômeno restrito aos neurônios.

Matematicamente o aprendizado como proposto por Hebb, e adaptado a um modelo de neurônio como o de McCulloch-Pitts, pode ser escrito como:

$$w^{n+1} = w^n + \Delta w \quad (2.5.1)$$

$$\Delta w = \eta \cdot \delta \cdot x \quad (2.5.2)$$

A atualização do peso  $w$ ,  $\Delta w$ , depende do erro  $\delta$  entre a resposta calculada pelo neurônio e aquela esperada, da entrada  $x$  (a ser ponderada pelo peso  $w$ ) e de um parâmetro  $\eta$  que é a taxa de aprendizado. Nota-se que trata-se de um processo de treinamento supervisionado, pois é feita uma comparação entre as saídas esperadas e aquelas calculadas pelo neurônio.

O primeiro algoritmo de treinamento de redes neurais foi proposto por ROSENBLATT [43] para os perceptrons. Este algoritmo é restrito a redes cuja função de transferência é um degrau unitário e a função de rede é linear. A outra restrição diz respeito ao fato dos perceptrons serem redes com apenas uma camada de neurônios.

Este primeiro algoritmo de treinamento foi uma implementação de um aprendizado supervisionado. Só há modificação nos pesos se a saída calculada for diferente do valor esperado.

Conforme já foi mencionado, para um perceptron o processamento consiste inicialmente no cálculo da ativação,  $a_i$ , de um neurônio  $i$ , a partir do vetor de entradas  $\mathbf{x}$  e dos pesos  $\mathbf{w}$ :

$$a_i = \mathbf{w}_i^t \cdot \mathbf{x} \quad (2.5.3)$$

Em seguida há o cálculo da saída  $y_i$ , de acordo com uma função do tipo degrau unitário.

Segundo WASSERMAN [52], o algoritmo de treinamento pode ser resumido nos seguintes passos:

- 1 - Aplicar um padrão de entrada e calcular a saída  $y$ .
- 2 a - Se a saída for correta então retornar ao passo 1.
- b - Se a saída for incorreta e for igual a zero, somar cada entrada a seu respectivo peso.
- c - Se a saída for incorreta e for igual a um, subtrair cada entrada de seu respectivo peso.
- 3 - Voltar ao passo 1.

### 2.5.1 Regra do delta

O passo seguinte, após o desenvolvimento dos perceptrons, foi a obtenção por WIDROW e HOFF [53] da regra do delta (Delta rule), como algoritmo de treinamento de redes do tipo ADALINE.

A ADALINE foi uma rede neural desenvolvida por Widrow poucos anos depois do desenvolvimento dos perceptrons por Rosenblatt. A rede de Widrow era composta por uma unidade de processamento com função de transferência linear, daí resulta o nome ADALINE (ADaptive LINear Element). Um desenvolvimento posterior levou a MADALINE, com múltiplos neurônios.

A principal inovação introduzida por Widrow foi a regra do delta para treinamento da ADALINE. Posteriormente a regra do delta foi adaptada para outros tipos de rede neural, sendo a base do tipo mais empregado atualmente, as redes multicamadas com retro-propagação do erro (backpropagation).

A diferença entre o método de treinamento dos perceptrons e aquele proposto por WIDROW e HOFF [53], é que o primeiro tinha como objetivo determinar o conjunto de pesos que levam a uma classificação correta das entradas em categorias. Já o que foi proposto por Widrow era um processo de treinamento cujo objetivo era encontrar um conjunto de pesos que minimizam uma função critério, em geral o erro quadrático entre as saídas esperadas e aquelas calculadas pela rede. No caso dos perceptrons, qualquer conjunto de pesos que leve às saídas desejadas pode ser adotado, enquanto na ADALINE existe um *único* conjunto de pesos que é procurado.

Sendo a ADALINE uma rede neural linear, a saída é calculada por:

$$y = \mathbf{w}^t \cdot \mathbf{x} \quad (2.5.1.1)$$

Neste caso, o vetor de pesos  $\mathbf{w}$ , também engloba a função de transferência linear da ADALINE.

A função critério é o erro quadrático médio, a ser minimizado

$$E(\mathbf{w}) = \frac{1}{2} \sum_{m=1}^M (y_m - t_m)^2 \quad (2.5.1.2)$$

Sendo  $M$  o total de componentes do conjunto de treinamento, composto por vetores de entrada,  $\mathbf{x}_m$ , e saídas esperadas,  $t_m$ .

Nota-se que  $E(\mathbf{w})$  é convexa e portanto existe um mínimo, dado pelo vetor de pesos  $\mathbf{w}^*$ . Uma maneira de calcular este vetor é pelo cálculo do gradiente de  $E(\mathbf{w})$ , igualando este a zero. O vetor  $\mathbf{w}^*$  que anula o gradiente corresponde a aquele que fornece o mínimo erro.

$$\begin{aligned} \text{grad } E(\mathbf{w}) &= \frac{1}{2} \cdot \frac{d}{d\mathbf{w}} \sum_{m=1}^M (\mathbf{w}^t \cdot \mathbf{x}_m - t_m)^2 \\ &= \sum_{m=1}^M \mathbf{x}_m \cdot (\mathbf{w}^t \cdot \mathbf{x}_m - t_m) = 0 \\ &= \left[ \sum_{m=1}^M \mathbf{x}_m \cdot \mathbf{x}_m^t \right] \cdot \mathbf{w} - \sum_{m=1}^M \mathbf{x}_m \cdot t_m = 0 \end{aligned} \quad (2.5.1.3)$$

A obtenção do vetor  $\mathbf{w}^*$  que minimiza o erro quadrático requer a inversão da matriz dada por:

$$\mathbf{A} = \left[ \sum_{m=1}^M \mathbf{x}_m \cdot \mathbf{x}_m^t \right] \quad (2.5.1.4)$$

Nem sempre é possível obter a inversa desta matriz. Seria necessário que, primeiramente os vetores de entrada  $\mathbf{x}_m$ , sejam linearmente independentes. Outro aspecto relevante é que sendo esta matriz inversa obtida por meio de métodos numéricos, sempre se está sujeito a existência de uma matriz mal condicionada e erros devidos a imprecisões do processo numérico adotado.

A regra do delta proposta por Widrow, evita este cálculo de matriz inversa. Esta regra baseia-se no fato de que, partindo-se de um ponto e seguindo no sentido contrário do gradiente de  $E(\mathbf{w})$ , é possível encontrar o mínimo da função de erro. Matematicamente este processo pode ser apresentado como sendo:

$$\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} - \eta \cdot \text{grad} (E(\mathbf{w})^{(n)}) \quad (2.5.1.5)$$

Sendo:

$$\begin{aligned} \text{grad} E(\mathbf{w}) &= \sum_{m=1}^M \mathbf{x}_m \cdot (\mathbf{w}_m^t \cdot \mathbf{x}_m - t_m) \\ &= \sum_{m=1}^M \mathbf{x}_m \cdot (y_m - t_m) \end{aligned} \quad (2.5.1.6)$$

$$\begin{aligned} \Rightarrow \mathbf{w}_{(n+1)} &= \mathbf{w}_{(n)} + \eta \cdot \sum_{m=1}^M \mathbf{x}_m \cdot (y_m - t_m) \\ \Rightarrow \mathbf{w}_{(n+1)} &= \mathbf{w}_{(n)} + \eta \cdot \sum_{m=1}^M \mathbf{x}_m \cdot \delta_m \end{aligned} \quad (2.5.1.7)$$

$$\delta_m = y_m - t_m \quad (2.5.1.8)$$

O parâmetro  $\eta$  é a taxa de aprendizado, que determina o quanto se caminha do sentido oposto ao do gradiente. Este parâmetro atua decisivamente na convergência do processo de treinamento. Valores muito pequenos para  $\eta$  implicam em muitas iterações para o treinamento. Já valores elevados da taxa de aprendizado podem levar a instabilidades no treinamento. No caso da ADALINE que é um sistema linear, é possível encontrar valores ideais para a taxa de aprendizado como é mostrado por KUNG [29].

O cálculo dos pesos é feito de forma iterativa. Então é necessário um critério de convergência para este processo. Uma primeira possibilidade é a comparação do erro quadrático médio entre duas iterações. Outra possibilidade é a imposição de um número máximo de iterações para o treinamento. A importância do critério de parada fica evidente quando se pensa num programa computacional para a implementação de uma RNA.

Widrow desenvolveu a regra do delta para uma rede neural com uma camada de neurônios e com função de transferência linear em cada um destes. Já foi mostrado (ver item 2.4.1.1) que redes com estas características são de aplicação bastante restrita pois possuem uma série de limitações. Entretanto, desenvolvimentos subsequentes levaram a aplicação da regra do delta a redes com múltiplas camadas de neurônios e função de transferência não-linear.

## 2.6. Algoritmo de retro-propagação do erro

Embora já fossem conhecidas as vantagens de redes neurais com múltiplas camadas de neurônios e com funções de transferência não-lineares associadas a estes, não havia algoritmo para treinamento destas redes. Este fato, aliado aos problemas emergentes com as pesquisas envolvendo redes mais simples que eram devidos às próprias limitações desta últimas, levou ao temporário esquecimento das redes neurais.

Durante este período de esquecimento poucos foram os desenvolvimentos em redes neurais. Não havia uma linha para as pesquisas neste ramo, aliado ao fato de que as redes neurais tinham caído em descrédito. O algoritmo de retro-propagação, baseado numa adaptação da regra do delta de Widrow veio como resposta ao problema de treinamento de redes neurais multicamada, gerando toda uma nova corrente de pesquisas no que se chamou de "ressurgimento" das redes neurais.

O desenvolvimento deste algoritmo é atribuído a RUMELHART, HINTON e WILLIAMS [45]. No entanto existiram trabalhos prévios como o de Werbos que levaram a algoritmos similares ao de retro-propagação [3].

Uma rede neural multicamada é mostrada na figura 2.6.1 Os neurônio possuem função de transferência logística:

$$y = \frac{1}{1 + e^{-a}} \quad (2.6.1)$$

Além das características já mencionadas da função de transferência do tipo logística, pode-se acrescentar que a derivada desta função em relação a entrada líquida é facilmente calculada, o que é bastante conveniente na implementação do algoritmo:

$$\frac{dy}{da} = y \cdot (1 - y) \quad (2.6.2)$$

Antes de prosseguir com a apresentação do algoritmo de retro-propagação, cabe um comentário a respeito da notação a ser adotada daqui em diante. Para ilustrar a notação será utilizada a figura 2.6.1

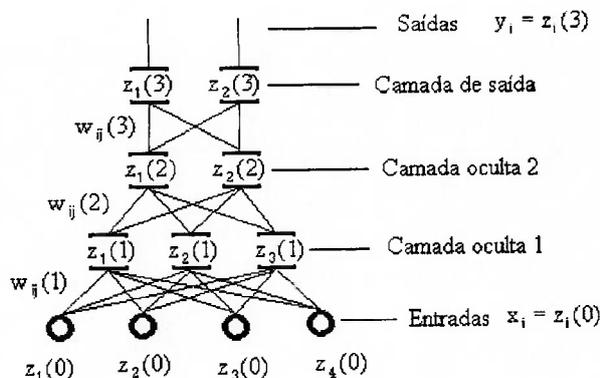


Fig 2.6.1 - Rede neural multi-camada

De acordo com esta notação, a saída de um neurônio  $i$  localizado numa camada  $k$  é dada por  $z_i(k)$ . As entradas deste neurônio são as saídas dos neurônios  $j$  da camada anterior,  $k-1$ . Logo as entradas de um neurônio são dadas por  $z_j(k-1)$ . As entradas da rede que antes eram denominadas  $x_j$ , agora passam a ser  $z_j(0)$  e as saídas da rede,  $y_j$ , serão  $z_j(K)$ , isto é, as saídas da camada  $K$ , onde  $K$  é o número de camadas da rede neural. No exemplo da figura  $K=3$ .

Os pesos de uma camada são escritos como sendo  $w_{ij}(k)$ , sendo o índice  $i$  indicativo do neurônio de destino, enquanto  $j$  designa as entradas do neurônio. Esta notação, embora possa ser confusa, é ideal para a apresentação do algoritmo de back-propagation, onde são necessários índices para os neurônios, para as camadas, mais um para indicar a iteração e ainda outro para indicação do caso de treinamento.

Feitas as considerações sobre a notação, a primeira etapa do algoritmo de retro-propagação consistirá no processamento das entradas para obtenção das saídas. Como em outras redes neurais, as entradas de um neurônio são somadas de forma ponderada, obtendo-se a entrada líquida. Esta última será usada para o cálculo da saída do neurônio de acordo com a função de transferência.

Para um neurônio  $i$ , localizado numa camada  $k$ , a etapa de cálculo das saídas, adotando-se a função de transferência logística, é dada por:

$$a_i(k) = \sum_{j=1}^n w_{ij}(k) \cdot z_j(k-1) \quad (2.6.3)$$

$$z_i(k) = f(a_i(k)) = \frac{1}{1 + e^{-a_i}} \quad (2.6.4)$$

O passo seguinte do algoritmo é o cálculo da atualização dos pesos para o caso de treinamento  $m+1$ . A expressão para modificação dos pesos é a seguinte:

$$w_{ij}^{(m+1)}(k) = w_{ij}^{(m)}(k) + \Delta w_{ij}^{(m)}(k) \quad (2.6.5)$$

A variação dos pesos  $\Delta w_{ij}^{(m)}(k)$  é obtida a partir do método do gradiente descendente como foi feito para a regra do delta de Widrow e o objetivo é encontrar o mínimo da função erro quadrático médio,  $E$ .

$$\Delta w_{ij}^{(m)}(k) = -\eta \cdot \frac{\partial E}{\partial w_{ij}^{(m)}(k)} \quad (2.6.6)$$

Aplicando a regra da cadeia obtém-se:

$$\Delta w_{ij}^{(m)}(k) = -\eta \cdot \frac{\partial E}{\partial z_i^{(m)}(k)} \cdot \frac{\partial z_i^{(m)}(k)}{\partial w_{ij}^{(m)}(k)} \quad (2.6.7)$$

Define-se o erro,  $\delta_i^{(m)}(k)$ , como sendo:

$$\delta_i^{(m)}(k) \equiv -\frac{\partial E}{\partial z_i^{(m)}(k)} \quad (2.6.8)$$

Logo:

$$\Delta w_{ij}^{(m)}(k) = -\eta \cdot \delta_i^{(m)}(k) \cdot \frac{\partial z_i^{(m)}(k)}{\partial w_{ij}^{(m)}(k)} \quad (2.6.9)$$

Sendo que:

$$\frac{\partial z_i^{(m)}(k)}{\partial w_{ij}^{(m)}(k)} = \frac{\partial z_i^{(m)}(k)}{\partial a_i^{(m)}(k)} \cdot \frac{\partial a_i^{(m)}(k)}{\partial w_{ij}^{(m)}(k)} \quad (2.6.10)$$

Mas:

$$\frac{\partial z_i^{(m)}(k)}{\partial a_i^{(m)}(k)} = f'(a_i^{(m)}(k)) \quad (2.6.11)$$

$$\frac{\partial a_i^{(m)}(k)}{\partial w_{ij}^{(m)}(k)} = z_j^{(m)}(k-1) \quad (2.6.12)$$

Portanto:

$$\frac{\partial z_i^{(m)}(k)}{\partial w_{ij}^{(m)}(k)} = f'(a_i^{(m)}(k)) \cdot z_j^{(m)}(k-1) \quad (2.6.13)$$

Assim, substituindo (6.13) em (6.9), obtém-se:

$$\Delta w_{ij}^{(m)}(k) = -\eta \cdot \delta_i^{(m)}(k) \cdot f'(a_i^{(m)}(k)) \cdot z_j^{(m)}(k-1) \quad (2.6.14)$$

Esta é a equação que fornece a atualização dos pesos, para um elemento  $m$  do conjunto de dados de treinamento. Resta agora obter o erro,  $\delta_i^{(m)}(k)$ , para as diferentes camadas da rede neural.

A função a ser minimizada é o erro quadrático,  $E$ , entre as saídas esperadas,  $t_i^m$ , e aquelas calculadas na camada de saída da rede neural,  $z_i^m(K)$ . Este erro é calculado para os  $M$  casos de treinamento e para os  $N$  neurônios da camada de saída.

$$E = \frac{1}{2} \sum_{m=1}^M \sum_{i=1}^N [t_i^m - z_i^m(K)]^2 \quad (2.6.15)$$

Para a camada de saída o valor do erro é facilmente calculado:

$$\delta_i^{(m)}(K) = -\frac{\partial E}{\partial z_i^{(m)}(K)} = t_i^m - z_i^m(K) \quad (2.6.16)$$

Já para as camadas ocultas não é possível calcular o erro como sendo a diferença entre as saídas esperadas e aquelas calculadas pela rede.

A novidade introduzida pelo algoritmo de retro-propagação é a maneira com que foi expandido o emprego da regra do delta em redes multicamada. Isto é conseguido transmitindo o erro da camada de saída para as camadas ocultas. Esta transmissão do erro é feita no sentido contrário daquele usado no cálculo das saídas da rede; esta é a razão do nome "retro-propagação".

Este algoritmo foi deduzido através do uso da regra da cadeia para derivadas parciais. O ponto de partida é a definição do erro numa camada  $k$ .

$$\delta_i^{(m)}(k) \equiv -\frac{\partial E}{\partial z_i^{(m)}(k)} \quad (2.6.8)$$

Um primeiro passo é fazer a aplicação da regra da cadeia, fazendo as derivadas parciais a respeito da entrada líquida dos neurônios da camada seguinte,  $a_j^{(m)}(k+1)$ :

$$\delta_i^{(m)}(k) = -\frac{\partial E}{\partial z_i^{(m)}(k)} = -\frac{\partial E}{\partial a_j^{(m)}(k+1)} \cdot \frac{\partial a_j^{(m)}(k+1)}{\partial z_i^{(m)}(k)} \quad (2.6.17)$$

Sendo que:

$$-\frac{\partial E}{\partial a_j^{(m)}(k+1)} = -\frac{\partial E}{\partial z_j^{(m)}(k+1)} \cdot \frac{\partial z_j^{(m)}(k+1)}{\partial a_j^{(m)}(k+1)} \quad (2.6.18)$$

Mas,

$$-\frac{\partial E}{\partial z_j^{(m)}(k+1)} = \delta_j^{(m)}(k+1) \quad (2.6.19)$$

e

$$\frac{\partial z_j^{(m)}(k+1)}{\partial a_j^{(m)}(k+1)} = f'(a_j^{(m)}(k+1)) \quad (2.6.20)$$

Logo:

$$-\frac{\partial E}{\partial a_j^{(m)}(k+1)} = \delta_j^{(m)}(k+1) \cdot f'(a_j^{(m)}(k+1)) \quad (2.6.21)$$

O outro componente do erro é dado por:

$$\frac{\partial a_j^{(m)}(k+1)}{\partial z_i^{(m)}(k)} = w_{ji}(k+1) \quad (2.6.22)$$

Substituindo (2.6.21) e (2.6.22) em (2.6.17) obtém-se:

$$\delta_i^{(m)}(k) = \sum_{j=1}^N \delta_j^{(m)}(k+1) \cdot f'(a_j^{(m)}(k+1)) \cdot w_{ji} \quad (2.6.23)$$

A equação (2.6.23) fornece o erro para um neurônio de uma camada oculta, como função dos sinais de erro dos neurônios da camada seguinte. De maneira esquemática podemos representar o algoritmo como na figura 2.6.2.

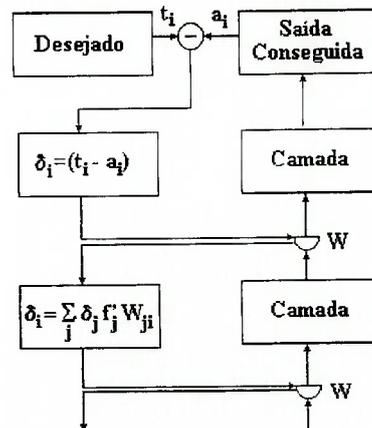


Fig. 2.6.2 Fluxograma para a retropropagação do erro

O algoritmo básico de retro-propagação do erro é apresentado, passo a passo, no quadro 2.6.1. Para a implementação deste algoritmo foram desenvolvidas algumas variantes. Uma primeira variação consiste na atualização dos pesos. No algoritmo original, os pesos só são alterados após a apresentação de todos os casos de treinamento. Em algumas implementações os pesos são ajustados após a apresentação de cada caso de treinamento.

### ALGORITMO DE RETRO-PROPAGAÇÃO DO ERRO

**Passo 1: Inicialização dos pesos**

Os pesos de cada conexão são inicializados aleatoriamente com valores não nulos.

**Passo 2: Apresentação das entradas e saídas desejadas**

Submete-se a rede neural um dos exemplos do conjunto de treinamento.

**Passo 3: Cálculo da saídas da rede neural artificial**

As entradas são propagadas através das camadas da rede neural segundo as funções de rede e de transferência dos neurônios. Estas saídas serão comparadas com aquelas esperadas no próximo passo.

**Passo 4: Atualização dos pesos**

Os pesos são atualizados de acordo com a expressão

$$\Delta w_{ij}^{(m)}(k) = -\eta \cdot \delta_i^{(m)}(k) \cdot f'(a_i^{(m)}(k)) \cdot z_j^{(m)}(k-1) \quad (2.6.14)$$

O valor do erro  $\delta_i^{(m)}$  é dado por:

Para os neurônios da camada de saída:

$$\delta_i^{(m)}(K) = t_i^m - z_i^m(K) \quad (2.6.16)$$

Para os neurônios das camadas intermediária:

$$\delta_i^{(m)}(k) = \sum_{j=1}^N \delta_j^{(m)}(k+1) \cdot f'(a_j^{(m)}(k+1)) \cdot w_{ji} \quad (2.6.23)$$

**Passo 5: Repete-se o processo desde o passo 2**

Até que o erro cometido pela rede neural esteja abaixo de um valor pré-determinado ou o número de iterações ultrapasse um certo limite.

Quadro 2.6.1 - Algoritmo de retro-propagação

Há que distinguir entre *iteração* e *varredura*. Uma *iteração* corresponde a apresentação de um par de treinamento (entradas e saídas esperadas) a rede neural. Já a *varredura* corresponde a apresentação de todo o conjunto de treinamento para a atualização dos pesos. Caso seja feita a atualização dos pesos após uma varredura, a tendência será equilibrar os efeitos de todo o conjunto de treinamento sobre a variação dos pesos. Uma vantagem da atualização dos pesos após uma iteração é a possibilidade de executar o aprendizado a medida que se usa a rede.

Uma das desvantagens do algoritmo de retro-propagação é o tempo necessário para o treinamento. Uma possível causa pode ser um valor inadequado para a taxa de aprendizado,  $\eta$ . Os efeitos deste parâmetro sobre a convergência deste algoritmo são similares àqueles apresentados na regra do delta de Widrow.

Outro fator que pode levar a lentidão do treinamento é a forma da superfície da função erro quadrático. Caso o gradiente desta função tenha uma variação muito pequena ao longo desta superfície, o treinamento será lento.

Um outro problema quando se usa o algoritmo de retro-propagação é a possibilidade do treinamento convergir para um *mínimo local* da função erro quadrático, ao invés do *mínimo global* conforme seria esperado. Para a ADALINE há sempre um mínimo global, garantido pelo fato das funções de transferência serem lineares. Já no caso de redes cujos neurônios possuem função de transferência não-linear, a função erro quadrático pode apresentar um ou mesmo vários mínimos locais. Não há meio de evitar a convergência para os mínimos locais sem que se modifique substancialmente o algoritmo. Alguns autores, entretanto, reportam que é muito pouco provável a convergência para um mínimo local [27].

Outro problema de convergência é a *paralisia* da rede causada quando esta opera com valores elevados para os pesos, o que faz com que as saídas calculadas pelos neurônios sejam muito altas. Isto provocará nas camadas próximas a de saída que os neurônios operem numa faixa onde o ganho da função de transferência dos mesmos é próximo de zero (lembrando que foram usadas funções como a logística). Como este ganho, representado pela derivada da função de transferência, é importante no cálculo do erro, vai ser bastante provável que este erro seja próximo de zero. Com isto deixa de haver modificação dos pesos e o treinamento fica paralisado.

De todos os problemas que podem ocorrer no treinamento, o mais comum é a lentidão do processo devido a superfície da função erro ser pouco inclinada, resultando em pouca variação do gradiente. Uma primeira solução para este problema é um cálculo mais preciso para o gradiente. Uma tentativa é tentar usar termos de ordem maior no cálculo do gradiente. Isto é feito a partir da expansão em série de Taylor do gradiente da função erro quadrático.

Em notação vetorial tem-se:

$$\text{grad}(E(\mathbf{w} + \Delta\mathbf{w})) = \text{grad}(E(\mathbf{w})) + \mathbf{H}(\mathbf{w}) \Delta\mathbf{w} \quad (2.6.24)$$

Onde  $\mathbf{H}(\mathbf{w})$  é a matriz Hessiana de  $E(\mathbf{w})$ . Esta matriz é dada por:

$$\mathbf{H}(\mathbf{w}) = \left[ \frac{\partial^2 E(\mathbf{w})}{\partial w_i \partial w_j} \right] \quad (2.6.25)$$

A condição de erro mínimo corresponde a  $\text{grad}(E(\mathbf{w} + \Delta\mathbf{w})) = 0$ . Logo:

$$\text{grad}(E(\mathbf{w})) + \mathbf{H}(\mathbf{w}) \Delta\mathbf{w} = \mathbf{0} \quad (2.6.26)$$

A atualização dos pesos,  $\Delta\mathbf{w}$ , será dada por:

$$\Delta\mathbf{w} = -\mathbf{H}^{-1}(\mathbf{w}) \text{grad}(E(\mathbf{w})) \quad (2.6.27)$$

Com o método de Newton, como é chamada esta maneira do cálculo do gradiente, não há taxa de aprendizado  $\eta$ , arbitrária e sim a inversa da matriz hessiana como fator que determinará o incremento na atualização dos pesos. Esta característica acaba tornando-se um inconveniente do ponto de vista computacional, justamente devido ao cálculo de uma matriz inversa. Entretanto, KUNG [29] apresenta uma série de métodos alternativos que evitam o cálculo de  $\mathbf{H}^{-1}(\mathbf{w})$ .

Outra maneira de aumentar a velocidade de convergência é a adoção do *termo de momento*. Este método soma à atualização dos pesos,  $\Delta w_{ij}^{(m)}(k)$ , um termo que é proporcional a última atualização feita,  $\alpha \Delta w_{ij}^{(m-1)}(k)$ .

$$\Delta w_{ij}^{(m)}(k) = -\eta \cdot \frac{\partial E}{\partial w_{ij}^{(m)}(k)} + \alpha \Delta w_{ij}^{(m-1)}(k) \quad (2.6.28)$$

Da mesma forma que a taxa de aprendizado,  $\eta$ , não há regra para determinação prévia do fator  $\alpha$  do termo de momento. Esta é uma desvantagem da adição do termo de momento. Valores de  $\alpha$  muito próximos a zero fazem com que o efeito do termo de momento seja mínimo sobre a atualização dos pesos.

Outra característica importante para a convergência do algoritmo é a inicialização dos pesos  $w_{ij}$  antes do treinamento. Se os pesos forem todos nulos, não haverá treinamento pois tanto entradas como erros serão multiplicados por estes pesos para o cálculos das atualizações  $\Delta w_{ij}^{(m)}(k)$ . Valores muito altos para os pesos podem levar a *paralisia* da rede neural.

Uma vez que o processo de treinamento de uma RNA tenha convergido para um mínimo global, pode-se garantir que, para as entradas pertencentes ao conjunto de treinamento, o erro na camada de saída será mínimo. Entretanto, deseja-se que a rede possa ser usada com valores de entrada não pertencentes ao conjunto de treinamento e que o desempenho da rede não seja prejudicado, isto é, as saídas devem estar em conformidade com o que se espera da rede neural.

Se tal fato ocorre, diz-se que a RNA consegue generalizar, isto é, fornecer saídas adequadas para valores de entrada quaisquer, que não são necessariamente os mesmos do treinamento.

Ressalta-se que um bom desempenho em generalização é restrito a interpolação. Isto implica que os limites do conjunto de treinamento sejam também os limites das possíveis entradas da rede neural. A generalização não implica em capacidade de extrapolação.

A precisão do treinamento melhora com o aumento no número de neurônios nas camadas ocultas. Não há nenhum método para a determinação de quantos neurônios serão necessários para que se tenha uma determinada precisão de treinamento.

Para a generalização há maiores restrições quanto ao número de neurônios nas camadas ocultas. Um número excessivo de neurônios pode levar a uma rede sem capacidade de generalização. Logo, há um compromisso entre a precisão do treinamento, que aumenta com o número de neurônios, e a precisão de generalização.

Uma maneira de resolver este problema, assim como a convergência para um mínimo local, é o emprego de técnicas de crescimento e corte (prunning) de rede.

Nos métodos de corte, parte-se de uma rede com muitas unidades de processamento nas camadas ocultas e que já foi treinada. O procedimento consiste em excluir algumas unidades das camadas ocultas na tentativa de melhorar a capacidade de generalização.

Já os métodos de crescimento de rede acrescentam neurônios à camadas ocultas a partir de uma rede com poucos neurônios. O objetivo, neste caso, é melhorar o treinamento de forma que apenas sejam treinadas as unidades necessárias e não uma rede com muitos neurônios.

Existem ainda maneiras mais simples para melhorar o desempenho das redes neurais. Uma medida muito adotada é a adoção de um termo que ajusta a origem da função de transferência. O efeito é similar ao ajuste do limiar num neurônio de um perceptron. Para fins de treinamento este termo é considerado como um peso treinável que está ligado a um entrada cujo valor é mantido constantemente igual a um.

A despeito destas limitações, o algoritmo de back-propagation é o mais aplicado. Foi a obtenção deste algoritmo que permitiu o retorno das pesquisas com redes neurais. Embora as expressões apresentadas possam parecer complexas, por conta dos índices, o algoritmo de retro-

propagação é facilmente implementável em computador, sendo esta outra a razão para o seu sucesso.

## **2.7. Diretrizes para a aplicação de RNAs**

O desenvolvimento das redes neurais artificiais gerou o aparecimento de várias aplicações para as mesmas. Entretanto, apesar das características peculiares das RNAs, é necessário tomar certos cuidados quando são propostas aplicações. Segundo WASSERMAN [51]: "Usar uma rede neural onde não é apropriado irá resultar sempre numa solução inferior, senão numa falha completa". O objetivo deste tópico é fornecer algumas diretrizes para a correta aplicação das RNAs.

A ênfase das diretrizes apresentadas está na adequação da rede neural ao problema a ser resolvido e a geração do conjunto de dados de treinamento.

### **2.7.1 Adequação da rede ao problema a ser resolvido**

Conforme já foi mencionado anteriormente, redes neurais artificiais não podem ser usadas para resolver todos os problemas. Se há um método matemático ou algoritmo conhecido para a solução do problema, estes devem ser usados preferencialmente em relação às RNAs. No entanto, se não é conhecido um algoritmo ou a dedução de um não é possível na prática pode-se pensar em aplicar uma rede neural. Um exemplo de problema não indicado para ser resolvido usando redes neurais é a inversão de matrizes [51].

De uma forma geral, pode-se classificar as aplicações de redes neurais nas seguintes categorias:

**Classificação:** Indicar a qual classe uma determinada entrada pertence.

**Aproximação:** Aproximar uma função com mínimo erro.

**Associação:** Uma vez dada parte de um padrão, obter o restante deste.

A tabela 2.7.1 apresenta as características de treinamento e geração de resultados para alguns das categorias citadas.

Tipo de treinamento	Aplicação	Formulação do treinamento	Formulação do cálculo das saídas
Supervisionado	Classificação	Dados $x_i$ , o símbolo desejado ( $s_i$ ), achar $W$ , t.q. $y=s_i$	Dados $x_i$ e $W$ determinar o símbolo $s_i$
	Aproximação	Dados $x_i$ e os valores esperados $t_i$ , achar $W$ t.q. $\sum_i [t_i - \Phi(x_i, W)]^2$ seja mínima	Dados $x_i$ e $W$ , achar o valor de $y = \Phi(x_i, W)$
Não supervisionado	Classificação	formação de aglomerados	Dado $x_i$ determinar a qual grupo ele pertence
Pesos fixos	Associação	Não há	Dado $W$ e $x'$ , achar o $x$ que é mínimo de $E(x, W)$

Tabela 2.7.1 Formulações para treinamento e cálculo das saídas (modificado a partir de Kung[29])

### 2.7.2 Geração do conjunto de dados de treinamento

Outro fator importante para uma satisfatória aplicação de redes neurais artificiais é o conjunto de dados disponíveis para o treinamento.

Segundo WASSERMAN [51], o conjunto de treinamento deve ser uma amostra estatisticamente significativa do problema. Do contrário, a rede terá aplicação restrita uma vez que a generalização não garante que a rede possa extrapolar.

Outro ponto importante na montagem do conjunto de treinamento é o número de dados de treinamento. O tamanho deste conjunto é condicionado pelo tamanho da RNA. Quanto maior a rede, e portanto com um grande número de pesos para serem determinados, maior deve ser o conjunto de treinamento.

Da mesma forma que o número de neurônios, não há teoria para determinação do tamanho ótimo do conjunto de treinamento. WASSERMAN [51] faz algumas considerações a este respeito baseando-se em experiências anteriores com o uso de redes neurais.

Para a verificação da generalização é necessário um outro conjunto de entradas com saídas esperadas que deve ser distinto do conjunto usado no treinamento. Este novo conjunto, o conjunto de dados de teste da RNA, tem características similares ao de treinamento. A verificação da generalização é feita pelo projetista da rede neural. O julgamento a respeito da

generalização pode ser feito com base no erro cometido pela RNA quando esta é submetida aos dados de teste.

### 3. REVISÃO BIBLIOGRÁFICA

O objetivo desta revisão bibliográfica é apresentar o estado da arte no assunto estudado. Foi feita uma divisão em três tópicos principais que são, em essência, os mesmos apresentados neste trabalho. São eles:

- Redes neurais artificiais e suas aplicações.
- Redes neurais artificiais aplicadas à detecção de defeitos em estruturas
- Ensaios não destrutivos baseados em modificações na resposta dinâmica de estruturas.

Para a seleção dos trabalhos analisados, além do critério de afinidade ao tema aqui desenvolvido, também foram considerados os textos mais relevantes do ponto de vista do desenvolvimento histórico do tema, ainda que hoje alguns conceitos apresentados nestes trabalhos já sejam considerados ultrapassados.

#### 3.1 Redes neurais artificiais e suas aplicações

Neste grupo estão incluídos os trabalhos referentes ao desenvolvimento das redes neurais desde a década de 40, bem como algumas aplicações de redes neurais artificiais. Maiores detalhes a respeito das RNAs e de alguns trabalhos citados estão no capítulo 2 - Redes neurais artificiais.

A origem das redes neurais artificiais pode ser considerada como sendo a publicação por McCULLOCH e PITTS [36] do trabalho "*A Logical Calculus of the ideas immanent in nervous activity*", embora existam trabalhos prévios a respeito de aprendizado e funcionamento do sistema nervoso humano [3]. McCulloch e Pitts fazem neste trabalho a primeira descrição de como deveria ser um neurônio artificial e de como seria o funcionamento do mesmo. Este modelo de neurônio artificial, unidade básica de uma RNA, é hoje conhecido como neurônio de McCulloch-Pitts. O modelo proposto em 1943 consistia num sistema binário, isto é, com duas saídas possíveis. Convenientemente configurados os neurônios de McCulloch-Pitts realizam operações booleanas com E (AND) e OU (OR), com isto seria possível encadear estes neurônios artificiais de forma que o conjunto deles implementasse qualquer função booleana. Hoje sabe-se que o modelo proposto em 1943 está bastante distanciado dos neurônios biológicos, entretanto,

todos os modelos de neurônio artificial desenvolvidos posteriormente conservam as mesmas características básicas do modelo de McCulloch-Pitts.

Continuando na linha iniciada por McCulloch e Pitts, o desenvolvimento subsequente aparece no trabalho de ROSENBLATT [43] onde é proposto o *perceptron*, que pode ser considerado a primeira rede neural artificial. Além desta característica, este trabalho ainda apresenta um primeiro algoritmo para *treinamento* de uma RNA. Posteriormente, foi determinado que os perceptrons não poderiam implementar todo o tipo de função conforme era esperado inicialmente. Outro ponto importante a respeito do trabalho de Rosenblatt é a demonstração de alguns teoremas a respeito do aprendizado de uma rede neural artificial.

Um trabalho teórico importante a respeito do treinamento é o de HEBB [17]. Nele são postulados alguns dos princípios que norteiam o desenvolvimento dos algoritmos de treinamento das redes neurais artificiais.

MINSKY e PAPPERT [37] fizeram um estudo minucioso das capacidades e limitações de RNAs similares aos perceptrons e afirmaram que muitas destas limitações não poderiam ser sobrepujadas. Com isto, as pesquisas com redes neurais ficaram praticamente paralisadas por quinze anos, período no qual os desenvolvimentos em redes neurais foram esporádicos e muitos não tiveram continuidade.

O desenvolvimento que ocorreu após a proposição dos perceptrons foi feito por WIDROW e HOFF [53]. Curiosamente, o ponto principal deste trabalho não foi o tipo de rede neural proposto, linear e com apenas um neurônio artificial (ADALINE), mas sim a maneira como esta rede foi treinada. Foi apresentada a regra do delta (delta rule) para o cálculo do gradiente de uma função critério usada para a atualização dos pesos das conexões nas entradas de um neurônio artificial. A regra do delta foi modificada e posteriormente usada no algoritmo de retro-propagação dos erros, que é o mais usado para o treinamento de uma RNA.

O algoritmo de treinamento proposto por Widrow e Hoff era do tipo supervisionado. De maneira distinta ao que fora proposto por ROSENBLATT [43], neste algoritmo de treinamento havia a preocupação em encontrar o conjunto de pesos  $w_{ij}$  com o qual obtém-se o mínimo de uma função erro, no caso o erro quadrático médio entre saídas calculadas e as esperadas.

Após o já mencionado período, durante o qual as pesquisas com redes neurais estiveram paralisadas, houve um desenvolvimento de várias aplicações que usavam o algoritmo de retro-propagação do erro para treinamento da RNA. Este algoritmo foi proposto por RUMELHART, HINTON e WILLIAMS [45].

O algoritmo de retro-propagação do erro é um desenvolvimento da regra do delta proposta por WIDROW [53]. Este algoritmo foi obtido para o treinamento de redes que possuam camadas ocultas de neurônios. Segundo este algoritmo, a atualização dos pesos relativos as conexões entre neurônios de duas camadas é feita de acordo com o gradiente da função erro quadrático médio, de maneira que se encontre o mínimo desta função. Este erro é calculado para os neurônios na camada de saída da rede neural como sendo a diferença, ao quadrado, entre saídas esperadas e aquelas calculadas pela RNA. No entanto, a maneira de calcular este erro para os neurônios das camadas ocultas não é tão trivial uma vez que não se conhece a princípio quais deveriam ser os valores esperados das saídas destes neurônios. Segundo o algoritmo proposto, este inconveniente é contornado pela retro-propagação dos erros calculados na camadas de saída em direção à camada de entrada, passando então pelas camadas intermediárias.

Os trabalhos citados a seguir tratam de aplicações de redes neurais artificiais, que não a detecção de defeitos em estruturas. São apresentadas mais aplicações ligadas à identificação de sistema, tema este mais próximo a linha mestra deste trabalho.

ALGUINDIGUE et alii[1] sugerem o uso de RNAs para a detecção de danos em rolamentos de esferas a partir da análise de sinais de vibração. O objetivo da técnica proposta pelos autores é automatizar este processo de detecção de defeitos. Além da identificação dos defeitos, RNAs são usadas para condicionamento dos sinais colhidos pela instrumentação.

BARDOU e SIDAHMED [5] apresentam uma aplicação onde uma rede neural artificial foi treinada para detectar e diagnosticar vazamentos em motores e compressores à pistão. São usados sinais colhidos por transdutores de vibração para a que a rede neural artificial encontre os vazamentos. É feita uma comparação da classificação pela rede neural com outros métodos de classificação e vetores de padrões a serem classificados.

CHOW, SHARPA e HUNG [7], propõem o uso de redes neurais para a detecção de falhas em motores elétricos. O trabalho foi dividido em duas partes, na primeira são apresentados os métodos de detecção de defeitos e as RNAs. Na segunda parte são feitas considerações a respeito do projeto da rede neural, bem como do seu desempenho, estabelecimento do conjunto de dados de treinamento e dos critérios de parada do treinamento.

As RNAs também estão sendo empregadas em controle ativo de estruturas. GHABOUSSI e JOGHATAIE [13] propuseram que uma RNA fosse treinada inicialmente como um emulador, para com isto prever qual seria a resposta da estrutura a partir das respostas prévias de um sistema constituído pela estrutura e pelo atuador. Já a rede neural controladora foi treinada para aprender as relações entre as respostas do sistema e os sinais de controle enviados para o atuador. A RNA treinada foi usada no controle de uma estrutura (que simula um pequeno prédio) submetida a diferentes cargas dinâmicas de origem sísmica.

Ainda nos sistemas de controle, BAZ [4] propôs o uso de RNAs como observadores para sistemas dinâmicos. Segundo o autor, através das redes neurais é possível obter os estados de um sistema dinâmico com uso de um número pequeno de sensores. Há a comparação das RNAs com outros métodos para estimação dos estados de sistemas lineares e não-lineares.

SHOURESHI [46] faz uma compilação das técnicas usadas no que se chama de controle inteligente, procurando verificar a aplicabilidade das mesmas em situações reais. Entre estas técnicas estão as redes neurais artificiais.

GHABOUSSI, WU e GARRET [14] apresentam o emprego de redes neurais artificiais na modelagem do comportamento de materiais. O objetivo do trabalho é substituir a maneira atual de modelagem dos materiais que é feita a partir de experimentação e dos modelos matemáticos de equações constitutivas. Neste caso, uma rede neural é treinada diretamente a partir de dados obtidos de ensaios com os materiais e passam a implementar relações do tipo tensão x deformação.

Mais voltados a solução de problemas inversos há os dois trabalhos seguintes. COOK [8] propõe o uso das RNAs para determinação do carregamento não-linear que age sobre um

helicóptero. Como entradas para a RNA são usados os parâmetros de vôo da aeronave bem como a posição dos comandos do piloto.

CRUSE e STEINKÜHLER [10] empregam redes neurais para a solução dos problemas cinemáticos inverso e direto. O objetivo é obter o modelo de um braço manipulador para em seguida projetar um sistema de controle. O modelo deste manipulador consiste num mecanismo de três peças que trabalham num plano. A rede é capaz de calcular um conjunto completo de variáveis de estado (os três ângulos do manipulador e as coordenadas espaciais da ponta do mesmo) a partir de apenas parte deste conjunto de variáveis. O sistema também é capaz de encontrar os estados estáveis e uma solução geometricamente correta apenas com parte das variáveis de estado disponíveis como entrada da rede neural.

### **3.2 Redes neurais artificiais aplicadas à detecção de defeitos em estruturas**

Os trabalhos contidos nesta série restringem-se a aplicação de RNAs a detecção de defeitos em estruturas.

ELKORDY, CHANG e LEE [12] tratam de identificação de mudanças na assinatura vibracional de uma estrutura como indicação da presença de defeitos nas mesmas. As redes neurais artificiais são usadas para esta tarefa de identificação. O trabalho apresenta o treinamento de uma rede neural a partir de estados de dano gerados analiticamente desde um modelo da estrutura. A rede neural treinada foi testada com dados experimentais.

MANNING [35] indica o uso de redes neurais para detecção de defeitos em estruturas, tendo como base os dados referentes aos pólos e zeros das funções de transferência de alguns membros estruturais. As modificações devidas aos danos fazem com que a estrutura passe a ter uma função de transferência diferente e portanto com novos pólos e zeros. Deve ser lembrado que os pólos destas funções de transferências equivalem às frequências naturais.

PRATT e SANSALONE [40] apresentam o uso de redes neurais para a interpretação de sinais de eco com intuito de automatizar este tipo de ensaio. Neste caso, a rede neural artificial recebe como entrada o sinal de eco obtido da instrumentação de ensaio e fornece como saída a probabilidade de se encontrar um determinado defeito a dada profundidade.

RHIM e LEE [41] propõem o emprego de redes neurais para detecção de defeitos e identificação de estruturas, neste caso constituídas por materiais compósitos. São consideradas quatro classes de danos a estrutura, referentes a posição dos mesmos. Como entradas para a rede neural são usados os pólos da função de transferência da estrutura identificada.

UDPA e UDPA [49] empregam redes neurais para a interpretação de sinais provenientes de transdutores eletromagnéticos usados em ensaios não-destrutivos do tipo "eddy-current". Neste caso, detecta-se apenas a presença ou não de defeitos sem no entanto haver preocupação com a extensão e localização dos mesmos.

WU et alii [55] apresentam o uso de RNAs como maneira de automatizar ensaios de detecção de danos estruturais. Neste caso, a rede é treinada para indicar a perda de rigidez de elementos estruturais. Como entradas para a rede neural artificial, os autores propõem o espectro, tomado em intervalos constantes, em frequência da amplitude da aceleração medida numa estrutura submetida a carregamento sísmico. Uma desvantagem deste opção para entrada de rede neural, é o fato de que a RNA fica com muitas entradas e por conseguinte com um número elevado de pesos que precisam ser treinados. Neste trabalho são feitas apresentações das redes neurais artificiais e são fornecidas referências bibliográficas que permitem traçar o desenvolvimento de ensaios não-destrutivos, baseados nas modificações da resposta dinâmica, para estruturas de grande porte.

Um outro tipo de abordagem considera a localização de defeitos em certos elementos estruturais. YEH, KUO e HSU [56] propõem que redes neurais artificiais sejam treinadas para diagnosticar pilares de concreto pré-tensionado. Os autores analisam vários aspectos do treinamento da rede neural e da configuração da mesma. São também vistos alguns modos de constituir o conjunto de dados de treinamento da RNA.

### **3.3 Ensaios não-destrutivos baseados em modificações na resposta dinâmica de estruturas**

A proposição do uso das variações da resposta dinâmica de estruturas como ensaio não destrutivo remonta a meados da década de 70. O trabalho mais antigo encontrado foi o de VANDIVER [50] que propôs este tipo de ensaio como método para detecção de defeitos em

plataformas fixas de exploração de petróleo, estruturas complexas onde outros métodos de exame, como a inspeção visual, apresentam uma série de inconvenientes, entre eles a segurança do pessoal responsável pela análise e custo dos equipamentos empregados. Naquela época, com os recursos computacionais limitados, buscava-se apenas a avaliação do estado geral da estrutura, isto é, se existem ou não defeitos que não poderão ser ou não foram detectados por outros métodos de inspeção. O autor apresenta um caso onde foi possível avaliar a integridade de uma estrutura a partir da comparação de frequências naturais medidas antes e depois de um evento que teria gerado danos à estrutura. O autor também expõe o uso da análise estatística de energia como método para a obtenção rápida de estimativas para a resposta dinâmica da estrutura.

Os trabalhos subsequentes continuaram propondo a análise da resposta dinâmica de uma estrutura como meio para a detecção de defeitos em plataformas de exploração de petróleo. LOLAND e DODDS [33] apresentam uma metodologia para a monitoração deste tipo de estrutura operando no Mar do Norte. Neste trabalho são apresentados requisitos básicos para que um procedimento como o proposto possa ser usado com sucesso. Os autores testaram a metodologia em três estruturas diferentes e concluem pela aplicabilidade prática do que propõem. Para a detecção dos defeitos é proposta a análise do espectro de resposta da estrutura e comparação com os obtidos previamente por pessoal especializado.

WOJNAROWSKI et alii [54] usaram um modelo em elementos finitos de uma plataforma fixa para verificar os efeitos sobre a resposta dinâmica de vários tipos de falhas e modificações estruturais. Com isto seria possível obter critérios para a avaliação da integridade estrutural com uma análise e comparação da resposta da estrutura tendo em mente os efeitos verificados na simulação feita.

COPPOLINO e RUBIN [9] fizeram uma análise de sensibilidade aos danos numa estrutura de plataforma de exploração de petróleo. Esta análise foi feita a partir de modelos em elementos finitos ajustados por dados experimentais.

KENLEY e DODDS [25] também analisaram a possibilidade de detectar defeitos usando medições da resposta dinâmica. Segundo estes autores medições dos modos globais da estrutura

permitem apenas detecção de defeitos muito severos, tais como a perda total de um dos elementos estruturais, enquanto pequenos defeitos microscópicos não podem ser detectados. Para melhorar a sensibilidade do procedimento, os autores sugerem o uso de medidas para a detecção dos modos locais de vibração.

RUBIN [44] trata de problemas relativos a obtenção em campo da resposta dinâmica de uma plataforma de exploração de petróleo. O autor procura avaliar a precisão do que foi medido, sejam os modos de vibrar ou as frequências naturais. Avalia-se também o que era possível medir com a instrumentação disponível até então.

TSAI e YANG [48] apresentam uma técnica para detecção de defeitos em estruturas constituídas por materiais compósitos. O método baseia-se na obtenção de modelos matemáticos para a estrutura após a medição da resposta dinâmica. As variações dos parâmetros destes modelos matemáticos informam a respeito do defeito.

CHEN e GARBA [6] propõem um método de detecção de defeitos onde são feitas comparações entre as frequências naturais e modos de vibrar medidos para estrutura sem defeitos e aqueles da estrutura defeituosa. Também é proposto um método para cálculo das variações devidas aos defeitos na matriz de rigidez de uma estrutura, tendo as matrizes de massa e rigidez da estrutura sem defeitos e dos modos de vibrar medidos para a estrutura com defeitos. Os autores, entretanto, afirmam que o método proposto ainda necessitava de confirmação experimental, uma vez que só foram apresentadas simulações numéricas. Também afirmam ser necessário verificar a influência da precisão das medições, a relação entre número de graus de liberdades medidos e aqueles do modelo analítico e ainda a seleção do melhor parâmetro modal a ser usado para o ensaio.

RIZOS, ASPRAGATHOS e DIMARAGONAS [42] apresentam um método para a identificação de trincas transversais em estruturas que podem ser modeladas como uma viga. É também apresentado um modelo analítico para uma viga com defeito e é feita uma dedução completa para uma viga engastada. A partir de dados experimentais, medições dos deslocamentos em dois pontos da estrutura, é feito o ajuste dos parâmetros do modelo da viga com defeito obtendo-se assim a localização e profundidade do mesmo.

Outros trabalhos como o de HUNT [18] apresentam aplicações para estruturas de grande porte. Neste trabalho é apresentado um sistema para inspeção de alguns componentes do "Ônibus Espacial". Neste caso, o objetivo era a detecção apenas da existência de algum defeito em peças onde a inspeção visual ou por outros métodos não é possível. A detecção dos defeitos é feita verificando-se modificações nas frequências naturais e modos de vibrar.

PANDEY, BISWAS e SAMMAN [39] apresentam uma metodologia distinta das vistas até então. Os efeitos podem ser localizados apenas pela análise dos modos de vibrar da estrutura. Neste caso, é comparada a curvatura dos modos de vibrar para que se encontre o dano procurado. No entanto o autor aponta para o fato de que os modos de vibrar sejam obtidos com precisão para que o método funcione. Isto implica no uso de muitos sensores e medições em vários pontos da estrutura.

LIANG, HU e CHOY [30] apresentam um método para detecção de múltiplas trincas em estruturas do tipo viga. É apresentada uma teoria que relaciona mudanças nas frequências naturais com a posição e extensão dos defeitos.

ZIMMERMAN e KAOUK [57] procuram incorporar os dados modais medidos a um modelo de elementos finitos refinado da estrutura para detectar e localizar defeitos.

Além da resposta dinâmica, SOEIRO e HAJELA [47] usam resultados de ensaios estáticos como informação adicional para a detecção de defeitos. Os autores comparam as respostas da estrutura danificada com aquela obtida de um modelo analítico calibrado da estrutura não-danificada, os valores que seriam necessários para a atualização deste modelo analítico indicam a presença ou não de defeitos. A aplicação da metodologia proposta é feita para materiais compósitos.

DIRR, POPP e ROTHKEGEL [11] utilizam uma técnica para detecção de defeitos em eixos trabalhando em rotação. A técnica foi patenteada por Iman e baseia-se na comparação do espectro dos sinais de vibração do eixo obtido para diferentes posições angulares. Os autores também desenvolvem um elemento finito do tipo viga que simula um defeito em determinada posição de uma estrutura.

Para estruturas treliçadas, LIM e KASHANGAKI [31] apresentam uma metodologia segundo a qual o defeito é localizado a partir do cálculo da distância euclidiana entre os modos de vibrar medidos e os melhores autovalores obtíveis. Estes autovetores são projeções dos modos medidos no subespaço definido pelo modelo analítico refinado da estrutura e as frequências medidas.

KIM [26] desenvolveu um sistema para monitoração de grandes estruturas. Neste caso, a detecção do defeito é feita em duas etapas. Inicialmente, é identificada qual região da estrutura apresenta danos. Identificada esta região, um modelo mais preciso permite determinar qual componente estrutural está danificado.

LYON [34] propõe o uso dos zeros das funções de transferência medidas em diferentes pontos da estrutura como parâmetro para a detecção de defeitos. Segundo o autor, os zeros destas funções são bastante sensíveis à modificações dos modos de vibrar da estrutura que por sua vez são bastante sensíveis aos defeitos presentes numa estrutura. A metodologia proposta é aplicada em um componente soldado de um automóvel.

Dentro desta categoria de trabalhos existem alguns que apenas modelam o defeito e detalhadamente apresentam os efeitos destes defeitos sobre as estruturas. GOMES e MONTALVÃO e SILVA [15] fizeram uma série de medições numa viga com o objetivo de determinar os efeitos de um entalhe sobre o comportamento dinâmico da estruturas. Foram ensaiadas vigas com o defeito em diferentes posições e profundidades em relação a altura da viga.

JOSHI e MADHUSUDHAN [19] obtiveram expressões analíticas para as frequências naturais de vigas com um defeito, sob diferentes condições de vínculo. KRAWCZUK e OSTACHOWICZ [28] desenvolveram um modelo para as vibrações laterais de uma viga com uma trinca que se fecha quando a estrutura vibra.

Outra categoria de trabalhos procura dar indicações mais genéricas para os métodos de detecção de defeitos usando vibrações, sem se limitarem a aplicação direta de um dos diferentes algoritmos disponíveis. NATKE e CEMPEL [38] analisaram diferentes parâmetros com respeito a suas sensibilidades a detecção de defeitos. Também são discutidos fatores como ordem dos modos, condições de contorno e tamanho das falhas que influenciam a detecção dos defeitos.

ZIMMERMAN et alii [58] aplicam a teoria de perturbação de mínima ordem para a detecção de defeitos, mas não se restringem a aplicação direta do algoritmo. Os autores propõem que antes da aplicação direta do algoritmo proposto, seja feita uma análise do comportamento da estrutura no que diz respeito a variação de sua resposta dinâmica com a presença de defeitos. O objetivo desta análise é indicar quais modos são mais afetados por defeitos em determinadas regiões, sendo que estes modos serão usados no procedimento proposto.

#### 4. DETECÇÃO DE DEFEITOS EM ESTRUTURAS

O desenvolvimento da tecnologia levou a criação de diferentes técnicas para detecção de defeitos em estruturas, estas técnicas ficaram conhecidas como *ensaios não-destrutivos* (END). A função destes ensaios é determinar se peças, estruturas e equipamentos apresentam alguma falha. O nome é devido ao fato de que estes elementos uma vez ensaiados não terminam sendo inutilizados.

Deve-se primeiramente definir o que se entende por detecção de defeitos, localização e avaliação da extensão destes. Por detecção entende-se a determinação da integridade ou não da estrutura, isto é, um método de detecção de defeitos deve indicar se uma estrutura possui ou não defeitos. Já um método de localização de defeitos deve ser capaz de fornecer o local onde está um defeito detectado e isto pode ser feito através de coordenadas no espaço, por exemplo, a posição do defeito em relação ao comprimento total de uma peça ou indicando que o defeito está num determinado componente da estrutura, previamente identificado. Já a avaliação da extensão do defeito consiste em fornecer a informação a respeito da gravidade deste último. Esta informação pode ser a profundidade de uma trinca, ou entalhe, ou ainda a perda de rigidez de um determinado elemento de uma estrutura.

Ressalta-se que alguns métodos funcionam apenas para a detecção de defeitos enquanto outros tem como objetivo detecção, localização e determinação da extensão destes. As limitações estão ligadas também ao número de defeitos que são tratados por cada método. Alguns conseguem apresentar a maior parte dos danos presentes numa única peça enquanto outros são capazes de fornecer a localização de um único defeito apenas em um estrutura complexa.

Uma outra distinção a ser feita é quanto ao caráter de ensaio que pode ser *local* ou *global*. No primeiro caso, o ensaio é executado no local onde há suspeita de existência de um defeito. No outro tipo, a estrutura é ensaiada como um todo e o método indica se há ou não defeitos e pode fornecer a localização e extensão dos mesmos.

Um exemplo de ensaio local é o de líquido penetrante, usado para a localização de trincas em regiões críticas de peças. Em estruturas complexas o uso de métodos locais fica inviabilizado devido a necessidade da realização do ensaio em diversas partes da estrutura, em outros casos

certos pontos da estrutura são de difícil acesso, o que também dificulta a realização de qualquer tipo de ensaio local. Nestes casos é recomendado o emprego de métodos globais.

Outras diferenças entre os ENDs diz respeito a natureza do defeito que pode ser detectado (trincas, entalhes, defeitos de fundição, etc.) e quanto a sensibilidade do método em relação a extensão dos defeitos. Por exemplo, alguns ENDs só permitem a detecção de defeitos macroscópicos enquanto outros permitem a localização de defeitos que não são visíveis à olho nu.

#### 4.1 Ensaios não-destrutivos baseados na análise dinâmica de estruturas

Dentre os métodos globais de ensaio não-destrutivo estão aqueles baseados nas variações da resposta dinâmica de uma estrutura. A validade do uso destas variações em ENDs vem do fato de que defeitos, de maneira genérica afetam rigidez, amortecimento e, em menor escala, a massa da estrutura. Esta influência faz com que a resposta dinâmica da estrutura (Frequências naturais e modos de vibrar) seja modificada, pois esta resposta é diretamente ligada aos parâmetros massa, rigidez e amortecimento.

Seja, por exemplo, um sistema discreto (com número finito de graus de liberdade) sujeito a oscilações livres. O seu comportamento dinâmico é dado por:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{0} \quad (4.1.1)$$

Sem perda de generalidade pode-se desprezar o amortecimento resultando em:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{0} \quad (4.1.2)$$

A resposta do sistema pode ser considerada como sendo uma composição de termos variáveis e não variáveis com o tempo:

$$\mathbf{u} = \Phi_i \text{sen}(\varpi_i t) \quad (4.1.3)$$

Sendo que  $\Phi_i$  corresponde ao modo de vibrar  $i$  e  $\varpi_i$  à frequência natural associada a este modo.

Substituindo 4.1.3 em 4.1.2 e separando os termos não dependentes do tempo obtém-se:

$$\mathbf{K}\Phi_i - \varpi_i^2 \mathbf{M}\Phi_i = \mathbf{0} \quad (4.1.4)$$

Neste caso a equação 4.1.4 corresponde a um problema de autovalores e autovetores, sendo os primeiros correspondentes a  $\varpi_i^2$  e os outros a  $\Phi_i$ . Fica clara, então, a influência da massa e rigidez sobre a resposta dinâmica, pois  $\varpi_i$  e  $\Phi_i$  são obtidos a partir de  $\mathbf{K}$  e  $\mathbf{M}$ .

Este tipo de ensaio começou a ser desenvolvido em meados da década de 70 para a detecção de defeitos em elementos submersos de plataformas *offshore* para exploração de petróleo [9, 25, 33, 44, 50, 54].

Devido às restrições existentes naquela época, este tipo de ensaio indicava apenas a existência ou não de defeitos, sem fornecer a localização dos mesmos nem a extensão destes.

As primeiras proposições [50] estabeleciam que este tipo de END poderia ser usado inicialmente como um método auxiliar às inspeções visuais realizadas por mergulhadores em plataformas *offshore*, isto é, caso alguma anomalia fosse detectada então procedia-se o mergulho para exame visual da estrutura, do contrário esta operação não deveria ser realizada. A avaliação da presença, ou não do defeito era feita através da comparação das frequências naturais da estrutura medidas em determinados períodos de tempo. Na época da publicação dos primeiros artigos não havia previsão para que este processo de comparação fosse automatizado.

Desde estas primeiras experiências com este tipo de END fica clara a necessidade de um modelo relativamente preciso para o comportamento da estrutura. A função primeira deste modelo era fornecer informações a respeito do comportamento da estrutura quanto a variação de sua resposta dinâmica (frequências e modos naturais) quando da presença de defeitos. Desta forma, foi possível fazer as primeiras estimativas da sensibilidade das variações na resposta dinâmica em relação a presença de defeitos [50] confirmando a possibilidade de uso destas variações para detecção destes.

O desenvolvimento posterior envolveu o uso das variações nos modos de vibrar para a detecção de defeitos. Isto é devido ao fato dos modos de vibrar serem mais sensíveis a presença de defeitos do que as frequências naturais.

Além dos modos de vibrar outros parâmetros também foram empregados para a detecção de defeitos entre eles está a distribuição da energia cinética pela estrutura [6]. Entretanto, todas estas novas metodologias necessitam modelos complexos e precisos para a estrutura e a medição

em vários pontos da mesma para que os procedimentos e algoritmos apresentados sejam eficazes na detecção e na avaliação da extensão dos defeitos.

Segundo ZIMMERMAN [58] os procedimentos baseados em análise da resposta dinâmicas de estruturas podem ser divididos em 5 categorias:

- Uma primeira categoria é composta pelos procedimentos que empregam algoritmos para modificação de matrizes, obtidas a partir de modelos de elementos finitos, tendo como base a resposta dinâmica medida. Estes métodos são variantes daqueles propostos para a otimização de modelos de elementos finitos.
- Outros métodos para a atualização das matrizes de massa e rigidez da estrutura em ensaio são baseados na sensibilidade das derivadas dos parâmetros modais em relação aos parâmetros físicos da estrutura (massa e rigidez). Estas derivadas são usadas no cálculo de parâmetros que fazem com que o modelo da estrutura passe a fornecer as respostas medidas (frequências e modos naturais). Este tipo de procedimento é encontrado em alguns programas comerciais de elementos finitos, como o "MSC-NASTRAN" para otimização de modelos, entretanto o seu uso implica em grande esforço computacional [31].
- A aplicação de métodos adaptados da teoria de controle também já foi tentada neste tipo de END. O objetivo é determinar o controlador que seria necessário para que a estrutura, neste caso um modelo daquela sem defeitos, tivesse um comportamento igual àquele medido. Este pseudo-controlador é em seguida convertido em modificações no modelo inicial.
- Uma quarta categoria é baseado na *teoria de perturbação de mínima ordem* [58].
- Por fim existem aqueles métodos que não se preocupam em fazer modificações num modelo já existente para a estrutura. Neste grupo está incluída a aplicação de redes neurais artificiais.

Vê-se claramente que os métodos desenvolvidos para ENDs, baseados na análise dinâmica de estruturas, dividem-se entre aqueles que se preocupam em solucionar o problema de identificação (problema inverso) usando para tal diferentes técnicas e as proposições que

procuram contornar a resolução do problema inverso. No primeiro grupo estão as quatro categorias iniciais acima citadas.

Os métodos englobados no primeiro grupo tiveram um desenvolvimento considerável graças aos progressos obtidos nos ensaios de análise modal. Com isto, foram desenvolvidos algoritmo para otimização de modelos que foram estendidos para a detecção e localização de defeitos.

A extensão destes métodos, apesar de atrativa do ponto de vista de desenvolvimento teórico, obriga que certas condições sejam satisfeitas para que estes métodos seja eficazes como END. Por exemplo, um método que aplica a teoria de perturbação de mínima ordem requer que os parâmetros medidos sejam livres de ruído, as medições sejam feitas para todos os graus de liberdade do modelo de elementos finitos e o número de modos medidos deve ser superior a ordem das matrizes de perturbação [58]. As duas primeiras condições são certamente muito difíceis de serem satisfeitas na prática, não existem medições sem ruído e se o modelo de elementos finitos for razoavelmente completo, a segunda condição resulta também impossível de ser atendida. Os outros métodos apresentam condições semelhantes que se não são satisfeitas em sua totalidade fazem com que o desempenho desses métodos seja degradado. ZIMMERMAN [58] afirma que é possível permitir que algumas destas condições sejam relaxadas fazendo-se uma análise prévia do comportamento da estrutura e da resposta desta afetada por defeitos. Assim, apesar da perda de precisão é ainda possível empregar o método proposto, o que fica impedido é a automação do processo de detecção de defeitos.

Os métodos englobados no segundo grupo procuram contornar todas estas desvantagens, muitas delas inerentes a resolução do problema inverso. Entretanto, não são muitos os métodos que podem substituir os outros procedimentos com vantagens.

Uma aproximação usa as matrizes de flexibilidade da estrutura sem defeitos e da estrutura com os mesmos. Uma multiplicação destas duas matrizes permite a obtenção de indicadores da presença de danos [32].

Esta última metodologia é ainda ligada aos métodos de otimização de modelos. Dentre as metodologias que rompem com esta linha está aquela proposta por WU et alii [55] que emprega redes neurais para o processamento da resposta de uma estrutura sujeita a carregamento sísmico.

#### **4.2 ENDS empregando variações na resposta dinâmica e RNAs**

Dentre os métodos propostos que evitam a solução do problema inverso em ENDS baseados na variação da resposta dinâmica de uma estrutura, está o uso de redes neurais artificiais.

O uso das RNAs para o problema em questão é indicado devido as características de reconhecimento de padrões e aproximação de funções típicas das redes neurais.

A primeira proposição foi feita por WU [55] e usava como entradas para a rede neural o espectro da resposta dinâmica de uma estrutura submetida a carregamento sísmico. A saída da rede neural fornece a fração de rigidez de alguns elementos estruturais. Trata-se portanto, de um método de localização e avaliação da extensão do dano. Os modelos estruturais adotados são simples e o propósito do trabalho dos autores era verificar a aplicabilidade das RNAs ao problema em questão.

Caberá a RNA o aprendizado da relação causa e efeito que existe entre a presença dos defeitos e a variação da resposta dinâmica de maneira que não será necessária a obter explicitamente uma equação que represente este comportamento da estrutura. Esta primeira etapa equivale ao treinamento da RNA.

Terminado o treinamento da rede neural o próximo passo será a verificação da generalização do aprendizado através de testes. São apresentados à rede novos dados, não usados no treinamento, e dependendo dos resultados obtidos considera-se que a rede é genérica ou não. Os processos de treinamento e teste são repetidos até que seja encontrada a rede ideal para o caso estudado.

Embora seja atrativo, como método computacional, o uso de redes neurais requer que alguns requisitos sejam satisfeitos. Entre estes está a necessidade de um conjunto de dados para treinamento da RNA. Este conjunto contém a resposta da estrutura sujeita a diferentes estados de

dano. Caso este arquivo seja muito pequeno, não será possível treinar a rede neural, ou então, a rede uma vez treinada não terá capacidade de generalização.

Outro problema a ser tratado é a escolha das entradas da rede neural e a maneira como os defeitos serão representados. Como ilustração para o primeira pode-se retomar o trabalho de WU [55]. Neste caso, a escolha das entradas gerou uma rede neural com muitos neurônios e que necessitava de um conjunto de dados de treinamento com muitos elementos não excluindo ainda a difícil convergência do processo de treinamento. Estes são algumas das questões que devem ser analisadas antes do emprego propriamente dito das RNAs.

Da mesma forma que as entradas, a representação dos defeitos deve ser feita com critério para a garantia de um bom desempenho das RNAs. Esta representação dos defeitos deve levar em conta desde a função de transferência dos neurônios da RNA até o número destes na camada de saída. Este número não deve ser muito elevado pois, do contrário, gera-se uma rede muito complexa de difícil treinamento.

Apesar destes inconvenientes citados, o uso de redes neurais artificiais é vantajoso sob diferentes pontos de vista. Neste caso, é possível treinar uma RNA para que defeitos sejam localizados e tenham suas extensões avaliadas tendo como informação disponível apenas algumas das frequências naturais. Logo, a medição a ser feita na estrutura é mais simples do que aquela que é realizada em outros métodos de detecção de defeitos. Do ponto de vista computacional as redes neurais também são vantajosas. De maneira distinta de outras proposições, não são feitas operações que possam gerar instabilidades nos processos numéricos, tais como inversões de matrizes. As redes neurais são mais simples do que os métodos usados para a resolução do problema inverso. Outra vantagem reside no fato das RNAs possibilitarem uma automatização dos ensaios não destrutivos pois, uma vez treinadas e testadas, dispensam a análise das medições por parte de um especialista. Com isto fica vantajosa a sua aplicação onde deve-se monitorar constantemente o estado da estrutura.

#### 4.2.1 Pré-processamento das entradas.

Basicamente, propõe-se o uso das frequências naturais para a detecção de defeitos. Entretanto, o uso direto destes valores como entrada da RNA pode não ser vantajoso. Um pré-processamento das frequências naturais pode melhorar consideravelmente o desempenho da RNA.

Não há uma regra que indique qual o melhor pré-processamento para cada caso. Nos modelos testados, ficou claro, entretanto, que o uso direto das frequências naturais não é a melhor opção. As características das funções de transferência das RNA explicam tal fato. No caso da função logística, por exemplo, as entradas devem estar no intervalo  $[-1, 1]$  para que as características de ganho variável deste tipo de função seja usada ao máximo, pois do contrário há risco de saturação do processo de treinamento. Como para as estruturas estudadas, bem como para aquelas encontradas na bibliografia, as frequências naturais estão fora deste intervalo há uma clara necessidade de executar um pré-processamento nos valores de entrada.

Este primeiro pré-processamento é necessário para que as redes neurais possam ser empregadas de maneira correta e pode ser feito a partir de uma normalização dos valores medidos para as frequências naturais. Entretanto, existem ainda outros parâmetros, calculados a partir das frequências naturais (e apenas destas) que podem melhorar o desempenho da rede neural.

O primeiro parâmetro testado é a diferença ponderada entre as frequências naturais da estrutura sem defeitos e aquelas da estrutura defeituosa. Este parâmetro é calculado para cada modo de vibrar e denominado  $nfr_i$ :

$$nfr_i = \frac{\omega_{oi} - \bar{\omega}_i}{\omega_{oi}} \quad (4.2.1)$$

Inicialmente, a idéia de aplicação deste parâmetro era transformar as frequências naturais em entradas compatíveis com as funções de transferência das RNAs. Entretanto, este parâmetro possibilitou um melhor desempenho destas.

Outro parâmetro empregado foi denominado  $nfm_i$  e consiste numa normalização do anterior  $nfr_i$ :

$$nfm_i = \frac{nfr_i}{\sum_i nfr_i} \quad (4.2.2)$$

KAMINSKI [22] demonstra que este último parâmetro depende apenas da localização dos defeitos, sendo portanto indicado para ensaios que visam apenas a localização e não a avaliação da extensão destes. Já o primeiro parâmetro,  $nfr_i$ , não só depende da localização como também da variação da rigidez e portanto, da extensão do dano quando este é modelado como uma perda de rigidez. Logo, este parâmetro deve ser usado quando se quer um ensaio para localização e avaliação da extensão dos defeitos.

Outro parâmetro, ainda a ser testado, é a diferença entre a frequência natural de um modo de vibrar para outro.

Nota-se que, apesar dos parâmetros levarem a resultados diferentes, do ponto de vista do ensaio nada foi modificado com a introdução destes. As grandezas medidas continuam sendo algumas das frequências naturais. Por outro lado, a introdução destes parâmetros não constitui-se em complicador pois estes podem ser calculados facilmente sendo as frequências naturais disponíveis.

Ainda quanto as entradas, deve-se limitar o número desta ao número de modos medidos. Caso para algum elemento do conjunto de treinamento não seja disponível uma das frequências naturais, este elemento deve ser retirado do conjunto.

## 5. EXEMPLOS DE APLICAÇÃO

Para ilustração do método proposto são apresentados dois exemplos. Em ambos, há aplicação de diferentes conceitos vistos neste trabalho, desde o uso de diferentes parâmetros de entrada da rede neural até formas distintas de representação do defeito.

Embora tenham o mesmo objetivo inicial, os dois exemplos são bastante distintos. Num deles é modelada uma estrutura mais simples onde o que se procura é verificar a aplicabilidade do método proposto a detecção de defeitos em componentes que podem ser elementos de uma estrutura mais complexa ou ainda elementos de máquinas como, por exemplo, um eixo de transmissão. Já o segundo exemplo, trata de uma estrutura mais complexa, onde o principal objetivo é identificar qual ou quais elementos estão danificados.

No primeiro exemplo é ainda feito um estudo da variação da resposta dinâmica da estrutura para diferentes posições e profundidades de defeitos. Nesta parte ainda também é analisado o comportamento dos parâmetros  $NFR_i$  e  $NFRN_i$  segundo as mesmas condições de defeitos. Com isto, é possível verificar a influência das características de um defeito (posição e profundidade) nestes dois parâmetros propostos como entradas de uma RNA.

Nos dois exemplos as frequências naturais foram obtidas a partir de modelos em elementos finitos, sendo que o programa usado foi o ANSYS REV 5.0. No primeiro os resultados foram comparados com os resultados obtidos por GOMES E SILVA [15]. Verificou-se que o modelo proposto tinha concordância com os dados obtidos experimentalmente. A principal razão para o uso de modelos em elementos finitos foi a possibilidade de obtenção rápida de valores de frequências naturais para muitos casos de dano à estrutura. Desta forma, foi possível obter com facilidade os conjuntos de treinamento das RNAs. Entretanto, deixa de ser feito o ensaio propriamente dito uma vez que os dados não foram obtidos experimentalmente. Ficam desconsideradas as imprecisões dos processos de medição e há o fato dos defeitos simulados nos modelos de elementos finitos não provocarem exatamente o mesmo efeito, que os seus congêneres reais, nas estruturas.

### 5.1 Exemplo - 1: Viga livre - livre

O primeiro exemplo consiste numa viga com condições de contorno livre-livre [2], tendo comprimento igual a 0,8 m e seção transversal quadrada de lado 0,03 m. Estrutura semelhante a esta foi ensaiada por GOMES E SILVA [15], onde o defeito foi considerado como sendo um entalhe passante ao longo de toda a largura da seção transversal (ver fig. 5.1.1)

O material de composição desta viga é o aço carbono com as seguintes propriedades:

- Módulo de elasticidade:  $E = 2,10 \times 10^{11} \text{ N/m}^2$
- Coeficiente de Poisson:  $\nu = 0,3$
- Massa específica:  $\rho = 7860 \text{ kg/m}^3$

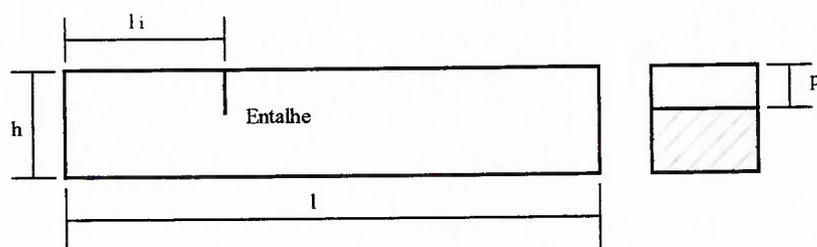


Fig. 5.1.1 - Viga livre-livre com entalhe

Pode ser questionável considerar o defeito como sendo um entalhe, principalmente quando se pensa num defeito como sendo uma trinca, por exemplo, uma que se propaga num fenômeno de fadiga. Este tipo de defeito é muito menor que o entalhe e não provoca as mesmas modificações nas frequências naturais da viga. No entanto, a referência [15] foi a única encontrada onde há uma listagem completa das medidas de frequência natural, o que possibilitou uma melhor calibração do modelo de elementos finitos usado.

No modelo empregado o entalhe foi simulado por um elemento de comprimento pequeno, em comparação ao total da viga, e com propriedades geométricas da seção transversal distintas do restante da estrutura.

Foram simulados entalhes ocupando posições desde uma das extremidades até a metade da viga em intervalos de 0,005m. As profundidades do entalhe são tomadas como sendo as seguintes frações da altura total da seção transversal: 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7. Desta forma tem-se o entalhe representado por dois números.

As saídas da rede neural serão justamente as frações do comprimento,  $Z_i/L$  e as da altura total  $p_i/h$ . Neste caso foram consideradas as cinco primeiras frequências naturais para cálculo dos parâmetros  $nfr_i$  e  $nfn_i$ , usados como entradas da RNA.

Este modelo foi usado para a avaliação das modificações da resposta dinâmica da estrutura devido aos defeitos e também do comportamento dos parâmetros  $nfr_i$  e  $nfn_i$ .

### 5.1.1 - Modificações na resposta dinâmica

O objetivo inicial desta análise foi entender como um defeito pode influenciar a resposta dinâmica de uma estrutura, ainda que simples como a deste exemplo.

Buscou-se identificar e separar os efeitos da posição e profundidade do defeito sobre cada um dos modos de vibrar, para cada um dos parâmetros propostos. Para tanto, os resultados obtidos a partir dos modelos em elementos finitos foram tabelados e são apresentados a seguir na forma de gráficos. (As tabelas com os valores das frequências naturais, usadas na construção destes, estão no anexo I).

Os gráficos podem ser agrupados em três categorias: Categoria I - *parâmetro* x  $p_i/h$ , para cada posição do defeito; categoria II - *parâmetro* x  $Z_i/L$ , para cada profundidade de trinca e categoria III - *parâmetro* x  $Z_i/L$ , para cada modo de vibrar. Por *parâmetro* entende-se  $nfr_i$  e  $nfn_i$ .

Para o parâmetro  $nfr_i$  verifica-se que algumas frequências naturais não sofrem variação considerável quando o defeito ocupa determinadas posições, sendo que estas correspondem a "nós" nos modos de vibrar, isto é, pontos da estrutura onde não há deformação naquele determinado modo de vibrar. No anexo I estão as formas modais para esta estrutura, bem como o valor das frequências naturais.

Os gráficos da categoria I permitem avaliar qual dos modos é mais afetado por determinada posição de trinca. Por exemplo, no gráfico 5.1.3 é possível observar que o segundo modo é o mais afetado para um defeito localizado na posição  $Z/L = 0,3$ . Cabe também notar que, alguns modos de vibrar passam a ser modificados de forma mais intensa a partir de certas profundidades.

Gráficos da categoria - I  
Parâmetro nfr

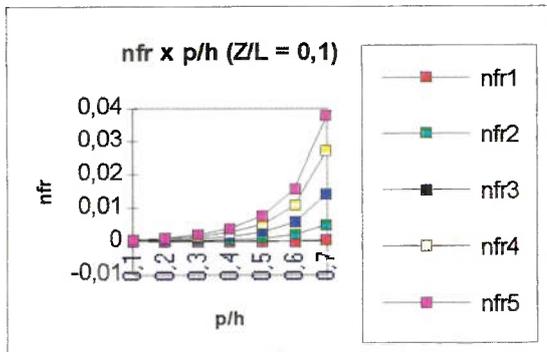


Gráfico - 5.1.1

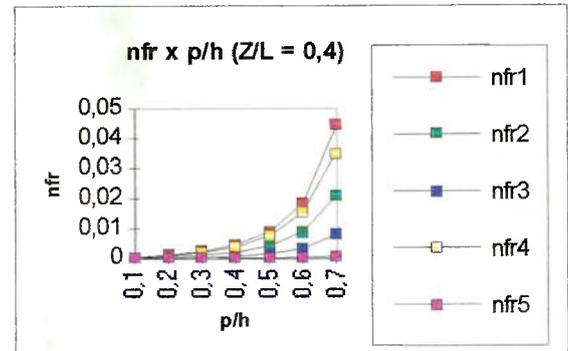


Gráfico - 5.1.4

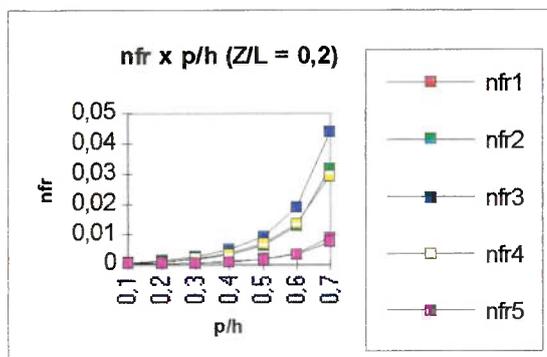


Gráfico - 5.1.2

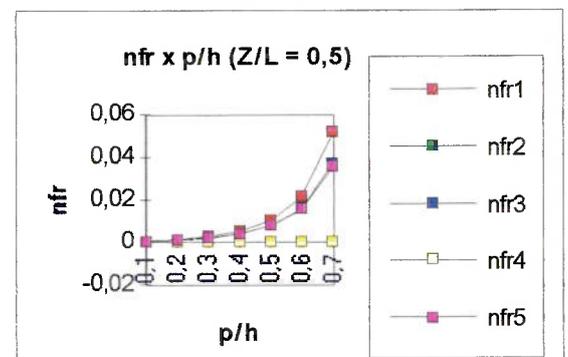


Gráfico - 5.1.5

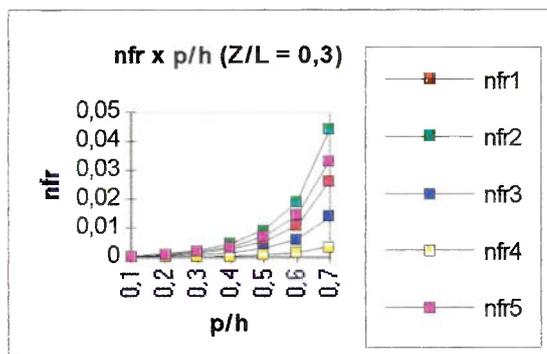


Gráfico - 5.1.3

## Gráficos da categoria - II Parâmetro nfr

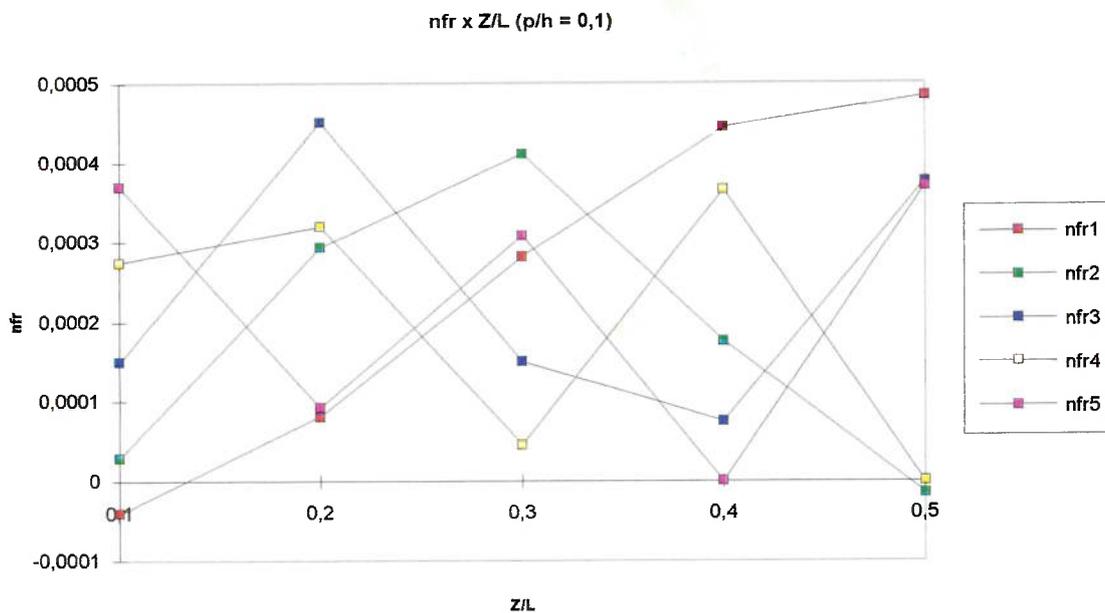


Gráfico - 5.1.6

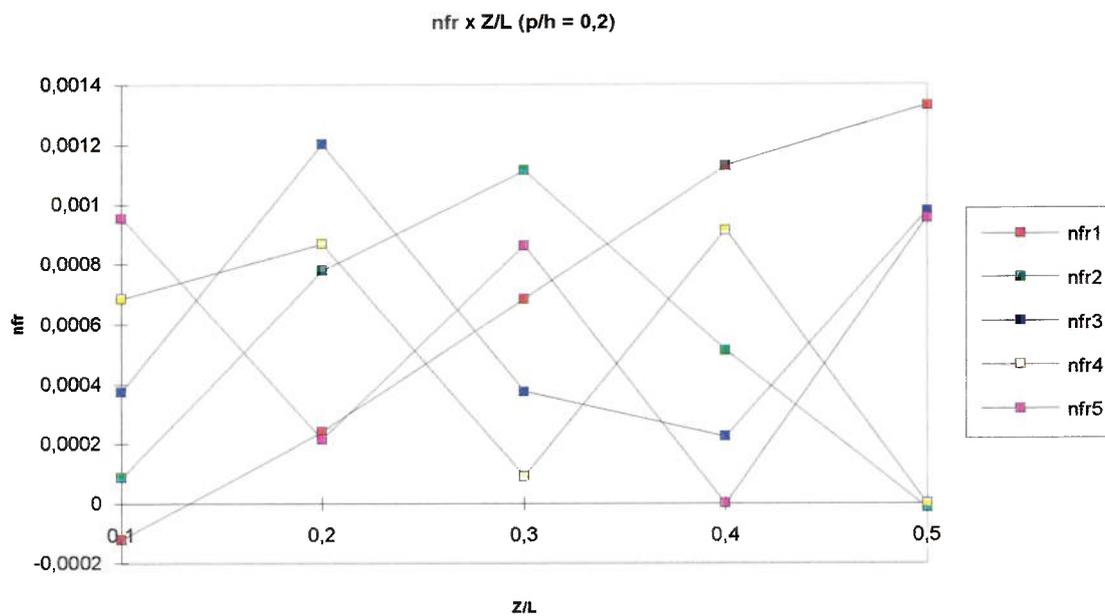


Gráfico - 5.1.7

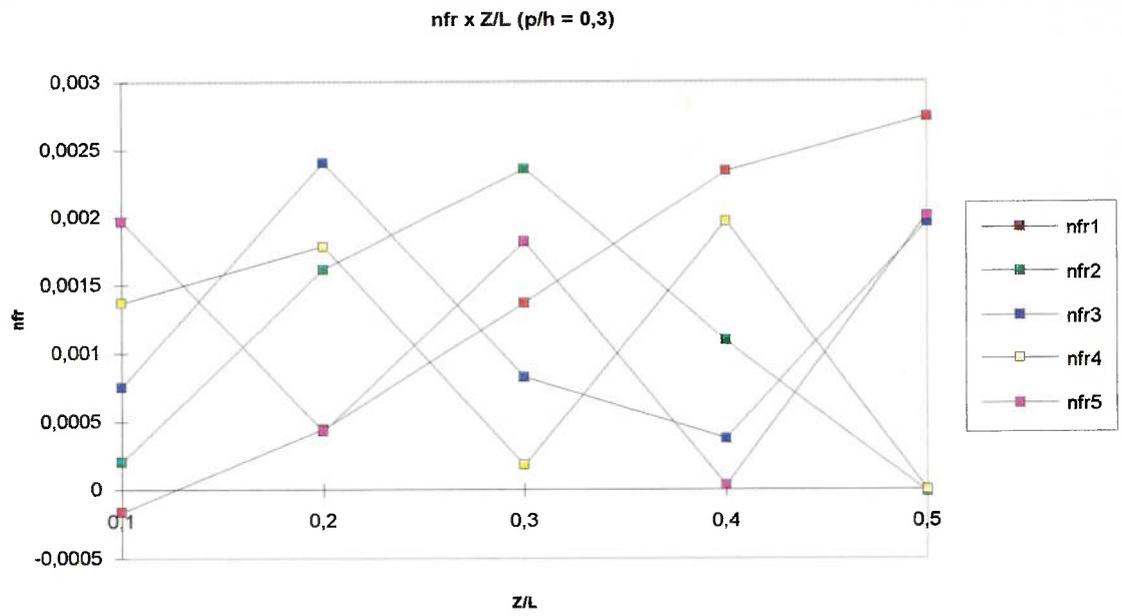


Gráfico - 5.1.8

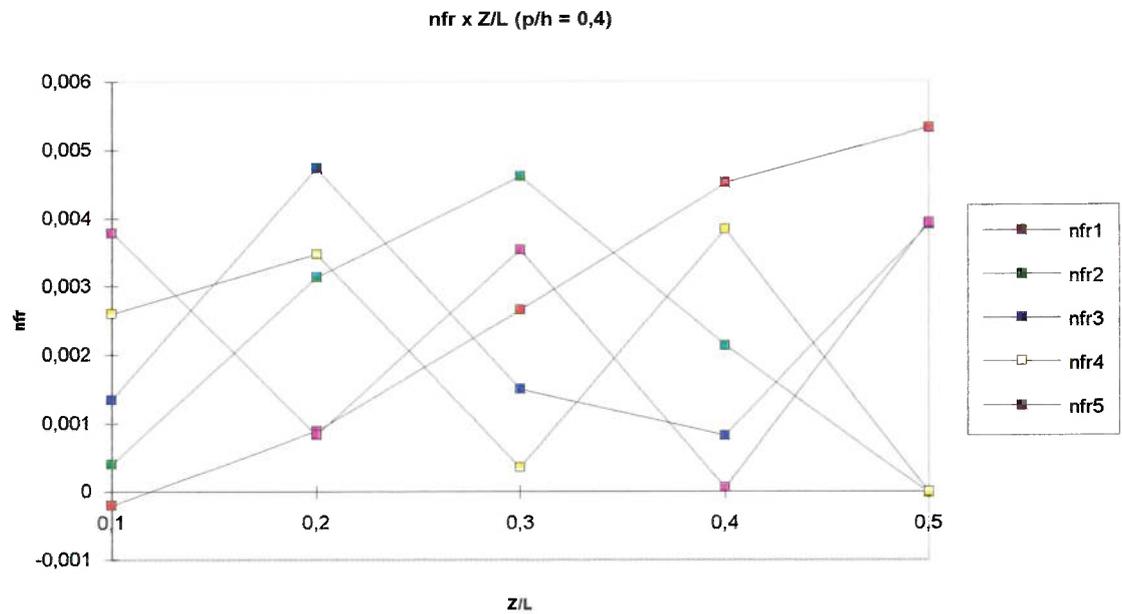


Gráfico - 5.1.9

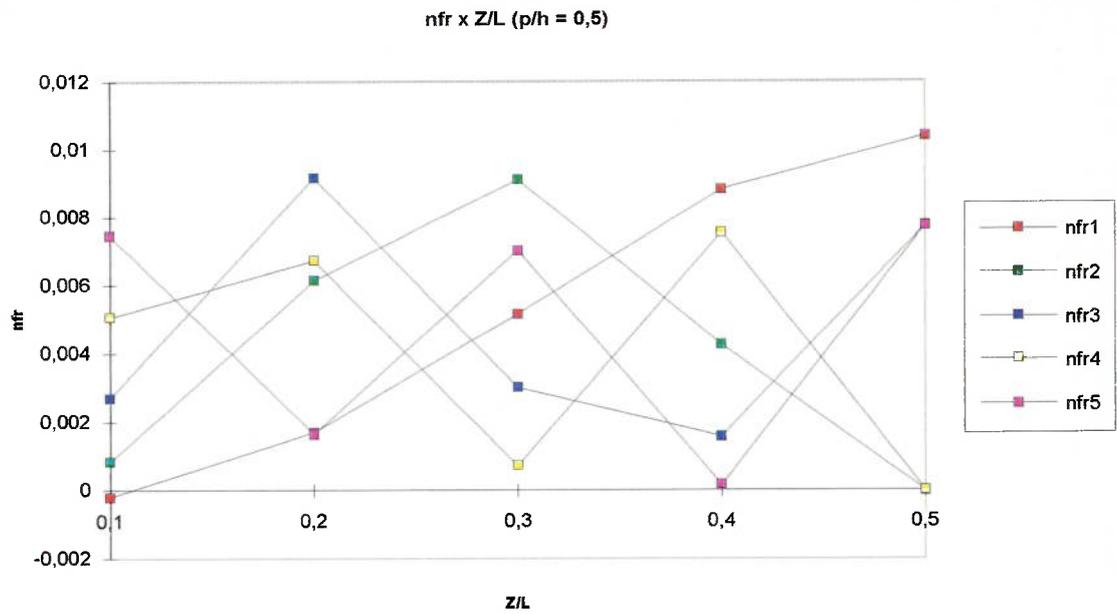


Gráfico - 5.1.10

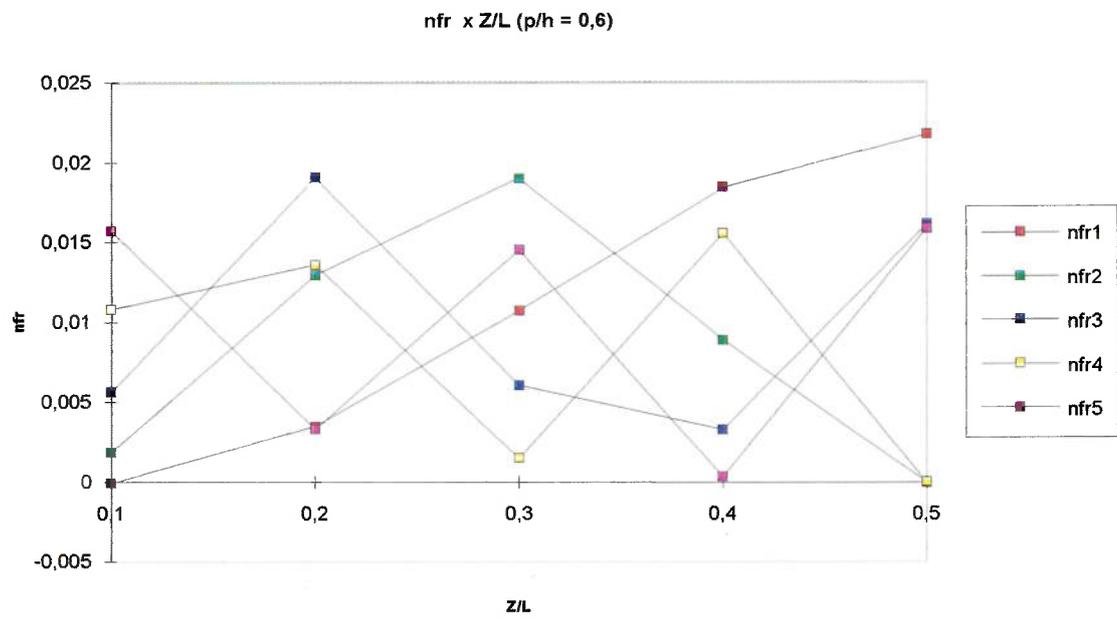


Gráfico - 5.1.11

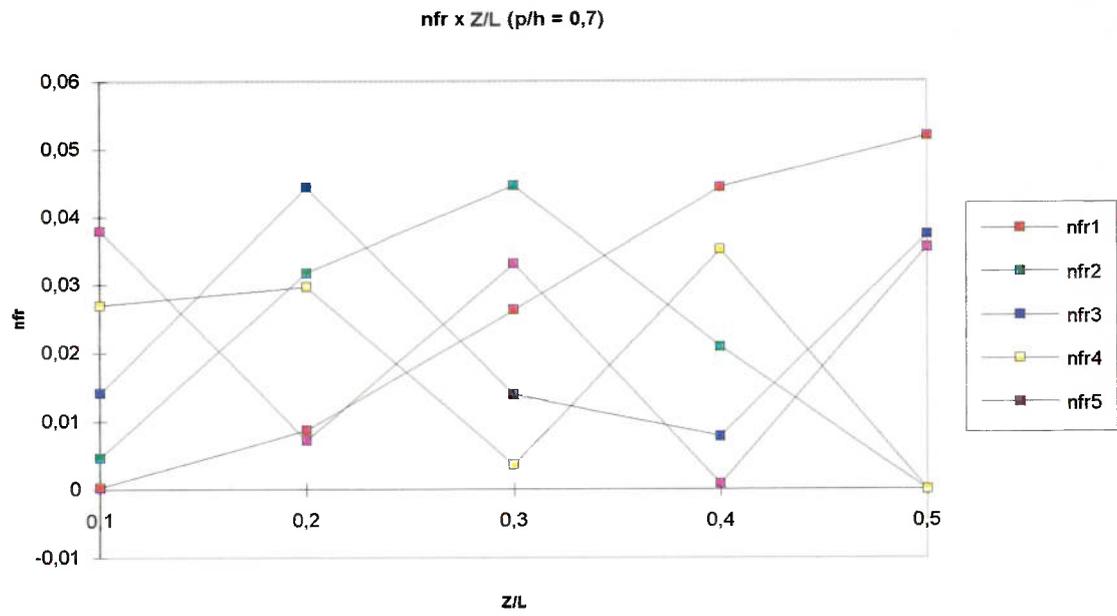


Gráfico - 5.1.12

**Gráficos da categoria - III**  
**Parâmetro nfr**

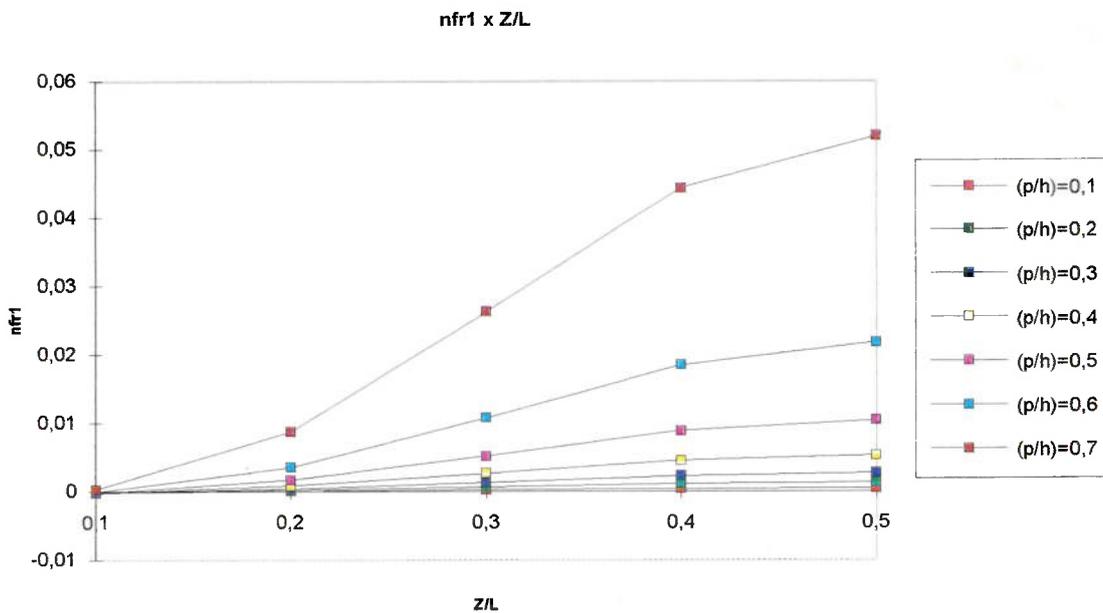


Gráfico - 5.1.13

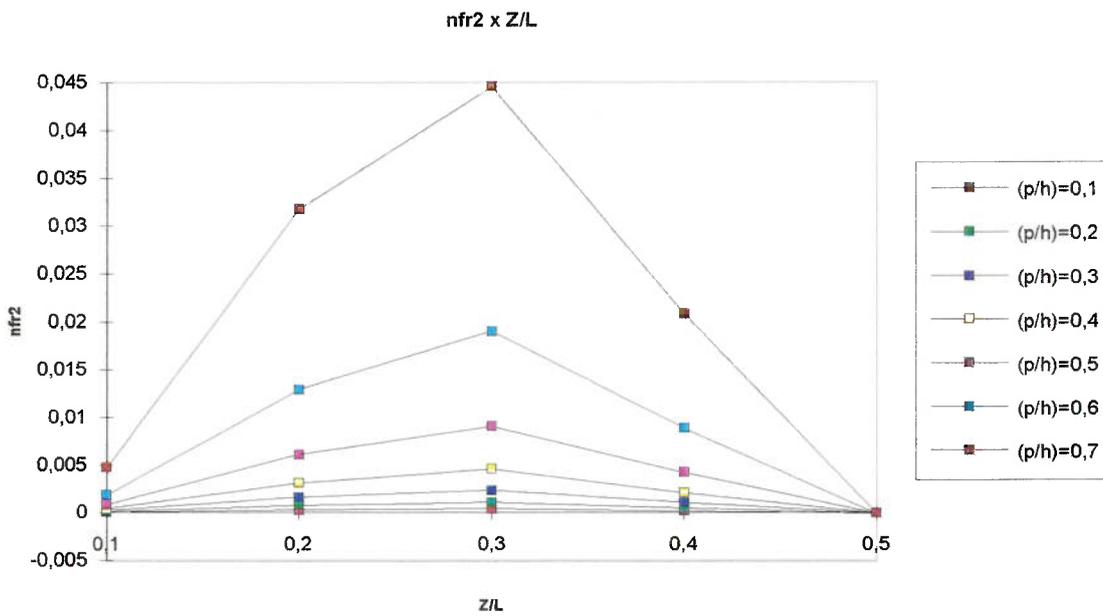


Gráfico - 5.1.14

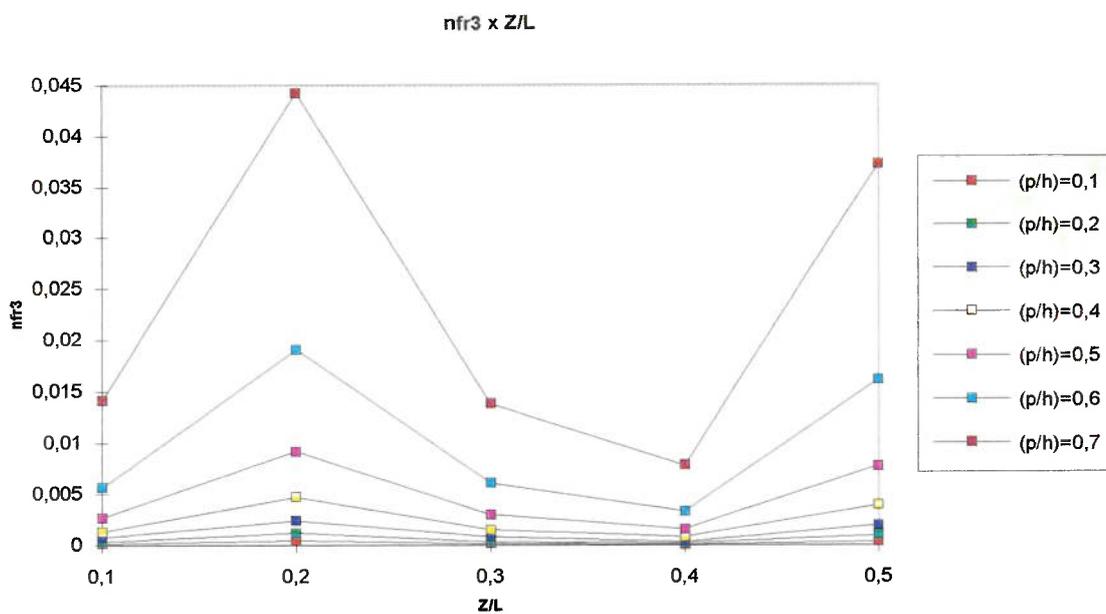


Gráfico - 5.1.15

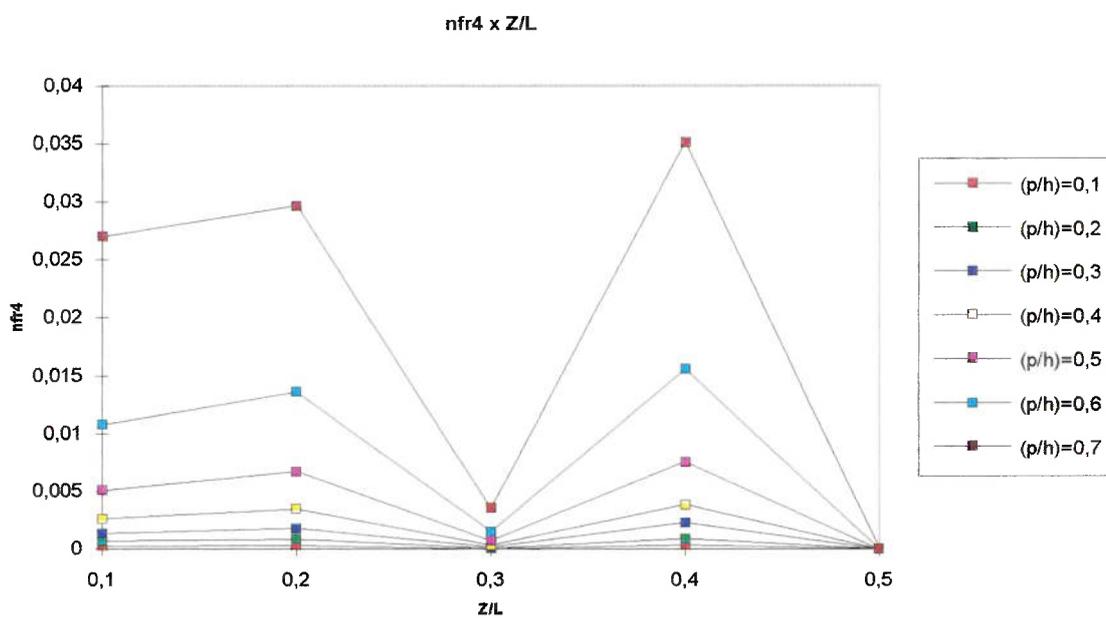


Gráfico - 5.1.16

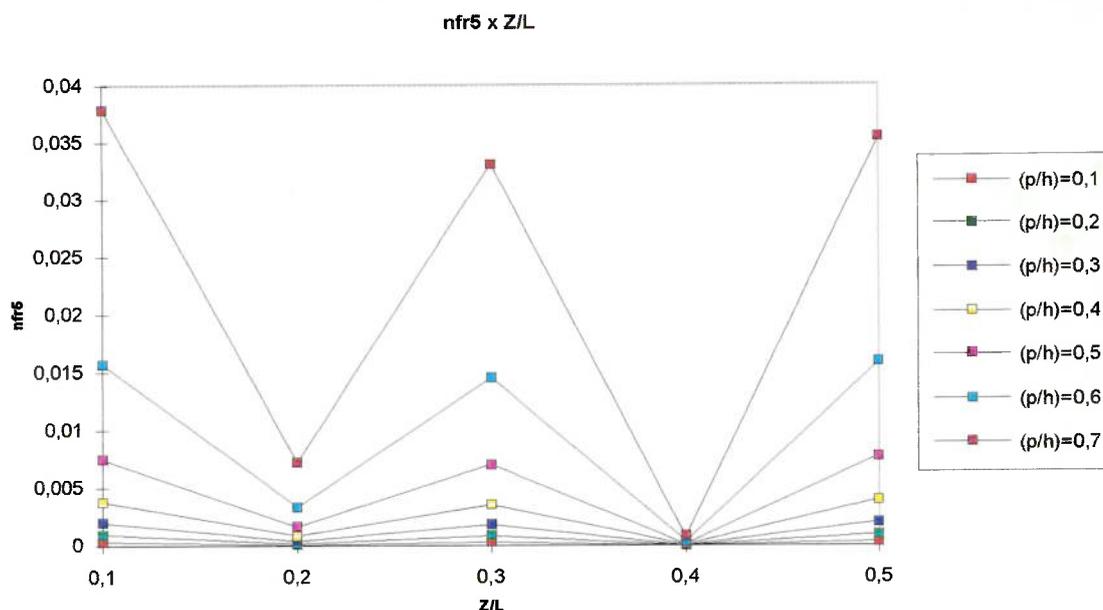


Gráfico - 5.1.17

A forma das curvas nos gráficos da categoria II não é substancialmente afetada pelo efeito da profundidade do defeito. A tendência destas curvas é acompanhar as formas dos modos de vibrar. Para maiores profundidades de trinca há um aumento em valor absoluto de  $nfr_i$ .

Os gráficos do tipo III ilustram o efeito da posição e profundidade do defeito sobre um determinado modo de vibrar. É também possível verificar que, para algumas posições do defeito, o valor de  $nfr_i$  aproxima-se de zero, como já foi mencionado. Este efeito é pouco influenciado pela profundidade.

Deste último conjunto de curvas pode-se inferir que para uma dada frequência natural e fixada uma posição para o defeito, o efeito da profundidade consiste em aumentar o valor de  $nfr_i$ , desde que a posição do defeito não coincida com um "nó" do modo de vibrar.

Nota-se também que, da mesma forma que alguns modos de vibrar não sofrem qualquer influência em determinadas posições do defeito, outros modos passam a ser mais afetados que os outros, não importando a sua ordem, isto é, um modo de ordem inferior pode ser mais afetado que outro de ordem superior, para um defeito com as mesmas características.

Este último efeito é bastante influenciado pela profundidade, sendo que alguns modos passam a ter este comportamento apenas para profundidades de defeito superiores a determinado valor. Isto pode ser observado em alguns dos gráficos da categoria I, quando há cruzamento nas curvas de cada modo.

Os gráficos para o parâmetro  $nfm_i$  demonstram que este parâmetro não é influenciado na prática pela profundidade do entalhe. Isto pode ser verificado em qualquer um dos grupos de gráficos, mas, de maneira mais evidente, a partir dos gráficos da categoria I e III. Nos primeiros, as curvas aproximam-se de retas paralelas ao eixo das abscissas ( $p/h$ ) conforme seria esperado. Já nos gráficos da categoria três há superposição quase que total das curvas obtidas para cada modo de vibrar. Assim fica evidente que, este parâmetro só poderá ser usado para a localização dos defeitos e nunca para determinação da profundidade, pois não contém nenhuma informação a respeito do efeito da profundidade do entalhe sobre a resposta dinâmica da estrutura.

É também possível concluir destes gráficos que, da mesma forma que para o parâmetro  $nfr_i$ , alguns modos não são influenciados por defeitos em determinadas posições e portanto  $nfm_i$  será nulo, logo não trará nenhuma informação para a localização do defeito caso este esteja nesta determinada posição. Por exemplo, no gráfico 5.1.22 é possível verificar que os segundo e quarto modos ( $nfm_2$  e  $nfm_4$ , respectivamente) praticamente não são afetados por um defeito localizado na posição  $Z/L = 0,5$ . Neste mesmo gráfico é também possível verificar que dois modos distintos são igualmente afetados por defeitos numa determinada posição. Neste exemplo, em particular, os modos 3 e 5 ( $nfm_3$  e  $nfm_5$ , respectivamente) apresentam este comportamento. Estes últimos efeitos são particularmente mais evidentes quando se considera o parâmetro  $nfm_i$  que não é influenciado pelo efeito da profundidade do entalhe.

Gráficos da categoria - I  
Parâmetro nfrn

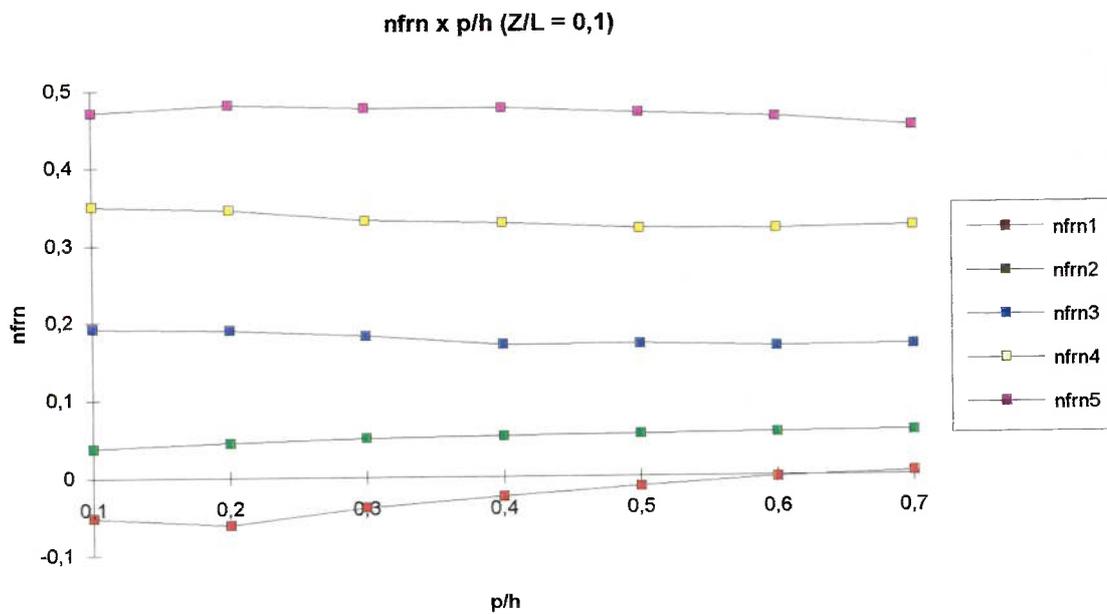


Gráfico - 5.1.18

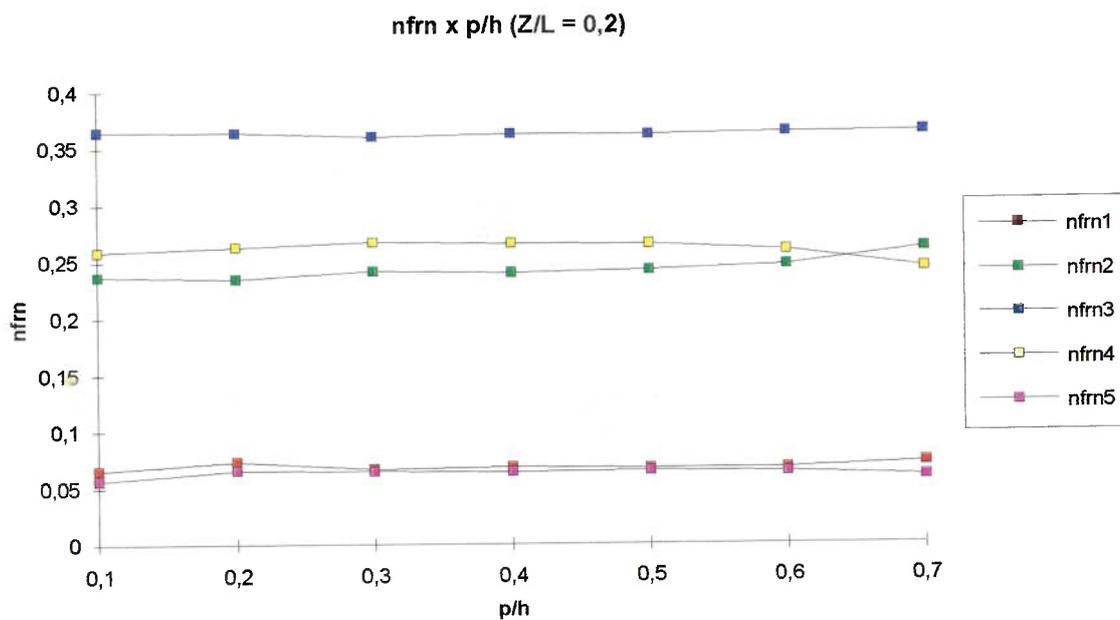


Gráfico - 5.1.19

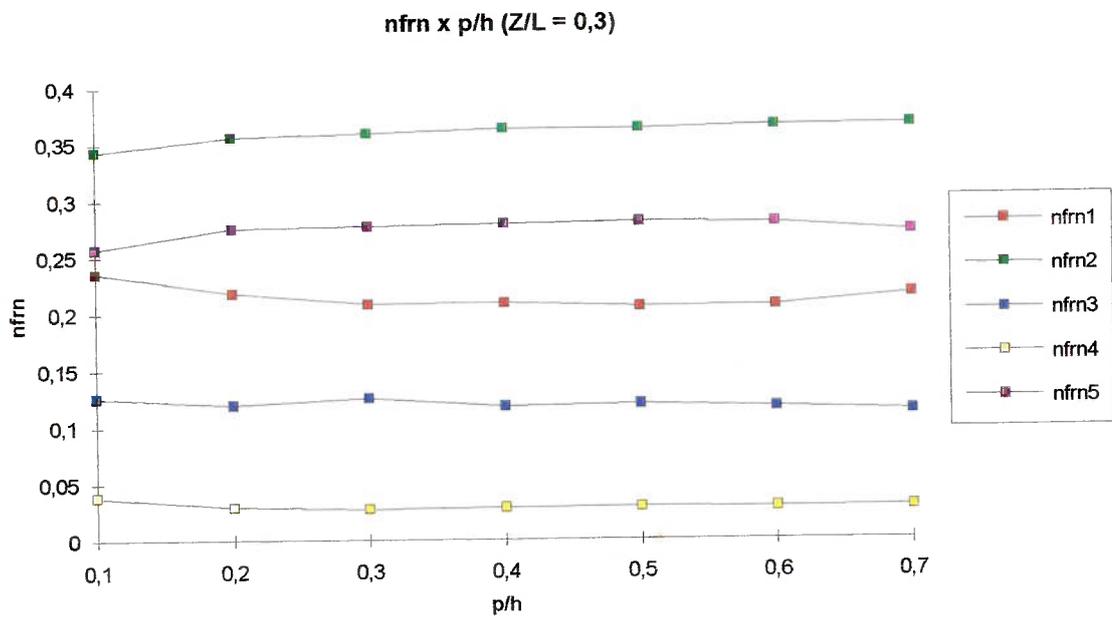


Gráfico - 5.1.20

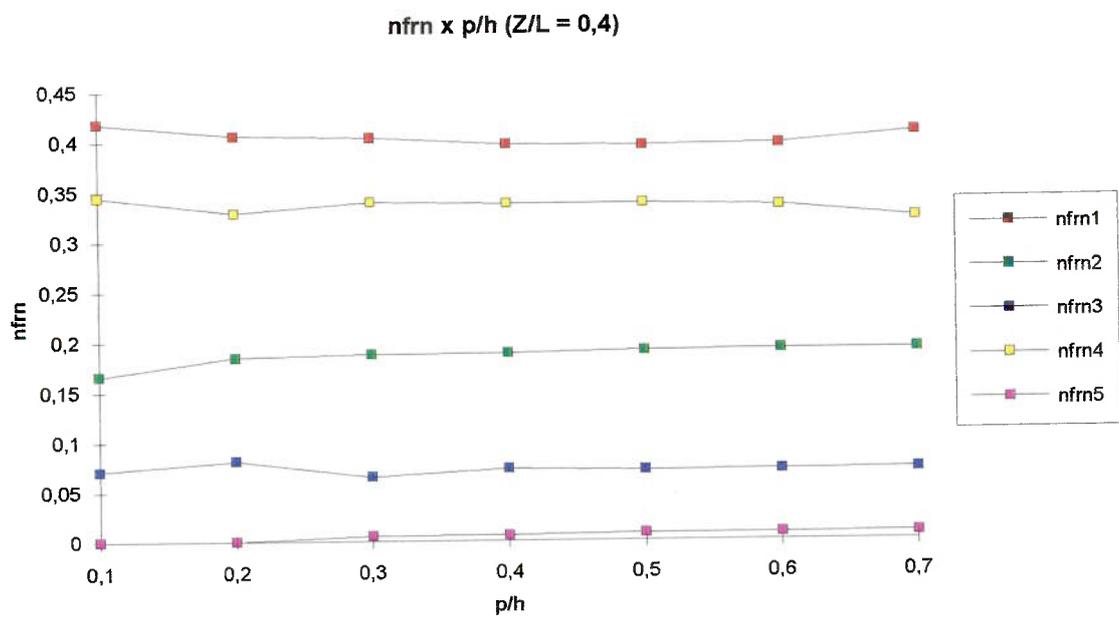


Gráfico - 5.1.21

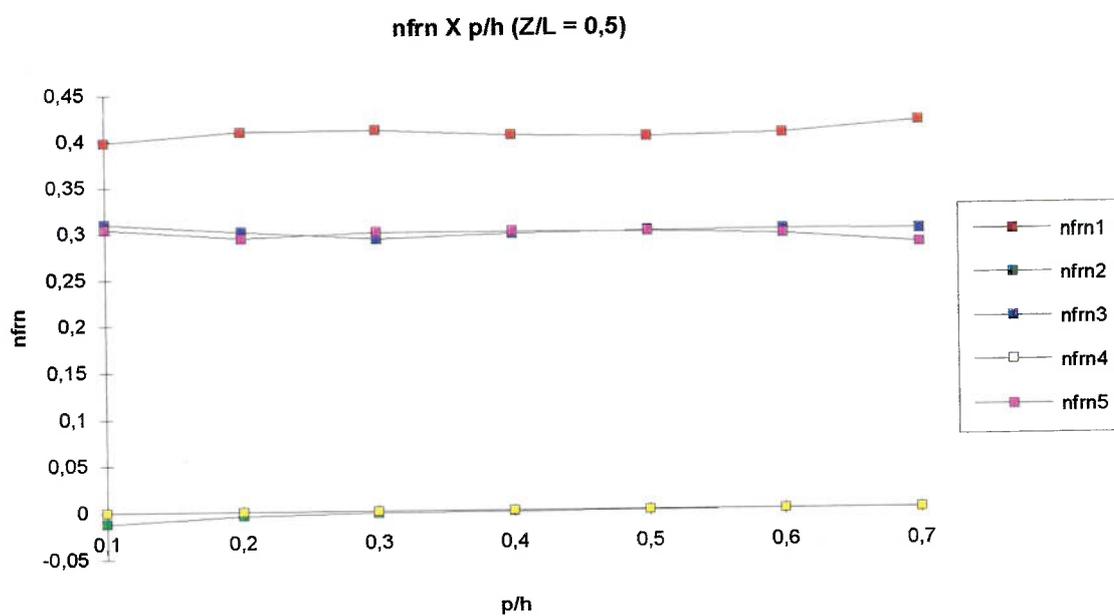


Gráfico - 5.1.22

Gráficos da categoria - II  
Parâmetro nfrn

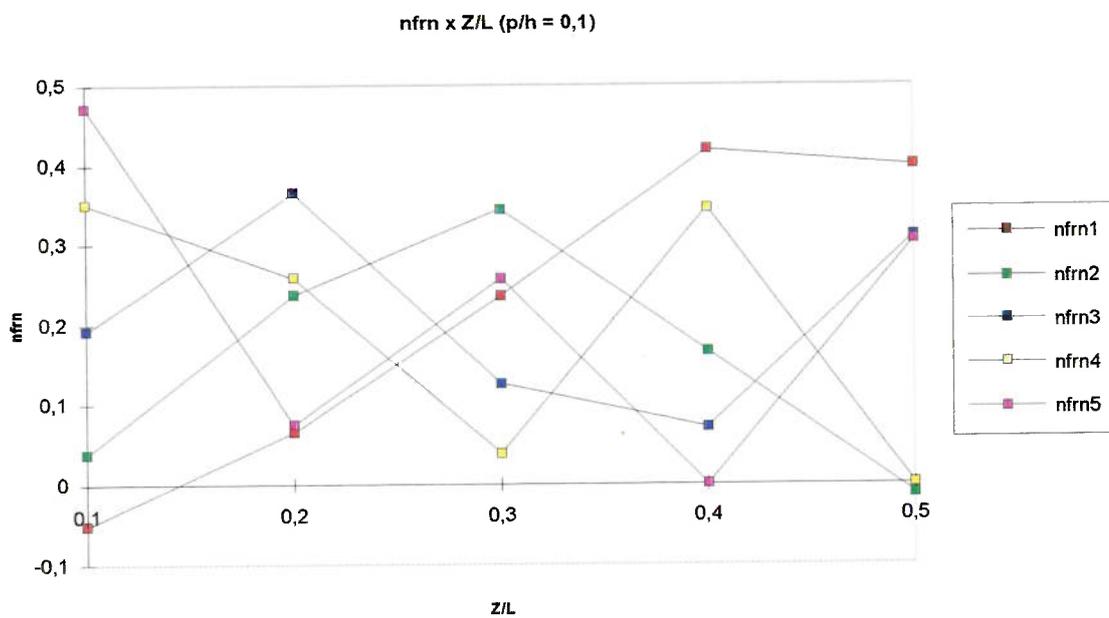


Gráfico - 5.1.23

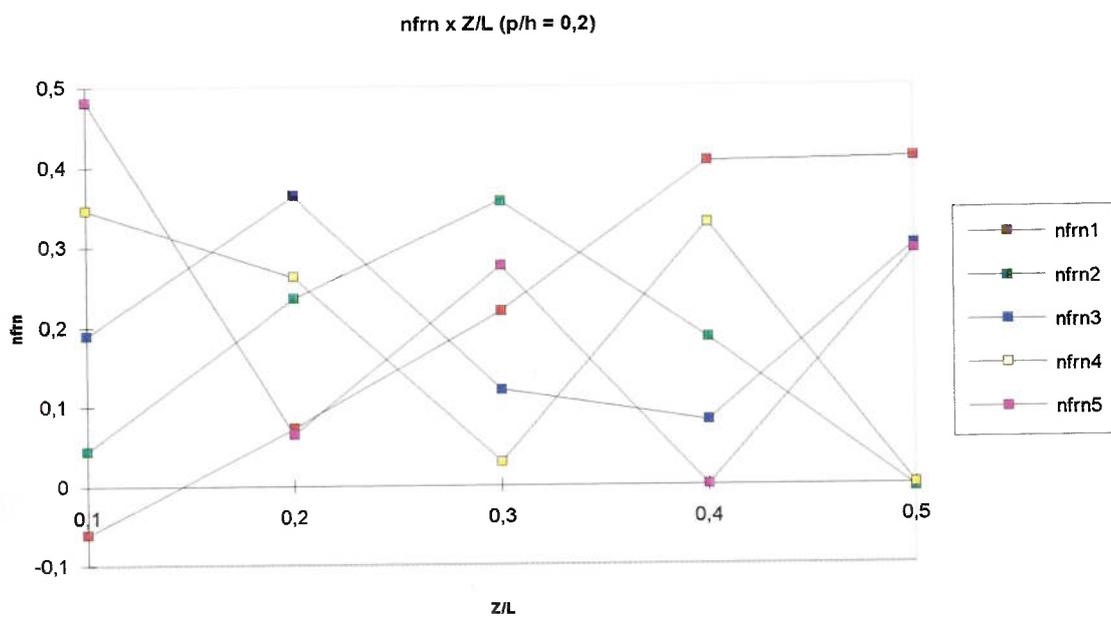


Gráfico - 5.1.24

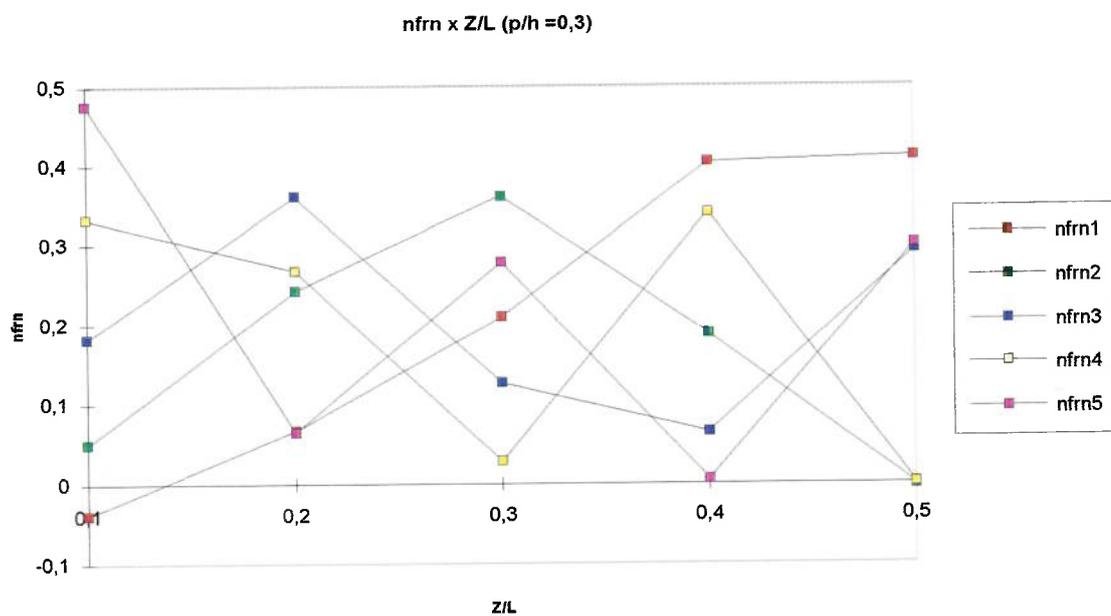


Gráfico - 5.1.25

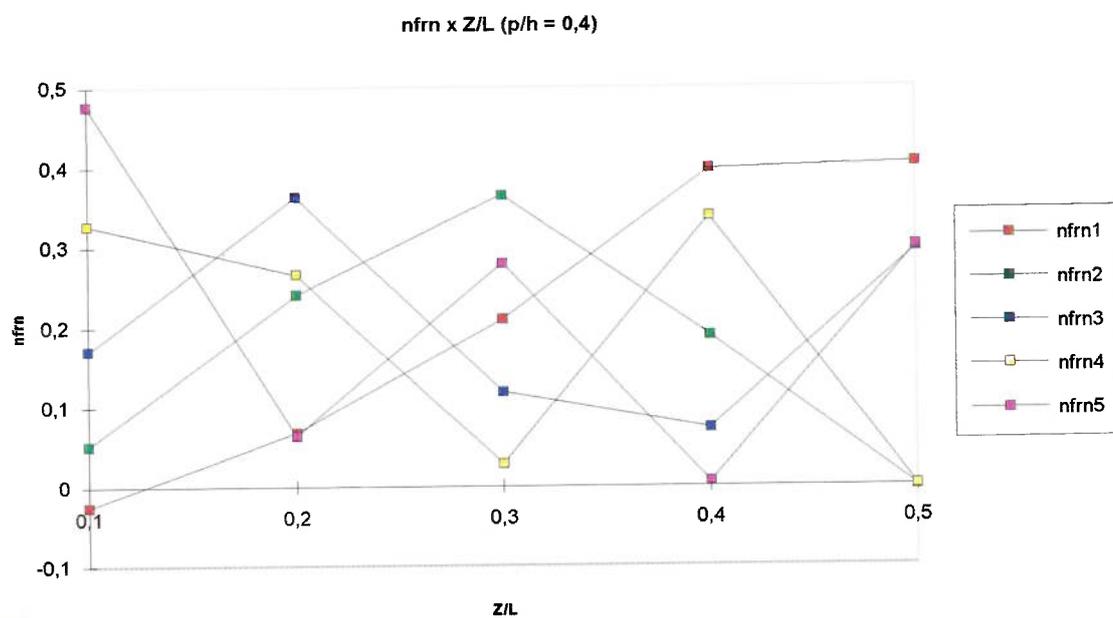


Gráfico - 5.1.26

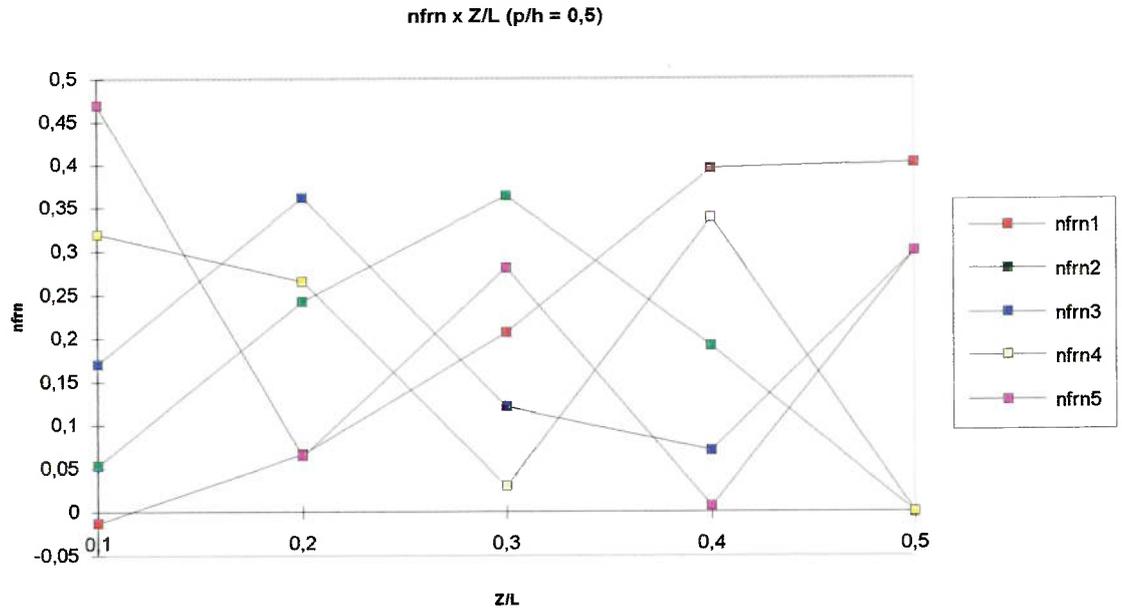


Gráfico - 5.1.27

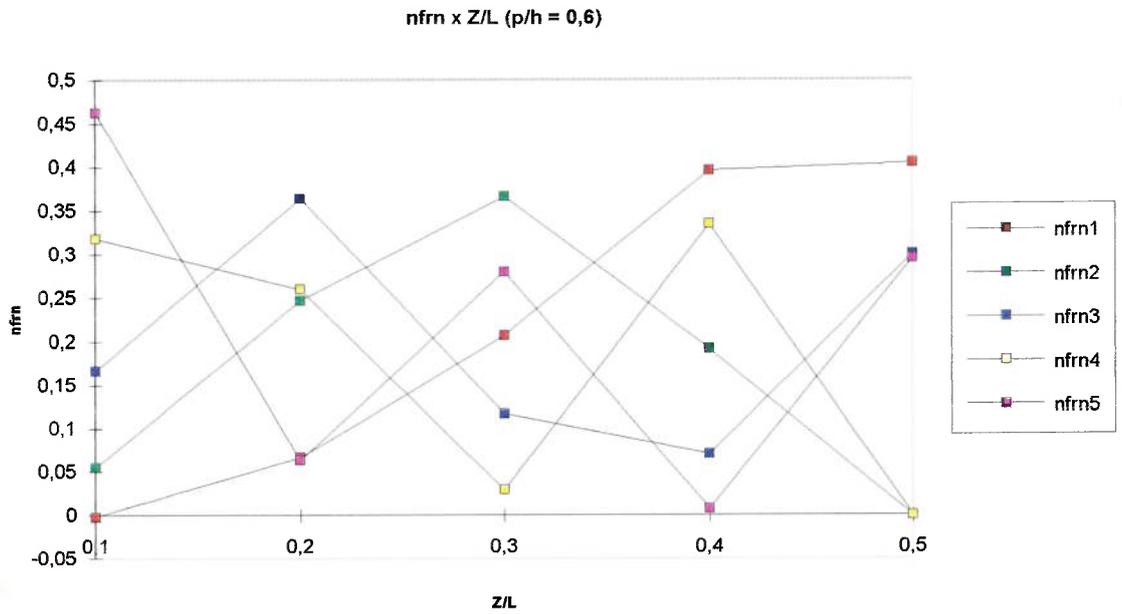


Gráfico - 5.1.28

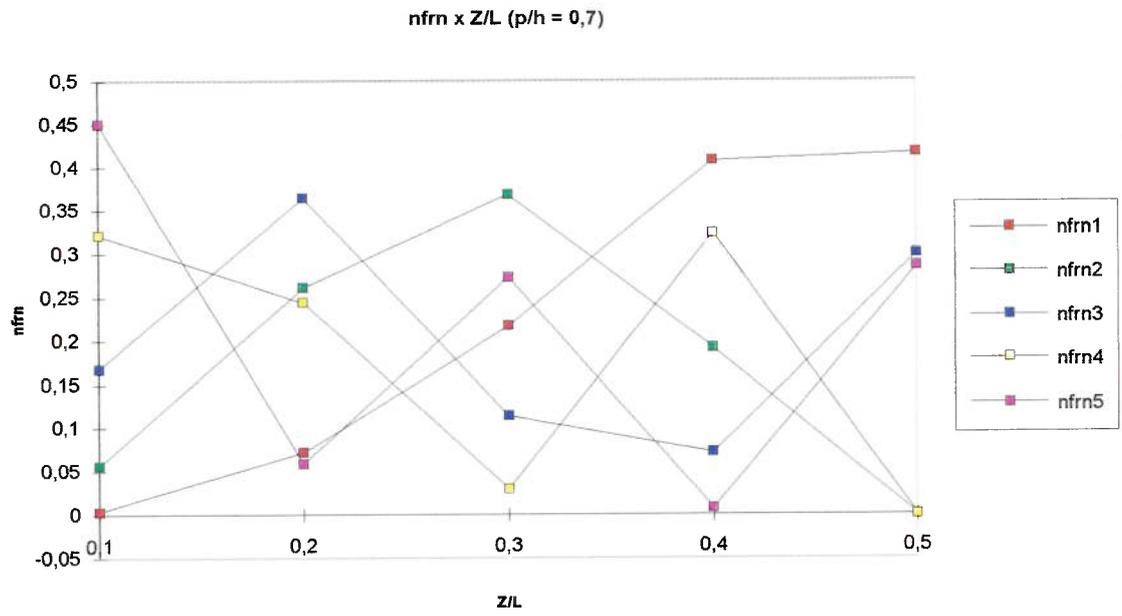


Gráfico - 5.1.29

**Gráficos da categoria - III  
Parâmetro nfr**

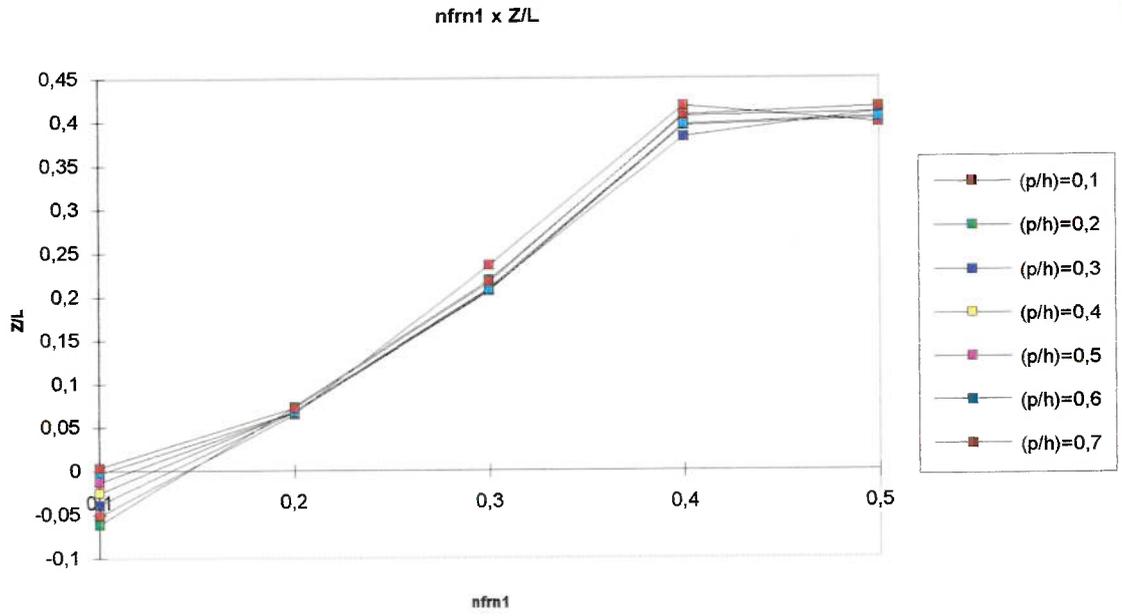


Gráfico - 5.1.30

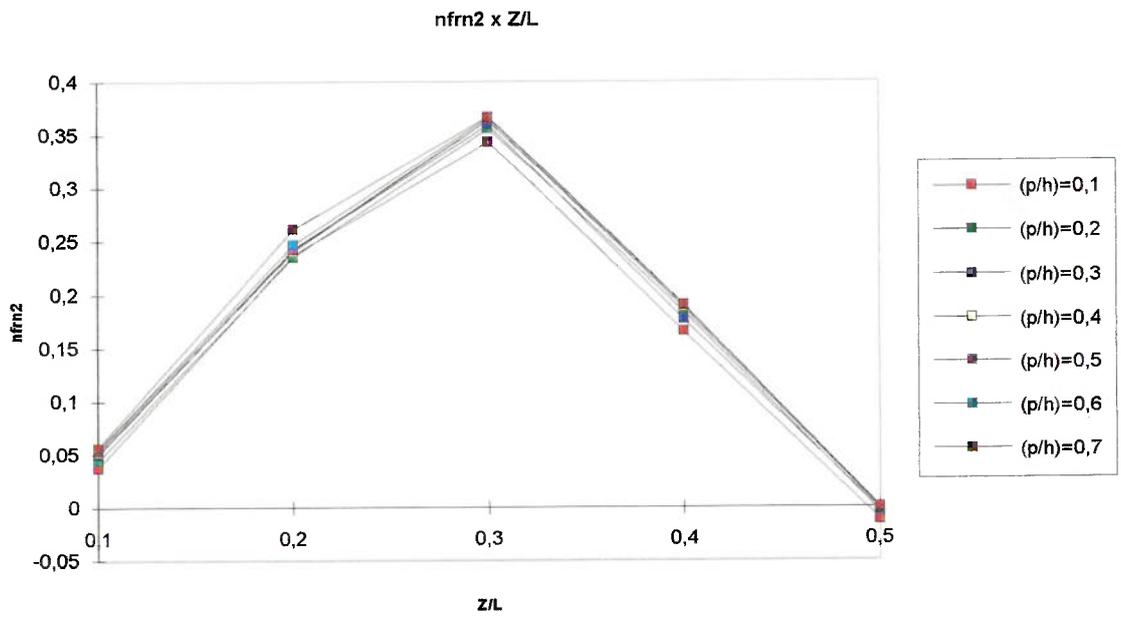


Gráfico - 5.1.31

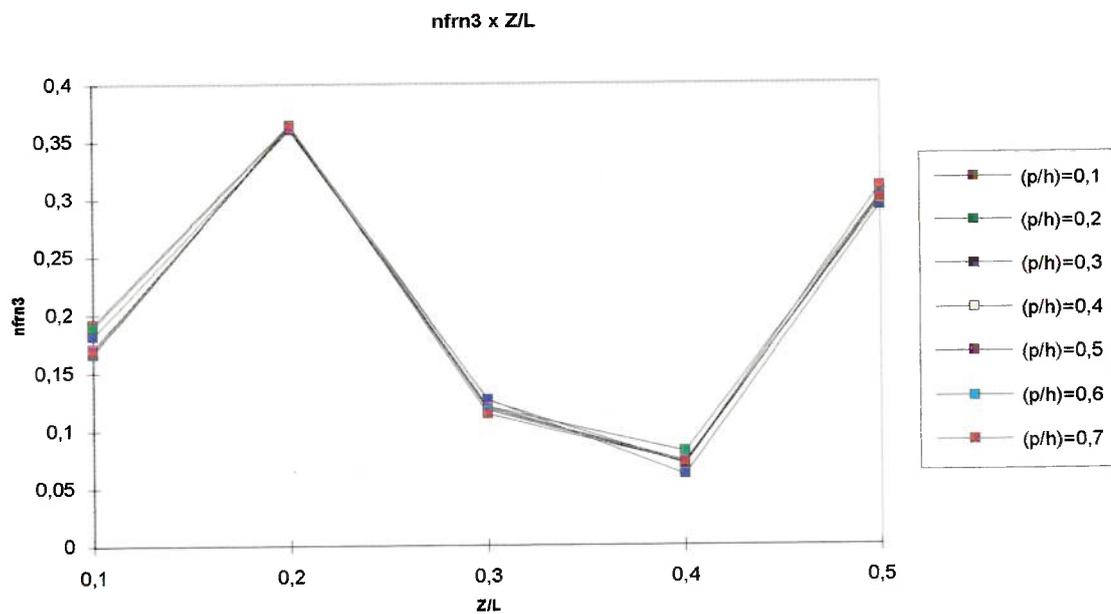


Gráfico - 5.1.32

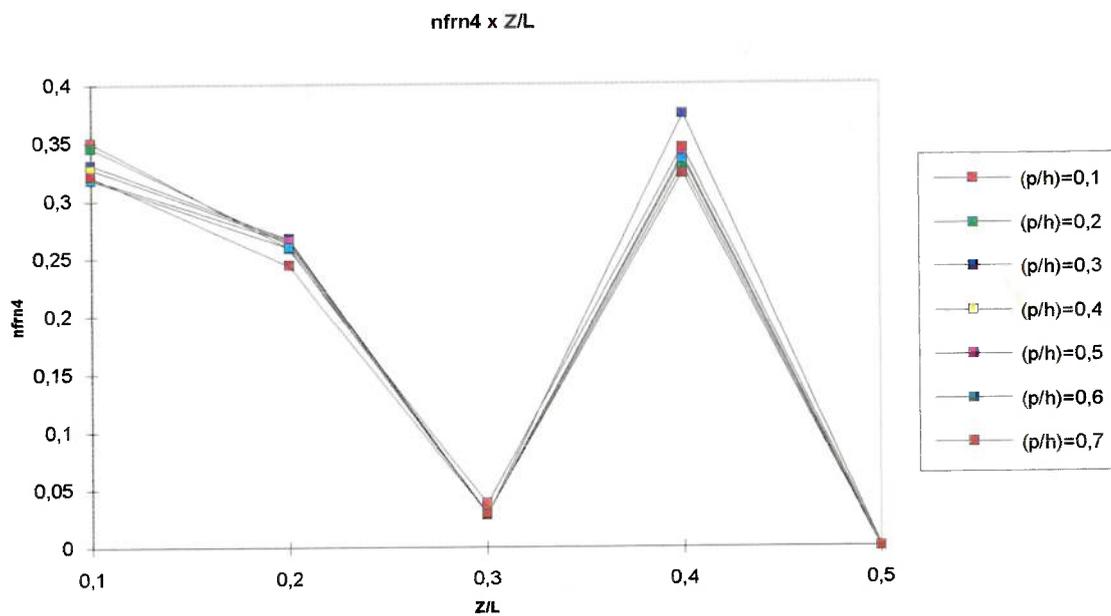


Gráfico - 5.1.33

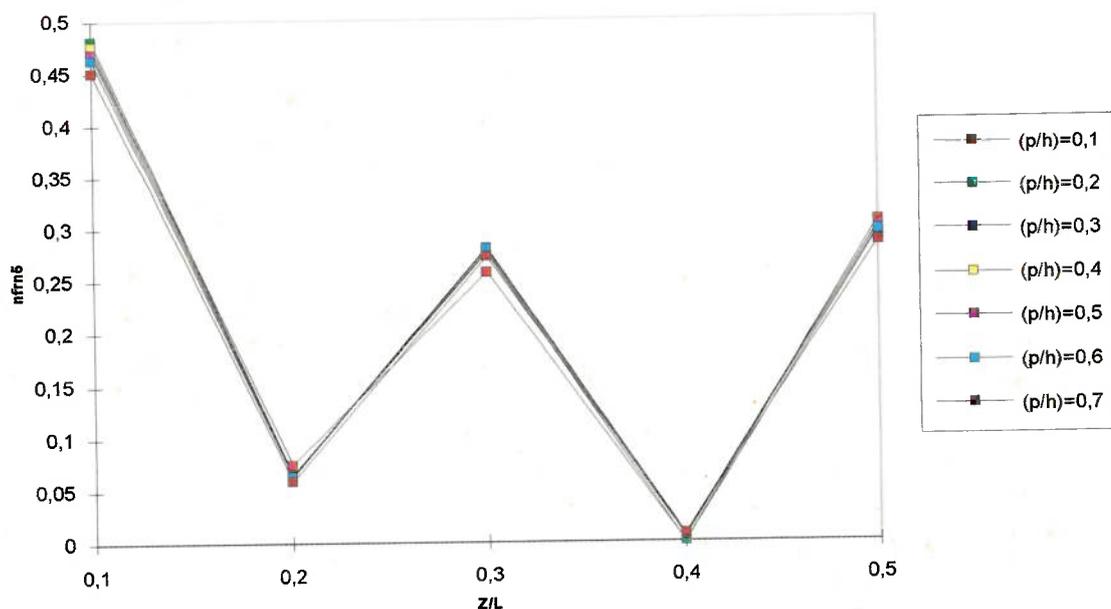


Gráfico - 5.1.34

### 5.1.2 Rede neural adotada

A etapa de aplicação de uma RNA ao problema estudado envolve, basicamente, duas fases. Na primeira é feita a construção dos conjuntos de dados de treinamento e teste. A segunda fase consiste na obtenção da configuração da RNA mais apropriada para este problema.

Na montagem do conjunto de dados de treinamento, deve-se ter o cuidado de abranger todas as possíveis respostas que a rede possa vir a ter que fornecer, isto é, este conjunto deve conter os limites do domínio do problema, ou, ao menos, valores próximos a estes limites uma vez que RNAs não são indicadas para extrapolações. O conjunto de dados do teste deve, por sua vez, conter dados que não estejam presentes no conjunto de dados de treinamento, sendo inéditos para a RNA.

Quanto ao estabelecimento da configuração da RNA mais apropriada, pode-se dizer que este processo visa determinar todos os aspectos da RNA, desde o número de camadas ocultas até qual será a função de transferência dos neurônios, passando ainda pelo ajuste dos parâmetros como a taxa de aprendizado,  $\eta$ . Como será visto mais adiante, este processo completo não é

simples. Esta complexidade exige que sejam fixados desde o princípio alguns dos parâmetros e que se estabeleça um procedimento para que seja encontrada a RNA mais adequada.

Para treinamento e teste das RNAs foram usados neste trabalho programas de redes neurais disponíveis no mercado. Estes foram o NWORKS e o MATLAB 4.0, com o "toolbox" de redes neurais. Dentre estes dois programas, o mais utilizado foi o NWORKS, por se tratar de um sistema dedicado ao desenvolvimento de aplicações em redes neurais. O MATLAB foi usado apenas para comparação de resultados, isto é, duas redes iguais mas simuladas em programas diferentes, com o mesmo conjunto de dados de treinamento, devem fornecer resultados similares.

#### 5.1.2.1 Conjunto de dados de treinamento e teste

O conjunto de dados usados no treinamento das diferentes configurações de RNA contém os mesmos casos de defeitos adotados no item 5.1.1. Entretanto, foram retirados deste conjunto três casos das posições  $Z_i/L = 0,4$  (escolhida aleatoriamente). Estes casos, juntamente com mais cinco, referentes a posições não treinadas, foram usadas como conjunto de dados de teste. A fonte destes dados novos também foi a simulação em elementos finitos de uma estrutura como a deste exemplo. Na tabela seguinte são apresentados os dados de teste.

	Pos.	Prof.		Pos.	Prof.
1	0.175	0.200	5	0.400	0.300
2	0.400	0.600	6	0.225	0.400
3	0.475	0.700	7	0.400	0.400
4	0.425	0.200	8	0.375	0.600

Tabela 5.1.1 Elementos do conjunto de teste.

A determinação do tamanho deste conjunto não segue nenhuma regra específica. É sabido que os seus elementos não devem pertencer ao conjunto de dados de treinamento. Mas, quanto ao número de casos de teste, não há regra definida para a sua determinação. Como em geral, não há disponibilidade de um grande número de dados e é dada prioridade para a construção do conjunto de dados de treinamento, o grupo de dados de teste acaba ficando restrito. De preferência, este último conjunto deve conter dados que cubram todo o domínio do problema ou, ao menos, boa parte deste para que seja possível determinar se a RNA treinada possui, ou não, capacidade de generalização. Deve ainda ser notado, que no caso deste conjunto de dados de teste, a posição do

defeito igual a  $Z/L = 0,4$  também aparece no conjunto de treinamento mas não nas profundidades que foram consideradas no teste.

### 5.1.2.2 Determinação da RNA mais adequada

Diferentes configurações de redes neurais artificiais foram tentadas até que se chegou a uma que forneceu melhores resultados. No entanto, desde o início já havia sido feita uma opção pelas redes com o algoritmo de retro-propagação do erro, justamente pelo fato deste tipo de rede e deste algoritmo serem os com maior documentação disponível e também pelo fato de já terem sido usados com sucesso em outras aplicações. Outra opção feita logo no início do treinamento das RNAs foi quanto a função de transferência dos neurônios. Foi logo percebido que as redes com neurônios cuja função de transferência era a logística, apresentaram dificuldades de convergência no treinamento, sendo que algumas efetivamente não convergiram. Tal fato não ocorreu com as redes que usavam a tangente hiperbólica como função de transferência dos neurônios. Usando esta função de transferência, todas as rede convergiram no processo de treinamento.

A determinação do número de camadas e o número de neurônios em cada uma destas é claramente dividida em dois casos. Num primeiro, há a determinação do número de neurônios nas camadas de entrada e saída da RNA. No caso das entradas o número de neurônios é dado pelo número de modos de vibrar considerados, que neste caso, foi limitado a cinco modos, visando executar a detecção de defeitos com um número pequeno de frequências a serem medidas. Já para a camada de saída o número de neurônios é dado pela as informações que se deseja que a rede forneça, neste caso são dois neurônios, um para a posição e outro para a profundidade do defeito.

O problema de determinação do número de camadas e de neurônios nestas fica bem mais complexo quando se considera o caso das camadas ocultas. Não foi encontrada na bibliografia nenhuma regra geral para, a partir de um problema e um conjunto de dados, determinar nem o número de camadas ocultas nem o de neurônios nestas. Pelo contrário, as referências apontam como uma desvantagem do emprego de redes neurais, justamente este problema [51].

Caso fossem conhecidas a priori as regiões de representação do problema seria possível determinar o número de camadas ocultas (ver item 2.1). Entretanto, este não é o caso do problema considerado. A saída, neste caso, foi seguir a tendência de configurações de redes encontradas na bibliografia, isto é, para problemas semelhantes (problema inverso) com números de saídas e entradas próximos aos deste caso e ainda com quantidade similar de dados de treinamento, foram adotadas redes neurais com certa configuração. Esta mesma configuração seria então usada como tentativa inicial para rede neural a ser usada. Após verificações na bibliografia foi determinado que uma rede neural com duas camadas ocultas, com dez neurônios em cada uma seria usada como configuração de partida para a procura da RNA mais adequada. Cabe ressaltar que, as RNAs encontradas nas referências bibliográficas não formam um grupo com características semelhantes quanto a configuração e que este método para determinação de uma RNA inicial também não é embasado em fundamentos teóricos, bem como não está descrito em nenhuma das referências bibliográficas consultadas.

A RNA inicial será treinada e em seguida testada com um conjunto de dados que não pertencem ao conjunto de dados de treinamento. Será a partir deste teste que será efetivamente determinada a configuração mais adequada ao problema. Isto é conseguido comparando-se o erro cometido pelas diferentes redes para o conjunto de dados de teste. As diferentes configurações de RNA são obtidas desde a inicial por meio da adição ou subtração de neurônios nas camadas ocultas. Cada uma destas configurações deve ser treinada com o mesmo conjunto de dados de treinamento que as outras e o mesmo vale para os dados de teste. A medida que vão sendo treinadas as RNAs, o erro cometido com os dados de teste é monitorado e assim é direcionado o processo de obtenção da configuração mais adequada. Aquela RNA que fornecer o menor erro será considerada a mais adequada. Deve-se ainda notar que, é considerado que as configurações testadas apresentaram convergência no treinamento, pois caso esta não ocorra não há sentido em testar a rede com novos dados. Para tanto, foi considerado que uma rede havia convergido no treinamento caso o erro quadrático médio para os dados do conjunto de treinamento fosse menor que 0,01.

A tabela abaixo apresenta o erro quadrático total (soma para todos os casos de testes) para as redes consideradas. A coluna “neurônios” corresponde ao número destes na primeira e segunda camada oculta respectivamente.

Neurônios	Erro na posição	Erro na profundidade
10 - 10	1,0350E-02	1,2198E-02
10 - 08	2,0717E-03	1,7150E-01
10 - 06	3,2110E-02	6,5561E-02
10 - 04	2,8492E-02	3,0172E-02
08 - 08	2,0125E-02	9,0749E-02
08 - 06	3,3245E-02	1,0205E-01
08 - 04	3,0594E-02	3,9553E-02
06 - 06	3,4625E-02	1,1307E-01
06 - 04	6,2339E-03	4,2636E-02
04 - 04	5,4785E-03	2,1604E-01

Tabela 5.1.2: Evolução do erro para diferentes RNAs

Neste exemplo, a rede que forneceu menor erro global para o conjunto de teste era constituída por uma camada inicial composta por cinco entradas seguida por uma camada oculta com seis unidades. A segunda camada oculta possuía quatro neurônios e, finalmente, a camada de saída contendo dois neurônios cada um correspondendo a uma saída, posição e profundidade do entalhe. Como outros parâmetros desta rede neural são ainda lembrados a taxa de aprendizado  $\eta = 0,3$  e o coeficiente do termo de momento  $\alpha = 0,4$ . No apêndice II estão disponíveis os pesos das conexões entre neurônios desta RNA.

### 5.1.3 Resultados do exemplo - 1

Os resultados obtidos para o conjunto de testes estão listados na tabela seguinte. A seguir são apresentados os mesmos resultados na forma de gráficos comparativos do resultado obtido frente ao que seria esperado. Nas abcissas destes gráficos estão os valores obtidos para a posição do entalhe, em fração do comprimento ( $Z/L$ ) e em fração da altura para a profundidade ( $p/h$ ).

	Pos.	Prof.		Pos.	Prof.
1	0,180	0,165	5	0,431	0,300
2	0,399	0,600	6	0,229	0,438
3	0,472	0,751	7	0,410	0,405
4	0,426	0,203	8	0,373	0,599

Tabela 5.1.3 Resultados do primeiro exemplo.

Vê-se tanto a partir da tabela quanto dos gráficos que os resultados obtidos estão próximos ao que seria esperado. Embora em alguns dos diagramas possa parecer que a diferença seja excessiva, uma análise levando em conta os valores marcados nas abcissas, e vistos na tabela, mostra que esta diferença não é tão grande.

Neste caso, um conjunto de treinamento maior e algumas novas configurações da RNA, isto é, outras redes com número de camadas ocultas e de neurônios diferentes do que foi adotado, diferentes funções de transferência e outros valores para os parâmetros do algoritmo utilizado, poderiam levar a resultados mais precisos.

Mesmo com as pequenas diferenças encontradas, pode-se afirmar que a rede neural conseguiu informar a posição e a profundidade dos defeitos. Nas referências bibliográficas [19, 28, 39, 42] onde aparecem exemplos semelhantes ao adotado, isto é, uma viga com um defeito em diferentes possíveis localizações e distintos graus de severidade, não foram vistos casos onde fosse possível detectar as posições em qualquer ponto até o meio da viga, como neste caso, e ao mesmo tempo determinar a severidade do defeito, em tantos graus distintos, e ainda usar apenas as informações contidas nas frequências naturais, sem empregar qualquer informação a respeito dos modos de vibrar.

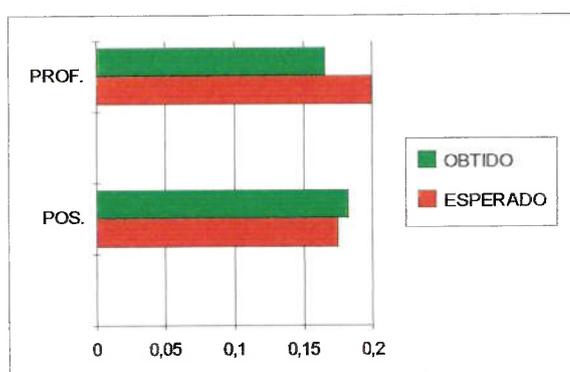


gráfico - 5.1.35

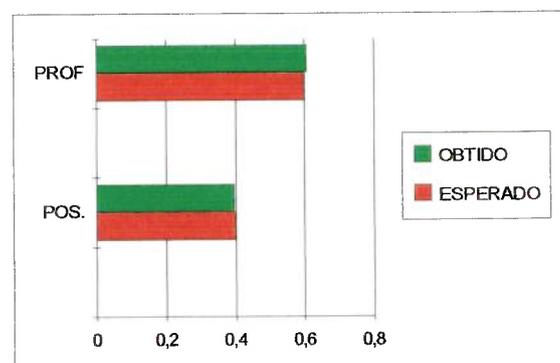


gráfico - 5.1.36

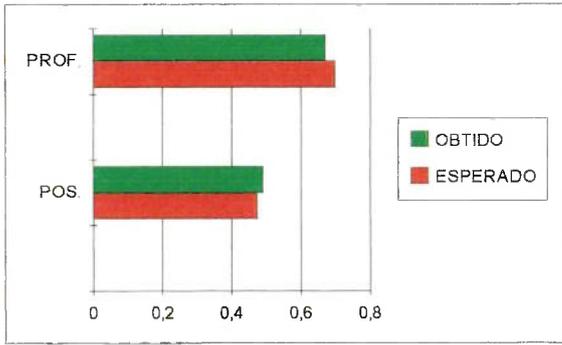


gráfico - 5.1.37

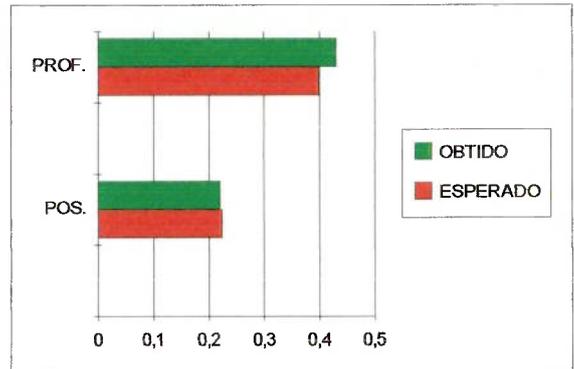


gráfico - 5.1.40

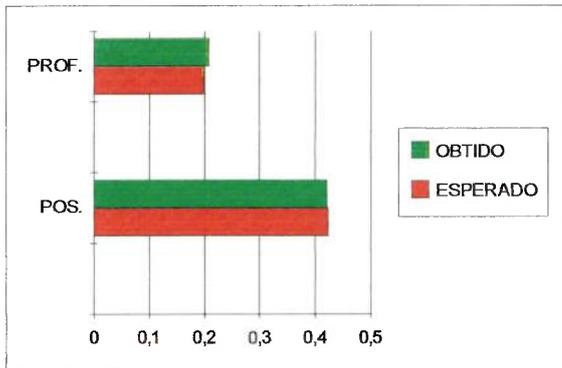


gráfico - 5.1.38

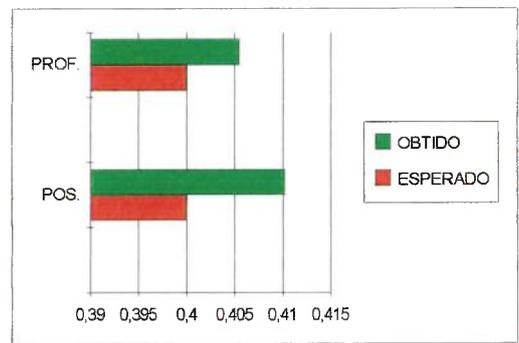


gráfico - 5.1.41

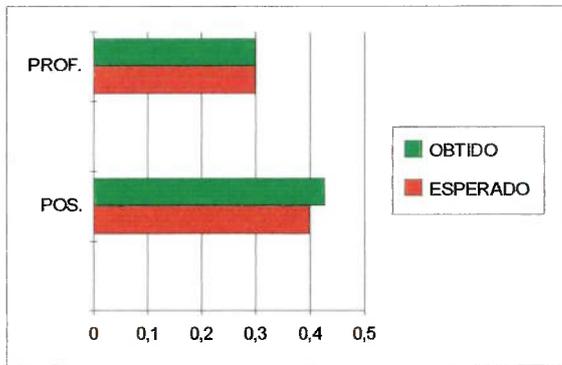


gráfico - 5.1.39

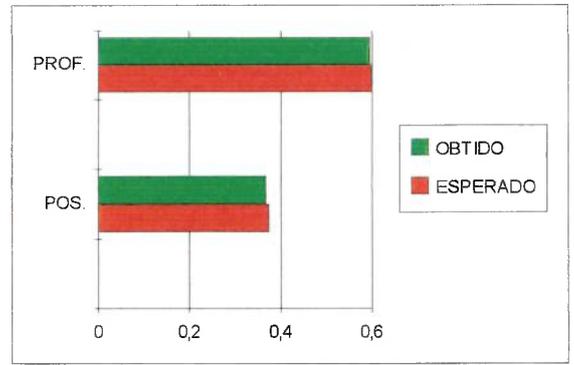


gráfico - 5.1.42

## 5.2 Exemplo - 2: Torre

O segundo exemplo consiste numa torre aperticada, composta por 48 elementos de viga, apresentada na figura 5.2.1. [2]

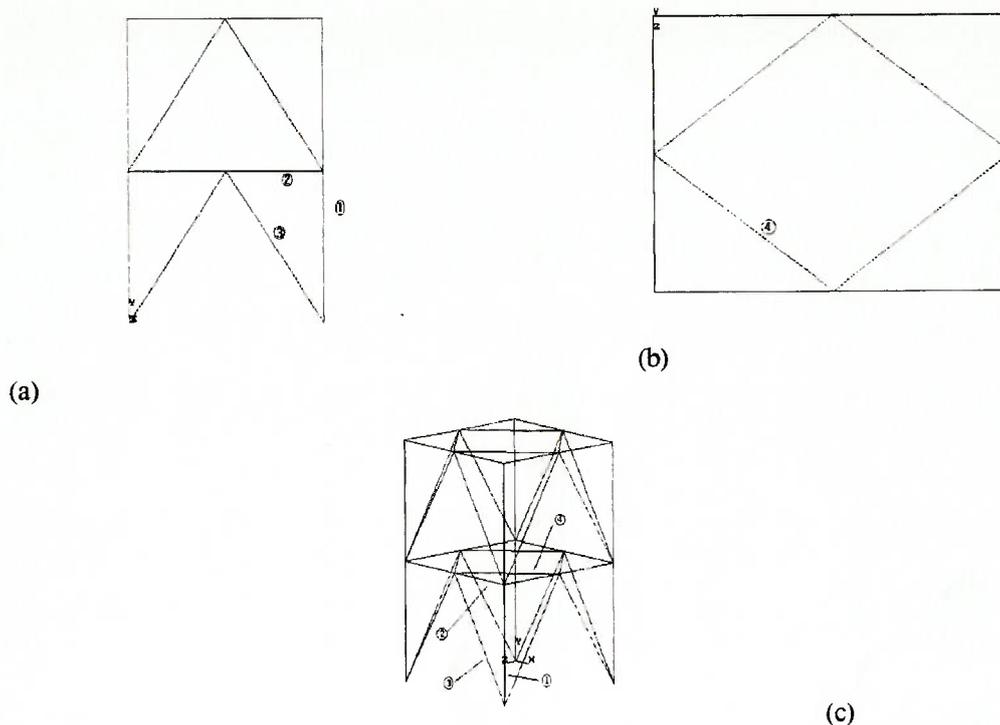


Fig. 5.2.1 Exemplo - 2 Torre aperticada - (a) Vista frontal. (b) Vista superior. (c) perspectiva isométrica

As vigas tem seção transversal quadrada de lado 0,010m. As verticais possuem comprimento igual a 1,0m e as horizontais a 0,5m. As barras diagonais (fig. 5.2.1 (a)) possuem comprimento igual a 1,18m enquanto as reforçadoras dos planos horizontais (fig. 5.2.1 (b)) tem 0,71m de comprimento.

O material usado é o mesmo do exemplo anterior. Da mesma forma que naquele caso, a estrutura foi modelada em elementos finitos empregando elementos de viga. As condições de contorno neste caso, estabelecem deslocamentos restritos para todos os nós que compõem a base da torre. O programa de elementos finitos usado para a obtenção das frequências naturais foi, como no outro exemplo, o ANSYS REV 5.0. As frequências naturais e modos de vibrar para esta estrutura estão no apêndice - I

O defeito foi modelado como uma perda de rigidez dos elementos. As seguintes frações de rigidez foram adotadas: 1,0; 0,9; 0,7; 0,6; 0,5; 0,2; 0,1; 0,0 (A simulação de um elemento com rigidez nula foi feita adotando-se um valor muito próximo de zero para evitar a interrupção do programa de elementos finitos). A viga não danificada possui fração de rigidez 1,0 e a totalmente danificada tem fração de rigidez igual a zero. Devido a simetria da estrutura foram testados

modelos onde apenas os elementos assinalados na figura 5.2.1 foram danificados. A rede neural deverá fornecer como saída a fração de rigidez destes elementos marcados.

Este tipo de modelo para o defeito é indicado para estruturas de maior porte, com muitos elementos, onde o objetivo principal é detectar se algum elemento pode estar danificado ou onde se quer monitorar alguns elementos importantes. Caso se detecte algum elemento danificado, este é em seguida avaliado em separado para um levantamento de seu estado.

O parâmetro de entrada para as RNAs usadas neste exemplo é o  $\eta$ , foram consideradas as frequências naturais referentes aos dez primeiros modos de vibrar da estrutura no cálculo dos parâmetros e, portanto, como entradas para a rede neural artificial.

Para o teste da rede foram adotados os seguintes valores para a fração de rigidez: 0,0; 0,4; 0,5; 0,8; 0,9. Os valores das frequências naturais, necessários para a obtenção dos conjuntos de dados de treinamento e teste, estão disponíveis no apêndice - I.

### 5.2.1 Rede neural adotada

Da mesma forma que no exemplo anterior foram testadas várias configurações de RNAs. O processo para obtenção da rede mais adequada é exatamente aquele descrito no primeiro exemplo. Os resultados apresentados são aqueles obtidos a partir da melhor configuração encontrada.

A RNA inicial continha 16 neurônios em cada uma das suas duas camadas ocultas. Na camada de saída havia um neurônio para cada elemento monitorado, perfazendo portanto, um total de quatro neurônios nesta camada. A rede possui dez entradas, uma para cada frequência natural considerada. O processo de treinamento e teste de diferentes configurações levou a obtenção de uma RNA considerada mais adequada ao problema. Este processo é ilustrado pela tabela seguinte (5.2.1) que apresenta o erro quadrático total obtido para o conjunto de dados de treinamento. A primeira coluna fornece o número de neurônios nas camadas ocultas e as demais colunas nesta tabela são referentes a cada um dos elementos monitorados. Logo, a coluna "Erro 1", p. ex., corresponde ao erro quadrático total cometido na avaliação da fração de rigidez do elemento um da estrutura (ver fig. 5.2.1) para todo o conjunto de teste.

A rede que apresentou melhor desempenho tinha, além dos dez neurônios da camada de entrada e dos quatro da de saída, quatorze neurônios na primeira camada oculta e mesmo número na camada oculta subsequente.

Neurônios	Erro 1	Erro 2	Erro 3	Erro 4
16 - 16	1,0579	0,0188	0,3384	0,0556
16 - 14	1,1978	0,0552	0,4583	0,0749
16 - 12	1,2447	0,1850	0,3734	0,1514
16 - 10	1,0361	0,0379	0,4009	0,0680
16 - 08	0,8813	0,0280	0,7055	0,1129
16 - 06	1,29640	0,2135	0,2961	0,0980
14 - 14	0,7702	0,0179	0,2669	0,0503
14 - 12	0,8708	0,0695	0,2362	0,0845
14 - 10	1,2159	0,0829	0,5069	0,1142
14 - 08	1,2426	0,0436	0,9877	0,1087
14 - 06	0,7337	0,0303	0,5352	0,0963
12 - 12	1,1157	0,0693	1,0402	0,0632
12 - 10	0,8831	0,0272	0,3458	0,1011
12 - 08	1,2126	0,0661	0,4663	0,2111
12 - 06	1,2616	0,0340	0,5684	0,1679
10 - 10	1,2251	0,0319	0,9430	0,1424
10 - 08	1,2364	0,0351	0,6721	0,0792
10 - 06	0,9946	0,0244	0,2824	0,0856
08 - 08	0,9155	0,0247	0,2961	0,0552
08 - 06	0,9663	0,0132	0,5343	0,0910
06 - 06	1,2667	0,0833	0,8608	0,1205

Tabela 5.2.1 - Evolução do erro para diferentes RNAs

Tal como no exemplo anterior, os neurônios possuem como função de transferência a tangente hiperbólica. As redes neurais artificiais usadas empregam o algoritmo de retropropagação do erro para o treinamento. Ainda como parâmetros da rede neural tem-se a taxa de aprendizado  $\eta = 0,25$  e o coeficiente do termo de momento  $\alpha = 0,4$ . Da mesma forma que no exemplo 1, estão disponíveis no apêndice II os pesos das conexões para os neurônios desta RNA.

### 5.2.2 Resultados do exemplo - 2

Os resultados obtidos estão apresentados sob a forma de tabelas e gráficos. Na tabela 5.2.1 aparecem os valores esperados e obtidos para cada caso do conjunto de teste. Estes mesmos resultados são apresentados na forma de gráficos de barras.

	ELEMENTOS				ELEMENTOS			
	1	2	3	4	1	2	3	4
esperado	1,0	0,5	1,0	1,0	1,0	1,0	0,9	0,6
obtido	0,98	0,48	0,98	1,00	0,99	1,00	0,88	0,68
erro	2,0%	4,2%	2,0%	0,0%	1,0%	0,0%	2,2%	13%
esperado	1,0	1,0	0,4	1,0	0,0	1,0	1,0	0,4
obtido	0,92	0,92	0,33	0,96	0,0	1,00	0,95	0,38
erro	8,0%	8,0%	18%	4,0%	-	0,0%	5,0%	5,0%
esperado	1,0	1,0	1,0	0,5	0,9	1,0	0,5	0,5
obtido	0,98	1,00	1,00	0,59	0,91	0,95	0,56	0,55
erro	2,0%	0,0%	0,0%	18%	1,1%	5%	12%	10%
esperado	0,9	1,0	1,0	0,9	0,8	0,5	0,5	1,0
obtido	0,94	0,99	0,98	0,92	0,84	0,50	0,51	0,99
erro	4,4%	1,0%	2,0%	2,2%	5,0%	0,0%	2,0%	1,0%
esperado	0,8	1,0	0,5	1,0	1,0	0,0	0,0	0,0
obtido	0,80	0,90	0,50	0,90	0,98	0,00	0,02	0,02
erro	0,0%	10%	0,0%	10%	2,0%	-	-	-
esperado	0,4	1,0	1,0	0,4	0,0	0,7	0,8	0,5
obtido	0,35	1,00	0,97	0,43	0,01	0,65	0,84	0,45
erro	13%	0%	3%	7,5%	-	7,1%	5,0%	10%

Tabela -5.2.2 Resultados do arquivo de teste do exemplo 2.

Nos gráficos, cada par de barras é correspondente a fração de rigidez de um dos elementos assinalados na figura 5.2.1. Para cada um deste há uma barra com a fração de rede esperada e aquela obtida pela rede neural.

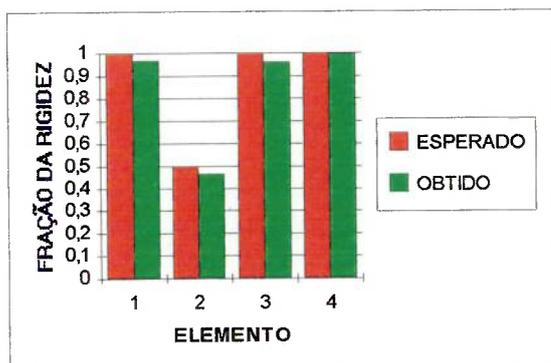


gráfico - 5.2.1

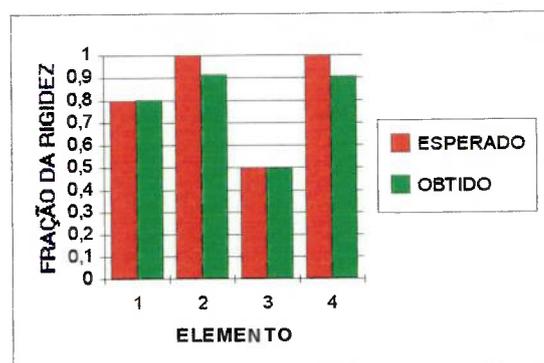


gráfico - 5.2.5

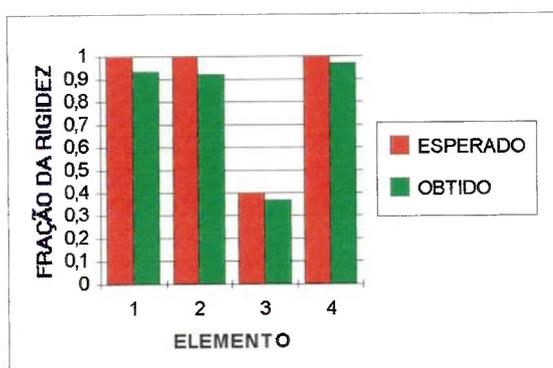


gráfico - 5.2.2

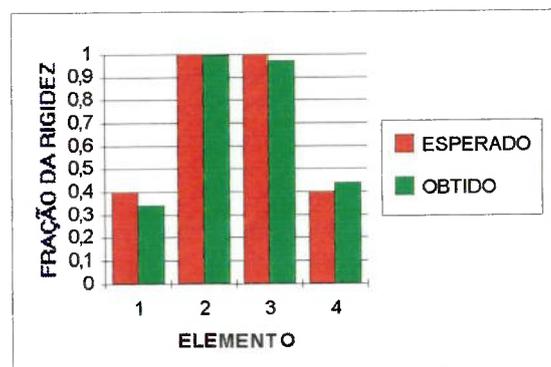


gráfico - 5.2.6

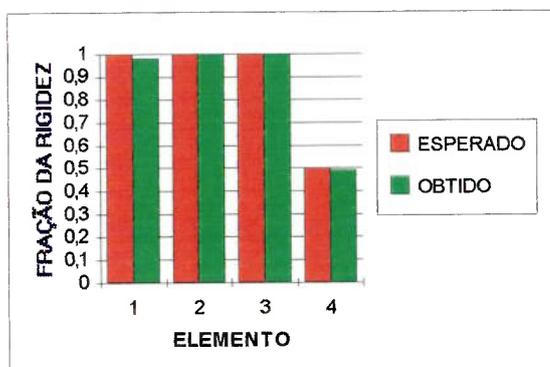


gráfico - 5.2.3

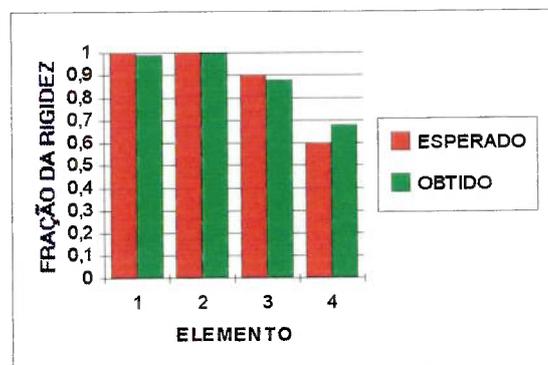


gráfico - 5.2.7

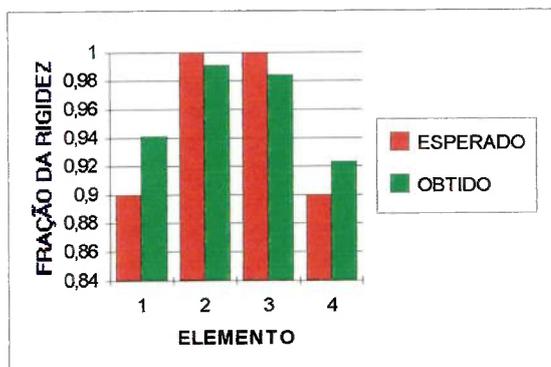


gráfico - 5.2.4

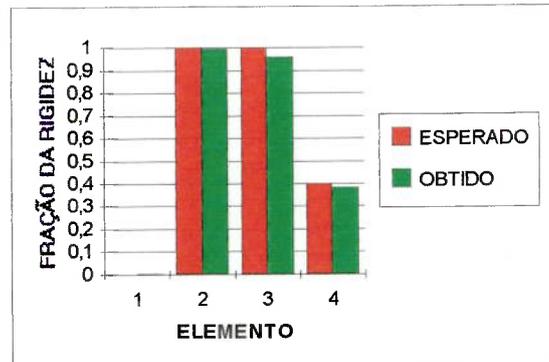


gráfico - 5.2.8

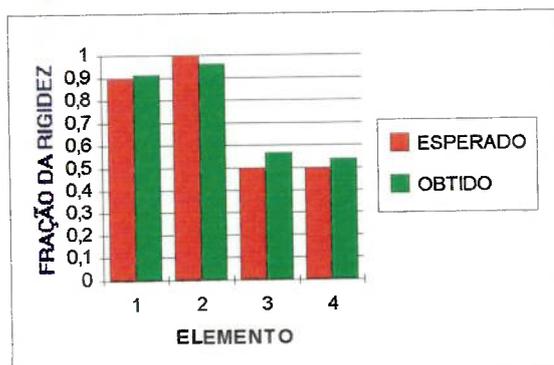


gráfico - 5.2.9

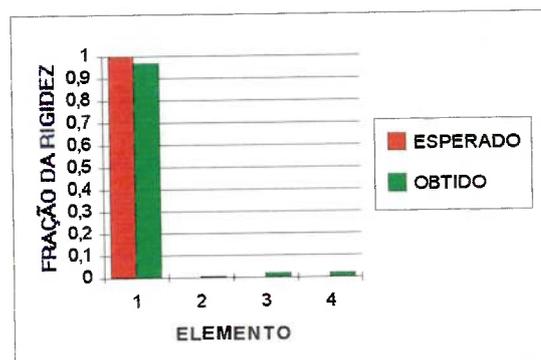


gráfico - 5.2.11

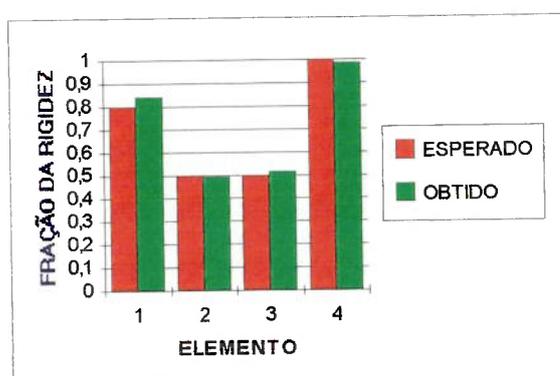


gráfico - 5.2.10

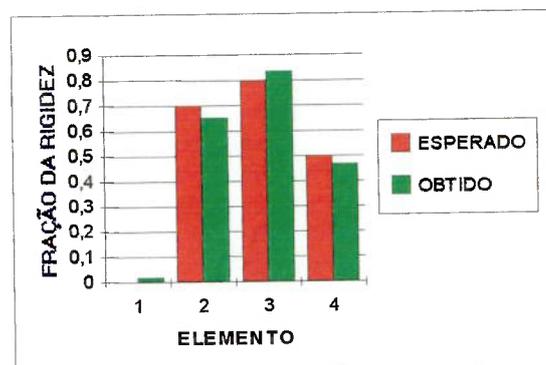


gráfico - 5.2.12

Nos casos onde houve mais de um elemento danificado, a rede conseguiu identificar quais dos elementos sofreu uma perda de rigidez. Entretanto, a avaliação da fração de rigidez não foi precisa em alguns casos, o que não impede a identificação dos elementos danificados.

As barras não defeituosas foram sempre identificadas como tendo fração de rigidez próxima a unidade. Os elementos que tinham perda total de rigidez também foram identificados pela RNA. Os pequenos valores encontrados se devem ao fato dos dados terem sido obtidos para valores de rigidez pequenos, no entanto, diferentes de zero.

Na maior parte dos casos, a fração de rigidez foi determinada com precisão de maneira que pode-se afirmar que a rede conseguiu não apenas indicar qual dos elementos monitorados apresentava defeitos como também conseguiu fornecer uma informação relativa a severidade deste dano através da fração de rigidez.

## 6. CONCLUSÕES E TRABALHOS FUTUROS

Analisando os resultados obtidos é possível afirmar que as redes neurais demonstram ser bons instrumentos computacionais, em substituição a outros métodos para resolução de um problema inverso, quando se aplica as variações na resposta dinâmica de uma estrutura para a detecção de defeitos na mesma [2, 23, 24].

As diferenças encontradas entre os valores obtidos pelas redes neurais e aqueles esperados para os exemplos podem ser creditadas a imprecisões intrínsecas às RNAs, que não fornecem sempre resultados absolutamente exatos, ao fato dos arquivos de treinamento serem de tamanho limitado e ainda a um prematuro término do treinamento das RNAs sem que estas tenham sido modificadas até obter-se o menor erro possível entre saídas calculadas e esperadas. Outro motivo para as diferenças anotadas pode ser ligado a própria configuração das RNAs. Neste caso, existem duas prováveis causas. Numa primeira, a rede estaria "super-dimensionada", isto é, com um número excessivo de unidades de processamento e portanto com muitos pesos para serem determinados. Desta forma, o número de dados do conjunto de treinamento deveria ser maior, havendo ainda o risco de obter-se uma rede que minimiza o erro no treinamento mas não consegue generalizar [20]. Numa outra causa, o número de neurônios pode ser muito pequeno para representar o problema e, portanto, a rede é insuficiente para a solução daquele problema.

Outro fator, que também pode ter levado as diferenças observadas, é o fato dos números de frequências naturais usados nos dois exemplos serem insuficientes para a representação dos defeitos. Logicamente, o fato de ter-se limitado o número de frequências naturais empregadas levou a uma perda de informação de como os defeitos afetam o comportamento dinâmico da estrutura, afinal as frequências naturais não consideradas, provavelmente também são afetadas pelos defeitos. Entretanto, um dos objetivos iniciais era obter um ensaio mais simples, usando um número mínimo de sensores. Assim, o número de frequências que podem ser medidas fica realmente comprometido, havendo portanto, um compromisso entre a precisão do método proposto e o número de frequências naturais medidas.

De forma global, observou-se que, os resultados indicam que a rede neural artificial conseguiu detectar os defeitos na estrutura e ainda determinou a sua localização e extensão com precisão razoável, menos de 10% de erro, na maioria dos casos usados para teste.

As diferenças entre os exemplos apresentados indicam que o método proposto tem aplicações em diversas categorias de estruturas e em diferentes situações de defeitos. Esta característica não é muito comum em outros tipos de END baseados nas modificações da resposta dinâmica.

Como trabalhos futuros, pode-se sugerir duas linhas básicas de atuação. Uma primeira linha refere-se a melhoria do desempenho das redes neurais artificiais aplicadas ao problema. A segunda linha compreende o desenvolvimento do ensaio como um todo.

Para o desempenho das redes neurais, sugere-se o desenvolvimento de novos parâmetros, obtidos como pré-processamento dos dados medidos, que possam ser mais sensíveis a presença de defeitos.

Uma outra possibilidade é o emprego de novos algoritmos ou de desenvolvimentos dos que já existem. Redes neurais ainda estão sendo desenvolvidas do ponto de vista teórico e muitos algoritmos tem desempenho melhor do que o algoritmo de retro-propagação empregado neste trabalho.

Um exemplo deste ultimo ponto é a possibilidade de usar métodos que vão variando a configuração da RNA, extraíndo ou colocando neurônios de maneira a obter a melhor rede possível para uma determinada precisão especificada.

Já no caso do desenvolvimento do ensaio de uma forma geral, a primeira sugestão é fazer uma verificação experimental do que foi proposto. Nos dois exemplos, os modelos para os defeitos não são tão próximos do que se encontraria na realidade. Uma verificação experimental é fundamental para que se tenha uma comprovação final da aplicação prática da metodologia. Esta verificação poderia ser feita com estruturas similares as dos exemplos.

Outro ponto importante e que liga as duas vertentes de atuação proposto é a maneira de construção do conjunto de dados de treinamento da RNA. Este conjunto deve conter o maior número de dados disponíveis para que seja possível treinar a rede neural, sendo que, alguns casos

ainda devem ser reservados para o teste da RNA. A obtenção de todo este conjunto de treinamento por meio de ensaio não parece ser muito produtiva, dado o número de casos necessários. Um estudo a ser feito deve levar em consideração a possibilidade de treinar a rede com dados obtidos, tanto de forma experimental, como de modelos em elementos finitos, por exemplo. O grau de complexidade e precisão destes modelos também devem ser determinados.

Um outro tópico a ser explorado diz respeito ao número de frequências a serem medidas que devem ser usadas quando se quer uma determinada precisão. Uma outra linha de ação visaria, dado o defeito que se está procurando, determinar quais os modos são mais afetados e portanto seriam medidas as frequências mais sensíveis a este defeito.

Uma vez que tenha sido comprovada experimentalmente a metodologia proposta, um desenvolvimento posterior englobaria o projeto de sistemas de ensaio e monitoração de estruturas que automaticamente executariam todas as tarefas, desde a medição dos dados até o fornecimento dos resultados de interesse. Redes neurais artificiais estão sendo implementadas em circuito eletrônico digital [29] e não é difícil imaginar "caixas pretas" que possam ser instaladas em estruturas e sejam responsáveis pela monitoração da integridade destas por exemplo.

## 7. BIBLIOGRAFIA

- [1] ALGUINDIGUE, I. E., LOSKIEWICZ-BUCZAK, A., UHRIG, R. E.; *Monitoring and diagnosis of rolling element bearings using artificial neural networks*, IEEE Transactions on industrial electronics, 40(2), 1993, p. 209-217
- [2] ALVES, M. A. L.; KAMINSKI, P.C. ; *Aplicação de análise modal e redes neurais na detecção de defeitos em estruturas e componentes*. Anais do III Congresso e Exposição Internacionais de Tecnologia da Mobilidade - SAE BRASIL 94, SAE Technical paper 942373, SAE Brasil, São Paulo, Nov. 1994
- [3] ANDERSON, J. A.; *Neuralcomputing - Foundations of research*, MIT Press, Cambridge - MA, E.U.A, 1988, pp 672
- [4] BAZ, A.; *A neural observer for dynamic systems*, Journal of sound and vibration, 152(2), 1992, p. 227-243
- [5] BARDOU, O., SIDAHMED, M.; *Early detection of leakages in the exhaust and discharge systems of reciprocating machines by vibration analysis*, Mechanical systems and signal processing, 8(5), 1994, p. 551-570
- [6] CHEN, J. C., GARBA, J. A.; *On-orbit damage assessment for large space structures*, AIAA Journal, 26(9), 1988, p. 1119-1126
- [7] CHOW, M., SHARPE, R. N., HUNG, J. C.; *On the application of artificial neural networks for motor fault detection (Parts I & II)*, IEEE Transactions on industrial electronics, 40(2), 1993, p. 181-196
- [8] COOK, A. B. et alii; *Artificial neural networks for predicting nonlinear dynamic helicopter loads.*, AIAA Journal, 32(5), 1994, p. 1072-1077
- [9] COPPOLINO, R. N., RUBIN, S.; *Detectability of structural failures in offshore platforms by ambient vibration monitoring*, Proceedings of the 12th Offshore Technology Conference, 1980, OTC Paper - 3865, p. 101-110
- [10] CRUSE, H., STEINKÜHLER, U.; *Solution of the direct and inverse kinematic problems by a common algorithm based on the mean of multiple computations.*, Biological Cybernetics, 69, 1993, p. 345-351
- [11] DIRR, B. O., POPP, K., ROTHKEGEL, M. H.; *Detection and simulation of small transverse cracks in rotating shafts*, Archive of applied mechanics, 64, 1994, p. 202-222
- [12] ELKORDY, M. F., CHANG, K. C., LEE, G. C.; *Impact-echo signal interpretation using artificial intelligence*, Journal of computing in civil engineering, 7(1), 1993, p. 71-93
- [13] GHABOUSSI, J., JOGHATAIE, A.; *Active control of structures using neural network.*, Journal of engineering mechanics, 121(4), 1995, p. 555-567
- [14] GHABOUSSI, J., WU, X., GARRETT, J. H.; *Knowledge-based modeling of material behavior with neural networks*, Journal of engineering mechanics, 117(1), 1991, p. 132-153
- [15] GOMES, A. J. M. A.; MONTALVÃO E SILVA, J. M.; *Experimental determination of the influence of the cross-section size in the dynamic behaviour of cracked beams*, Proceedings of the IMMDC2, Los Angeles, USA, 1, 1990, p. 414-423

- [16] GROSSBERG, S.; *Contour enhancements, short-term memory and consistencies in reverberating neural networks.*, Studies in applied mathematics, 52, 1973, p. 217-257
- [17] HEBB, D. O.; **The organization of behavior**, Wiley, Nova Iorque - NY, E.U.A, 1949
- [18] HUNT, D. L. et alii; *Development and implementation of a shuttle modal inspection system*, Sound and Vibration, August 1990, p. 34-42
- [19] JOSHI, A, MADHUSUDHAN, B. S.; *A unified approach to free vibration of locally damaged beams having various homogeneous boundary conditions* Journal of Sound and Vibration, 147(3), 1991, p. 475-488
- [20] KAMINSKI, P. C., ALVES, M. A. L.; **Introdução as redes neurais artificiais**, Monografia do departamento de Engenharia Mecânica, EPUSP, 1996
- [21] KAMINSKI, P. C.; **Análise dinâmica de estruturas modificadas**, Tese de doutorado, EPUSP, 1992
- [22] KAMINSKI, P. C.; *The approximate location of damage trough the analysis of natural frequencies with artificial neural networks*, Proc. Inst. of Mech. Engrs. - Journal of Process Mechanical Engineering, 209, 1995
- [23] KAMINSKI, P. C., ALVES, M. A. L.; *Utilização de frequências naturais e redes neurais para a detecção de defeitos estruturais*, Anais do 12º Congresso Nacional de Ensaios Não-destrutivos, ABENDE, São Paulo, 1993, p. 95-114
- [24] KAMINSKI, P. C., ALVES, M. A. L.; *Damage detection in structures through the analysis of natural frequencies using artificial neural networks*, Anais do DINAME 95 - VI Symposium on dynamical problems of mechanics, ABCM, Caxambu, 1995, p. 146-149
- [25] KENLEY, R. M., DODDS, C. J.; *West sole WE platform: Detection of damage by structural response measurements* Proceedings of the 10th Offshore Technology Conference, 1980, OTC Paper - 3886, p. 111-118
- [26] KIM, H. M. et alii; *Health monitoring of large structures*, Sound and Vibration, April 1995, p. 18-21
- [27] KOVÁČZ, Z. L.; **Redes neurais artificiais - Fundamentos e aplicações**, Edição acadêmica, São Paulo - SP, Brasil, 1996
- [28] KRAWCZUK, M., OSTACHOWICZ, W. M.; *Parametric vibrations of beam with crack*, Archive of applied mechanics, 62, 1992, p. 463-473
- [29] KUNG, S. Y.; **Digital neural networks**, Prentice-Hall, Englewood Cliffs - NJ, E.U.A, 1993
- [30] LIANG, R. Y., HU, J., CHOY, F.; *Quantitative NDE technique for assessing damages in beam structures*, Journal of engineering mechanics, 118(7), 1992, p. 1468-1487
- [31] LIM, T. W., KASHANGAKI, T. A. L.; *Structural damage detection of space truss structures using best achievable eigenvectors*, AIAA Journal, 32(5), 1994, p. 1049-1057
- [32] LIN, C. S.; *Location of modeling errors using modal teste data*, AIAA Journal, 28(9), 1990, p. 1650-1654

- [33] LOLAND, O., DODDS, C. J.; *Experiences in developing and operating integrity monitoring systems in the North Sea*, Proceedings of the 8th Offshore Technology Conference, 1976, OTC Paper - 2551, p. 313-319
- [34] LYON, R. H.; *Structural diagnostics using vibration transfer functions*, Sound and Vibration, January 1995, p. 28-31
- [35] MANNING, R. A.; *Structural damage detection using active members and neural networks.*, AIAA Journal, 32(6), 1994, p. 1331-1333
- [36] MCCULLOCH, W. S., PITTS, W.; *A logical calculus of the ideas immanent in nervous activity.*, Bulletin of mathematical Biophysics, 5, 1943, p. 115-133
- [37] MINSKY, M., PAPPERT, S.; *Perceptrons*, MIT Press, Cambridge - MA, E.U.A, 1969
- [38] NATKE, H. G., CEMPEL, C.; *Fault detection and localisation in structures: A discussion*, Mechanical systems and signal processing, 5(5), 1991, p. 345-356
- [39] PANDEY, A. K., BISWAS, M., SAMMAN, M. M.; *Damage detection from changes in curvature modes shapes*, Journal of Sound and Vibration, 145(2), 1991, p. 321-332
- [40] PRATT, D., SANSALONE, M.; *Impact-echo signal interpretation using artificial intelligence*, ACI material journal, 89(2), 1992, p. 178-187
- [41] RHIM, J., LEE, S. W.; *A neural network approach for damage detection and identification of structures.*, Computational mechanics, 16, 1995, p. 437-443
- [42] RIZOS, P. F., ASPRAGATHOS, N., DIMARAGONAS, A. D.; *Identification of crack location and magnitude in a cantilever beam from the vibration modes*, Journal of Sound and Vibration, 138(3), 1990, p. 381-388
- [43] ROSENBLATT, F.; *The perceptron: A probabilistic model for information storage and organization in the brain.*, Psychological Review, 65, 1958, p. 386-408
- [44] RUBIN, S.; *Ambient vibration survey of offshore platform.*, Journal of the engineering mechanics division (ASCE), 106(EM3), 1980, p. 425-441
- [45] RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J.; *Learning Internal representations by error propagation*, **Parallel distributing processing: Explorations in the microstructures of cognition V. 1**, MIT Press, Cambridge - MA, E.U.A, 1986
- [46] SHOURESHI, R.; *Intelligent control systems: Are they for real?.*, Journal of dynamic systems, measurements and control, 115, 1993, p. 392-401
- [47] SOEIRO, F. J., HAJELA, P.; *Damage detection in composite materials using identification techniques*, Journal of aerospace engineering, 6(4), 1993, p. 363-380
- [48] TSAI, W. H., YANG, J. C. S.; *Nondestructive evaluation of composite structures using system identification technique*, Journal of engineering materials and technology, 110, 1988, p. 134-139
- [49] UDPA, L., UDPA, S. S.; *Neural networks for the classification of nondestructive evaluation signals*, IEE Proceedings, 188(1), 1991, p. 41-45

- [50] VANDIVER, J. K.; *Detection of structural failure on fixed platforms by measurement of dynamic response*, Proceedings of the 7th Offshore Technology Conference, 1975, OTC Paper - 2267, p. 243-252
- [51] WASSERMAN, P. D.; *Advanced methods in neural computing*, Van Nostrand Reinhold, Nova Iorque - NY, E.U.A, 1993
- [52] WASSERMAN, P. D.; *Neural computing - Theory and practice*, Van Nostrand Reinhold, Nova Iorque - NY, E.U.A, 1993
- [53] WIDROW, B., HOFF, M. E.; *Adaptive Switching circuits*, 1960 IRE Wescon Convention Record, p. 96-104
- [54] WOJNAROWSKI, M. E., STIANSEN, S. G., REDDY, N. E.; *Structural integrity evaluation of a fixed platform using vibration criteria*, Proceedings of the 9th Offshore Technology Conference, 1977, OTC Paper - 2909, p. 247-256
- [55] WU, X., GHABOUSSI, J., GARRETT, J. H.; *Use of neural networks in detection of structural damage*, Computers and structures, 42(4), 1992, p. 649-659
- [56] YEH, Y. C., KUO, Y. H., HSU, D. S.; *Building KBES for diagnosing PC pile with artificial neural networks*, Journal of computing in civil engineering, 7(1), 1993, p. 71-93
- [57] ZIMMERMAN, D. C., KAOUK, M.; *Eigenstructure assignment approach for structural damage detection*, AIAA Journal, 30(7), 1992, p. 1848-1855
- [58] ZIMMERMAN, D. C., KAOUK, M., SIMMERMACHER, T.; *Structural health monitoring using vibration measurements and engineering insight*, Transactions of the ASME - Special 50th anniversary design issue, 117, June 1995, p. 214-221

**APÊNDICE I**

**MODOS DE VIBRAR DA ESTRUTURA DO EXEMPLO - I****Viga livre-livre**

1º modo - frequência = 248,34 Hz



2º modo - frequência = 682,20 Hz



3º modo - frequência = 1331,5 Hz



4º modo - frequência = 2189,0 Hz

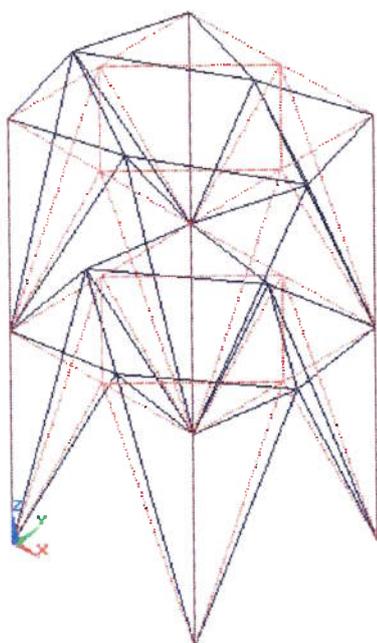


5º modo - frequência = 3249,0 Hz

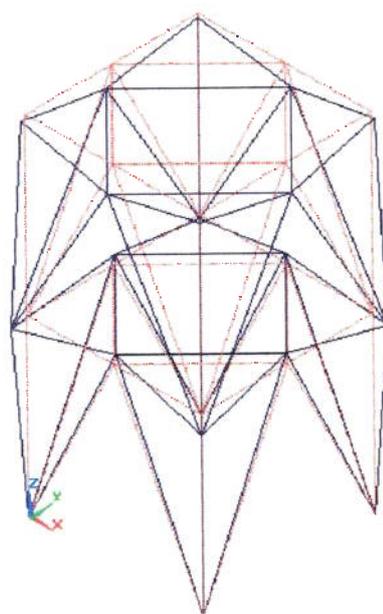
Na tabela abaixo estão as frequências naturais usadas no exemplo - 1.

Posição (Z/L)	Profundidade (p/h)	$\omega_1$ (Hz)	$\omega_2$ (Hz)	$\omega_3$ (Hz)	$\omega_4$ (Hz)	$\omega_5$ (Hz)
0,1	0,1	248,35	682,18	1331,3	2188,4	3247,8
	0,2	248,37	682,14	1331,0	2187,5	3245,9
	0,3	248,38	682,06	1330,5	2186,0	3242,6
	0,4	248,39	681,92	1329,7	2183,3	3236,7
	0,5	248,39	681,62	1327,9	2177,9	3224,8
	0,6	248,36	680,93	1324,0	2165,4	3198,0
	0,7	248,36	678,96	1312,7	2129,9	3125,9
0,2	0,1	248,32	682,00	1330,9	2188,3	3248,7
	0,2	248,28	681,67	1329,9	2187,1	3248,3
	0,3	248,23	681,10	1328,3	2185,1	3247,6
	0,4	248,12	680,06	1325,2	2181,4	3246,3
	0,5	247,92	678,01	1319,3	2174,3	3243,7
	0,6	247,47	673,37	1306,1	2159,2	3238,2
	0,7	246,18	660,56	1272,6	2124,1	3225,6
0,3	0,1	248,27	681,92	1331,3	2188,9	3248,0
	0,2	248,17	681,44	1331,0	2188,8	3246,2
	0,3	248,00	680,59	1330,4	2188,6	3243,1
	0,4	247,68	679,05	1329,5	2188,2	3237,5
	0,5	247,06	675,99	1327,5	2187,4	3226,2
	0,6	245,67	669,22	1323,4	2185,7	3201,8
	0,7	241,81	651,79	1313,1	2181,2	3141,7
0,4	0,1	248,23	682,08	1331,4	2188,2	3249,0
	0,2	248,06	681,85	1331,2	2187,0	3249,0
	0,3	247,76	681,46	1331,0	2184,7	3248,9
	0,4	247,22	680,74	1330,4	2180,6	3248,8
	0,5	246,15	679,30	1329,4	2172,5	3248,5
	0,6	243,76	676,13	1327,1	2154,9	3247,9
	0,7	237,32	667,99	1321,1	2112,1	3246,4
0,5	0,1	248,22	682,21	1331,0	2189,0	3247,8
	0,2	248,01	682,21	1330,2	2189,0	3245,9
	0,3	247,66	682,21	1328,9	2189,0	3242,5
	0,4	247,02	682,21	1326,3	2189,0	3236,2
	0,5	245,76	682,21	1321,2	2189,0	3223,9
	0,6	242,94	682,21	1310,0	2189,0	3197,4
	0,7	235,46	682,21	1281,9	2189,0	3133,8

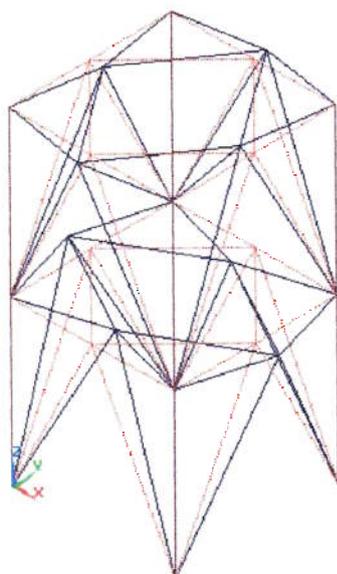
Tab. A.1.1 - Frequências naturais do exemplo - 1.

**MODOS DE VIBRAR DA ESTRUTURA DO EXEMPLO - II****Torre**

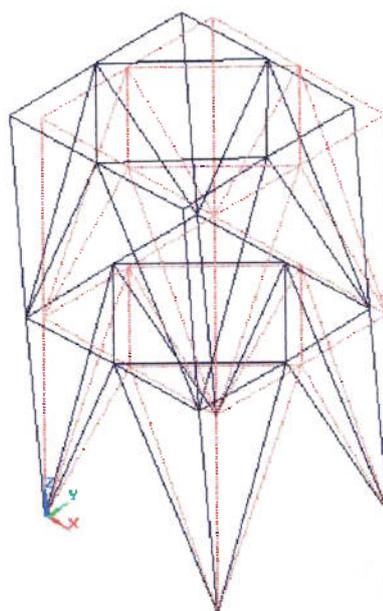
1º modo - frequência = 19,188 Hz



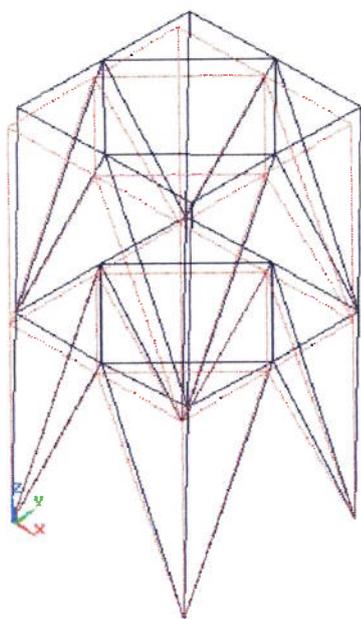
3º modo - frequência = 39,832 Hz



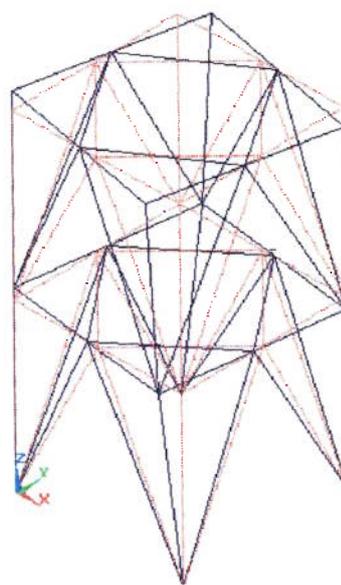
2º modo - frequência = 23,001 Hz



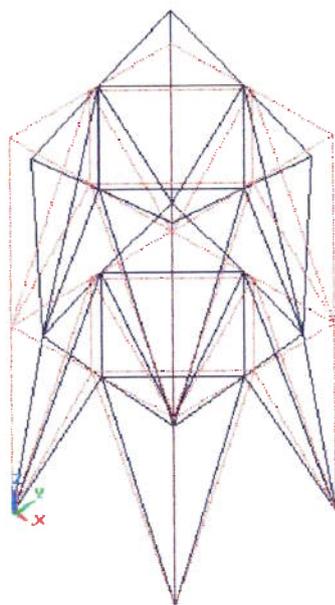
4º modo - frequência = 40,966 Hz



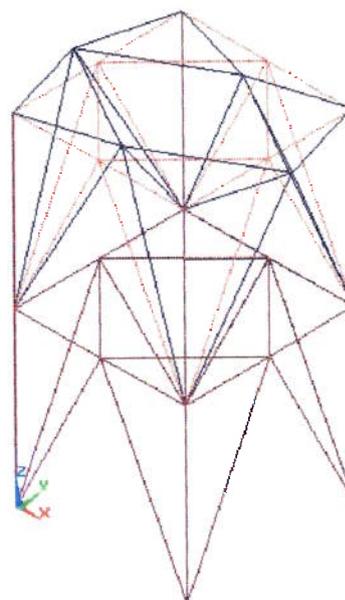
5º modo - frequência = 40,966 Hz



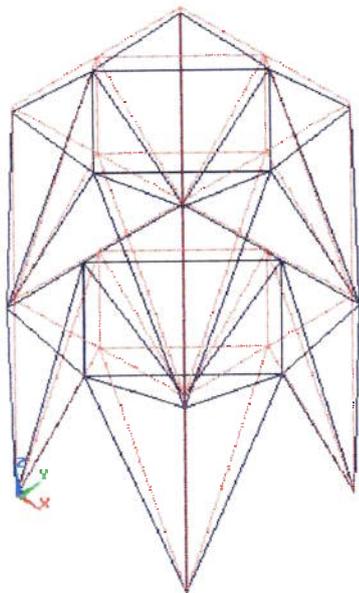
7º modo - frequência = 46,796 Hz



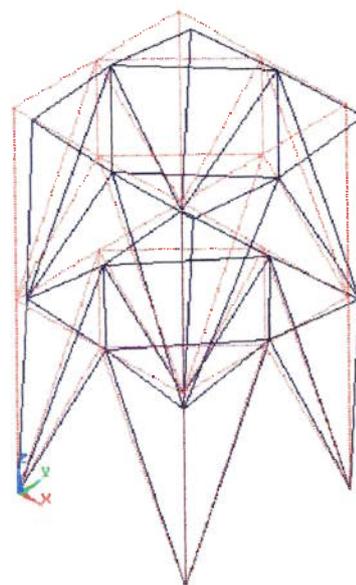
6º modo - frequência = 45,183 Hz



8º modo - frequência = 48,704 Hz



9º modo - frequência = 49,580 Hz



10º modo - frequência = 50,528 Hz

Os valores apresentados a seguir foram usados para a elaboração do conjunto de dados de treinamento e teste para o segundo exemplo. Estes são valores de frequências naturais, em Hertz, para a estrutura danificada.

A apresentação corresponde àquela dos arquivos com as saídas do programa ANSYS 5.0A. Os casos de defeitos podem ser identificados pelo nome do arquivo. Por exemplo, o arquivo TW6579.dat refere-se ao caso de defeito onde o primeiro elemento possui fração de rigidez igual a 0,6, o segundo igual a 0,5, o terceiro igual a 0,7 e o quarto igual a 0,9. Caso um dos elementos não esteja danificado, a fração de rigidez igual a 1,0 será indicada pela letra "X".

## TW0000.DAT

SET	TIME/FREQ
1	7,0795
2	9,2496
3	17,738
4	20,857
5	35,742
6	39,205
7	39,998
8	41,308
9	43,135
10	44,037

## TWOXX4.DAT

SET	TIME/FREQ
1	18,873
2	22,323
3	38,444
4	39,882
5	40,244
6	44,285
7	45,424
8	47,949
9	48,578
10	49,657

## TW1XX9.DAT

SET	TIME/FREQ
1	19,114
2	22,852
3	38,891
4	40,128
5	40,398
6	44,446
7	45,643
8	48,123
9	49,435
10	49,734

## TW050X0.DAT

SET	TIME/FREQ
1	18,226
2	21,499
3	36,843
4	39,202
5	40,091
6	43,463
7	44,517
8	46,380
9	47,311
10	49,251

## TWOXXX.DAT

SET	TIME/FREQ
1	19,140
2	22,934
3	38,648
4	40,007
5	40,270
6	44,346
7	45,492
8	48,099
9	49,572
10	49,665

## TW1XXX.DAT

SET	TIME/FREQ
1	19,145
2	22,942
3	38,917
4	40,147
5	40,404
6	44,453
7	45,653
8	48,138
9	49,576
10	49,736

## TWOX00.DAT

SET	TIME/FREQ
1	13,015
2	18,368
3	21,623
4	37,579
5	40,045
6	40,919
7	44,208
8	44,938
9	46,002
10	48,514

## TW111X.DAT

SET	TIME/FREQ
1	18,931
2	22,289
3	34,812
4	38,980
5	40,034
6	43,382
7	43,804
8	46,136
9	46,941
10	48,955

## TW2287.DAT

SET	TIME/FREQ
1	18,851
2	22,176
3	36,145
4	39,476
5	40,299
6	43,733
7	44,426
8	47,262
9	47,799
10	49,180

## TWOX98.DAT

SET	TIME/FREQ
1	19,069
2	22,736
3	38,568
4	39,945
5	40,253
6	44,319
7	45,469
8	48,056
9	49,116
10	49,658

## TW1X6X.DAT

SET	TIME/FREQ
1	19,131
2	22,884
3	38,836
4	40,025
5	40,358
6	44,401
7	45,625
8	48,098
9	48,892
10	49,731

## TW22XX.DAT

SET	TIME/FREQ
1	19,022
2	22,557
3	36,222
4	39,556
5	40,337
6	43,746
7	44,486
8	47,579
9	48,692
10	49,287

## TW30XX.DAT

SET	TIME/FREQ
1	18,956
2	22,367
3	30,569
4	39,152
5	40,248
6	43,573
7	44,171
8	46,563
9	48,085
10	49,187

## TW466X.DAT

SET	TIME/FREQ
1	19,089
2	22,735
3	38,567
4	39,971
5	40,494
6	44,293
7	45,446
8	47,908
9	48,429
10	49,552

## TW4X99.DAT

SET	TIME/FREQ
1	19,126
2	22,859
3	39,348
4	40,455
5	40,574
6	44,685
7	46,090
8	48,265
9	49,279
10	49,974

## TW3997.DAT

SET	TIME/FREQ
1	19,049
2	22,672
3	39,182
4	40,334
5	40,511
6	44,599
7	45,927
8	48,180
9	49,128
10	49,890

## TW49XX.DAT

SET	TIME/FREQ
1	19,148
2	22,925
3	39,266
4	40,393
5	40,576
6	44,626
7	45,961
8	48,209
9	49,503
10	49,866

## TW4XX4.DAT

SET	TIME/FREQ
1	18,900
2	22,352
3	39,230
4	40,379
5	40,520
6	44,638
7	46,011
8	48,156
9	48,601
10	49,973

## TW41XX.DAT

1	19,013
2	22,496
3	36,247
4	39,770
5	40,414
6	43,782
7	44,591
8	47,493
9	48,586
10	49,324

## TW4X44.DAT

SET	TIME/FREQ
1	18,859
2	22,249
3	39,026
4	40,110
5	40,426
6	44,540
7	45,906
8	47,140
9	48,144
10	49,970

## TW4XXX.DAT

SET	TIME/FREQ
1	19,160
2	22,962
3	39,384
4	40,496
5	40,601
6	44,707
7	46,112
8	48,284
9	49,579
10	49,974

## TW456X.DAT

SET	TIME/FREQ
1	19,072
2	22,683
3	38,241
4	39,902
5	40,476
6	44,193
7	45,267
8	47,807
9	48,302
10	49,466

## TW4X88.DAT

SET	TIME/FREQ
1	19,086
2	22,749
3	39,306
4	40,408
5	40,547
6	44,662
7	46,065
8	48,240
9	48,947
10	49,973

## TW50XX.DAT

SET	TIME/FREQ
1	18,973
2	22,385
3	31,464
4	39,435
5	40,388
6	43,700
7	44,411
8	47,018
9	48,200
10	49,299

## TW5544.DAT

SET	TIME/FREQ
1	18,705
2	21,958
3	38,193
4	39,787
5	40,412
6	44,217
7	45,252
8	46,150
9	47,632
10	49,491

## TW6X5X.DAT

SET	TIME/FREQ
1	19,117
2	22,810
3	39,005
4	40,285
5	40,648
6	44,509
7	45,785
8	48,041
9	49,227
10	49,705

## TW7X7X.DAT

SET	TIME/FREQ
1	19,165
2	22,940
3	39,628
4	40,661
5	40,745
6	44,905
7	46,459
8	48,461
9	49,089
10	50,239

## TW555X.DAT

SET	TIME/FREQ
1	19,073
2	22,672
3	38,438
4	39,946
5	40,529
6	44,255
7	45,418
8	47,808
9	48,063
10	49,518

## TW6X84.DAT

SET	TIME/FREQ
1	18,912
2	22,365
3	39,463
4	40,568
5	40,612
6	44,788
7	46,261
8	48,293
9	48,604
10	50,147

## TW7XX3.DAT

SET	TIME/FREQ
1	18,851
2	22,253
3	39,530
4	40,629
5	40,638
6	44,839
7	46,337
8	48,334
9	48,415
10	50,238

## TW666X.DAT

SET	TIME/FREQ
1	19,100
2	22,747
3	38,940
4	40,142
5	40,619
6	44,452
7	45,761
8	47,991
9	48,431
10	49,678

## TW6XXX.DAT

SET	TIME/FREQ
1	19,170
2	22,976
3	39,586
4	40,681
5	40,718
6	44,868
7	46,382
8	48,409
9	49,580
10	50,148

## TW855X.DAT

SET	TIME/FREQ
1	19,089
2	22,690
3	39,000
4	40,144
5	40,725
6	44,492
7	45,880
8	47,939
9	48,066
10	49,689

## TW66XX.DAT

SET	TIME/FREQ
1	19,117
2	22,810
3	39,005
4	40,285
5	40,648
6	44,509
7	45,785
8	48,041
9	49,227
10	49,705

## TW732X.DAT

SET	TIME/FREQ
1	19,027
2	22,501
3	37,994
4	39,727
5	40,589
6	44,130
7	45,242
8	46,895
9	47,483
10	49,432

## TW859X.DAT

SET	TIME/FREQ
1	19,108
2	22,757
3	39,069
4	40,316
5	40,759
6	44,555
7	45,920
8	48,009
9	48,947
10	49,716

## TW85XX.DAT

SET	TIME/FREQ
1	19,112
2	22,772
3	39,081
4	40,346
5	40,765
6	44,568
7	45,926
8	48,021
9	49,127
10	49,726

## TW89X9.DAT

SET	TIME/FREQ
1	19,136
2	22,860
3	39,651
4	40,718
5	40,809
6	44,939
7	46,485
8	48,447
9	49,358
10	50,191

## TW8XXX.DAT

SET	TIME/FREQ
1	19,179
2	22,988
3	39,730
4	40,836
5	40,841
6	45,028
7	46,611
8	48,552
9	49,580
10	50,333

## TW86XX.DAT

SET	TIME/FREQ
1	19,127
2	22,822
3	39,293
4	40,431
5	40,783
6	44,673
7	46,090
8	48,136
9	49,233
10	49,819

## TW8X54.DAT

SET	TIME/FREQ
1	18,892
2	22,295
3	39,517
4	40,490
5	40,642
6	44,839
7	46,358
8	47,463
9	48,450
10	50,331

## TW9485.DAT

SET	TIME/FREQ
1	18,821
2	22,121
3	38,847
4	40,212
5	40,708
6	44,491
7	45,812
8	47,468
9	47,838
10	49,652

## TW888X.DAT

SET	TIME/FREQ
1	19,148
2	22,882
3	39,554
4	40,562
5	40,796
6	44,833
7	46,358
8	48,348
9	49,075
10	50,048

## TW8X5X.DAT

SET	TIME/FREQ
1	19,162
2	22,915
3	39,667
4	40,620
5	40,786
6	44,951
7	46,519
8	48,474
9	48,742
10	50,332

## TW9595.DAT

SET	TIME/FREQ
1	18,861
2	22,208
3	39,111
4	40,316
5	40,734
6	44,605
7	45,988
8	47,868
9	48,034
10	49,753

## TW88XX.DAT

SET	TIME/FREQ
1	19,155
2	22,911
3	39,575
4	40,629
5	40,813
6	44,863
7	46,378
8	48,357
9	49,421
10	50,052

## TW8X88.DAT

SET	TIME/FREQ
1	19,107
2	22,776
3	39,681
4	40,735
5	40,781
6	44,974
7	46,545
8	48,515
9	48,951
10	50,333

## TW96XX.DAT

SET	TIME/FREQ
1	19,132
2	22,828
3	39,411
4	40,498
5	40,850
6	44,755
7	46,232
8	48,192
9	49,236
10	49,873

## TW9955.DAT

SET	TIME/FREQ
1	18,937
2	22,368
3	39,563
4	40,480
5	40,716
6	44,867
7	46,394
8	47,537
9	48,455
10	50,263

## TW9XX9.DAT

SET	TIME/FREQ
1	19,154
2	22,905
3	39,776
4	40,884
5	40,887
6	45,094
7	46,691
8	48,618
9	49,438
10	50,429

## TWX111.DAT

SET	TIME/FREQ
1	17,915
2	21,122
3	37,213
4	39,396
5	39,983
6	41,851
7	44,097
8	45,175
9	46,543
10	49,260

## TW999X.DAT

SET	TIME/FREQ
1	19,169
2	22,943
3	39,719
4	40,772
5	40,882
6	45,013
7	46,598
8	48,530
9	49,344
10	50,271

## TW9XXX.DAT

SET	TIME/FREQ
1	19,184
2	22,995
3	39,786
4	40,903
5	40,903
6	45,106
7	46,709
8	48,628
9	49,580
10	50,430

## TWX11X.DAT

SET	TIME/FREQ
1	18,990
2	22,338
3	37,617
4	39,559
5	40,729
6	44,078
7	45,049
8	46,288
9	47,085
10	49,406

## TW9X55.DAT

SET	TIME/FREQ
1	18,959
2	22,417
3	39,626
4	40,575
5	40,726
6	44,936
7	46,481
8	47,683
9	48,561
10	50,428

## TWX000.DAT

SET	TIME/FREQ
1	9,7093
2	17,472
3	20,832
4	31,869
5	38,381
6	40,525
7	41,845
8	43,920
9	44,543
10	46,520

## TWX1XX.DAT

SET	TIME/FREQ
1	19,050
2	22,529
3	38,040
4	40,179
5	40,813
6	44,251
7	45,458
8	47,577
9	48,640
10	49,539

## TW9XX1.DAT

SET	TIME/FREQ
1	18,695
2	22,012
3	39,634
4	40,677
5	40,730
6	44,899
7	46,409
8	48,016
9	48,405
10	50,428

## TWX0XX.DAT

SET	TIME/FREQ
1	19,006
2	22,414
3	32,583
4	39,731
5	40,775
6	43,939
7	44,990
8	47,366
9	48,317
10	49,490

## TWX44X.DAT

SET	TIME/FREQ
1	19,077
2	22,625
3	39,035
4	40,109
5	40,826
6	44,526
7	45,966
8	47,654
9	47,879
10	49,683

## TWX45X.DAT

SET	TIME/FREQ
1	19,082
2	22,644
3	39,063
4	40,180
5	40,838
6	44,545
7	45,992
8	47,885
9	47,940
10	49,691

## TWX55X.DAT

SET	TIME/FREQ
1	19,100
2	22,701
3	39,284
4	40,255
5	40,853
6	44,652
7	46,158
8	48,047
9	48,077
10	49,791

## TWX6XX.DAT

SET	TIME/FREQ
1	19,137
2	22,834
3	39,512
4	40,563
5	40,915
6	44,838
7	46,365
8	48,255
9	49,238
10	49,927

## TWX4XX.DAT

SET	TIME/FREQ
1	19,106
2	22,729
3	39,137
4	40,384
5	40,882
6	44,625
7	46,052
8	47,985
9	49,022
10	49,732

## TWX5XX.DAT

SET	TIME/FREQ
1	19,122
2	22,784
3	39,351
4	40,469
5	40,899
6	44,735
7	46,217
8	48,121
9	49,134
10	49,821

## TWX88X.DAT

SET	TIME/FREQ
1	19,157
2	22,895
3	39,700
4	40,701
5	40,923
6	44,992
7	46,584
8	48,497
9	49,075
10	50,190

## TWX549.DAT

SET	TIME/FREQ
1	19,054
2	22,579
3	39,239
4	40,167
5	40,818
6	44,627
7	46,112
8	47,571
9	48,032
10	49,781

## TWX694.DAT

SET	TIME/FREQ
1	18,824
2	22,154
3	39,393
4	40,438
5	40,786
6	44,761
7	46,222
8	47,889
9	48,133
10	49,908

## TWX8X4.DAT

SET	TIME/FREQ
1	18,888
2	22,289
3	39,635
4	40,654
5	40,822
6	44,940
7	46,463
8	48,390
9	48,403
10	50,186

## TWX555.DAT

SET	TIME/FREQ
1	18,834
2	22,138
3	39,157
4	40,186
5	40,733
6	44,613
7	46,000
8	46,876
9	47,960
10	49,771

## TWX694.DAT

SET	TIME/FREQ
1	18,824
2	22,154
3	39,393
4	40,438
5	40,786
6	44,761
7	46,222
8	47,889
9	48,133
10	49,908

## TWX8XX.DAT

SET	TIME/FREQ
1	19,164
2	22,923
3	39,718
4	40,768
5	40,943
6	45,023
7	46,609
8	48,502
9	49,422
10	50,192

## TWX999.DAT

SET	TIME/FREQ
1	19,142
2	22,858
3	39,765
4	40,821
5	40,926
6	45,080
7	46,677
8	48,598
9	49,192
10	50,351

## TWXX00.DAT

SET	TIME/FREQ
1	18,459
2	21,677
3	32,378
4	39,031
5	40,426
6	43,511
7	44,971
8	45,207
9	48,350
10	50,430

## TWXX09.DAT

SET	TIME/FREQ
1	19,115
2	22,741
3	38,884
4	39,654
5	40,785
6	44,982
7	45,984
8	47,127
9	48,702
10	50,502

## TWX99X.DAT

SET	TIME/FREQ
1	19,173
2	22,950
3	39,774
4	40,839
5	40,945
6	45,091
7	46,695
8	48,608
9	49,344
10	50,351

## TWXX04.DAT

SET	TIME/FREQ
1	18,860
2	22,199
3	38,697
4	39,431
5	40,609
6	44,780
7	45,769
8	45,950
9	48,596
10	50,498

## TWXX0X.DAT

SET	TIME/FREQ
1	19,148
2	22,835
3	38,905
4	39,682
5	40,815
6	44,999
7	46,008
8	47,351
9	48,716
10	50,503

## TWX9X7.DAT

SET	TIME/FREQ
1	19,070
2	22,675
3	39,752
4	40,814
5	40,898
6	45,068
7	46,647
8	48,575
9	49,042
10	50,350

## TWXX05.DAT

SET	TIME/FREQ
1	18,926
2	22,317
3	38,753
4	39,489
5	40,649
6	44,852
7	45,841
8	46,169
9	48,626
10	50,499

## TWXX14.DAT

SET	TIME/FREQ
1	18,870
2	22,223
3	39,486
4	39,853
5	40,643
6	44,836
7	46,109
8	46,337
9	48,604
10	50,511

## TWX9XX.DAT

SET	TIME/FREQ
1	19,177
2	22,963
3	39,783
4	40,869
5	40,955
6	45,106
7	46,709
8	48,610
9	49,504
10	50,352

## TWXX08.DAT

SET	TIME/FREQ
1	19,076
2	22,643
3	38,859
4	39,621
5	40,754
6	44,962
7	45,957
8	46,895
9	48,687
10	50,502

## TWXX1X.DAT

SET	TIME/FREQ
1	19,154
2	22,856
3	39,691
4	40,020
5	40,836
6	45,023
7	46,379
8	47,640
9	48,724
10	50,516

## TWXX4X.DAT

SET	TIME/FREQ
1	19,167
2	22,911
3	39,761
4	40,672
5	40,890
6	45,084
7	46,655
8	48,391
9	48,755
10	50,524

## TWXX85.DAT

SET	TIME/FREQ
1	18,983
2	22,475
3	39,750
4	40,788
5	40,841
6	45,067
7	46,639
8	48,395
9	48,643
10	50,523

## TWXX9X.DAT

SET	TIME/FREQ
1	19,185
2	22,987
3	39,823
4	40,936
5	40,955
6	45,168
7	46,781
8	48,701
9	49,428
10	50,528

## TWXX5X.DAT

SET	TIME/FREQ
1	19,171
2	22,928
3	39,777
4	40,752
5	40,905
6	45,102
7	46,692
8	48,570
9	48,802
10	50,525

## TWXX89.DAT

SET	TIME/FREQ
1	19,151
2	22,883
3	39,804
4	40,882
5	40,925
6	45,140
7	46,744
8	48,684
9	49,113
10	50,527

## TWXXX0.DAT

SET	TIME/FREQ
1	18,595
2	21,885
3	39,669
4	40,688
5	40,770
6	44,791
7	46,388
8	47,764
9	48,423
10	50,517

## TWXX67.DAT

SET	TIME/FREQ
1	19,068
2	22,657
3	39,750
4	40,748
5	40,853
6	45,073
7	46,653
8	48,321
9	48,693
10	50,524

## TWXX8X.DAT

SET	TIME/FREQ
1	19,182
2	22,973
3	39,814
4	40,901
5	40,944
6	45,153
7	46,764
8	48,696
9	49,266
10	50,527

## TWXXX1.DAT

SET	TIME/FREQ
1	18,703
2	22,019
3	39,701
4	40,725
5	40,795
6	44,964
7	46,480
8	48,020
9	48,494
10	50,520

## TWXX6X.DAT

SET	TIME/FREQ
1	19,175
2	22,944
3	39,790
4	40,813
5	40,919
6	45,120
7	46,721
8	48,658
9	48,926
10	50,526

## TWXX96.DAT

SET	TIME/FREQ
1	19,038
2	22,600
3	39,780
4	40,849
5	40,880
6	45,107
7	46,690
8	48,644
9	48,795
10	50,525

## TWXXX4.DAT

SET	TIME/FREQ
1	18,937
2	22,391
3	39,764
4	40,821
5	40,860
6	45,079
7	46,642
8	48,609
9	48,613
10	50,524

## TWXXX5.DAT

SET	TIME/FREQ
1	18,993
2	22,506
3	39,779
4	40,849
5	40,880
6	45,103
7	46,677
8	48,636
9	48,790
10	50,525

## TWXXX8.DAT

SET	TIME/FREQ
1	19,125
2	22,817
3	39,815
4	40,923
5	40,933
6	45,157
7	46,756
8	48,684
9	49,288
10	50,527

## TWXXX6.DAT

SET	TIME/FREQ
1	19,043
2	22,615
3	39,792
4	40,875
5	40,898
6	45,123
7	46,707
8	48,655
9	48,964
10	50,526

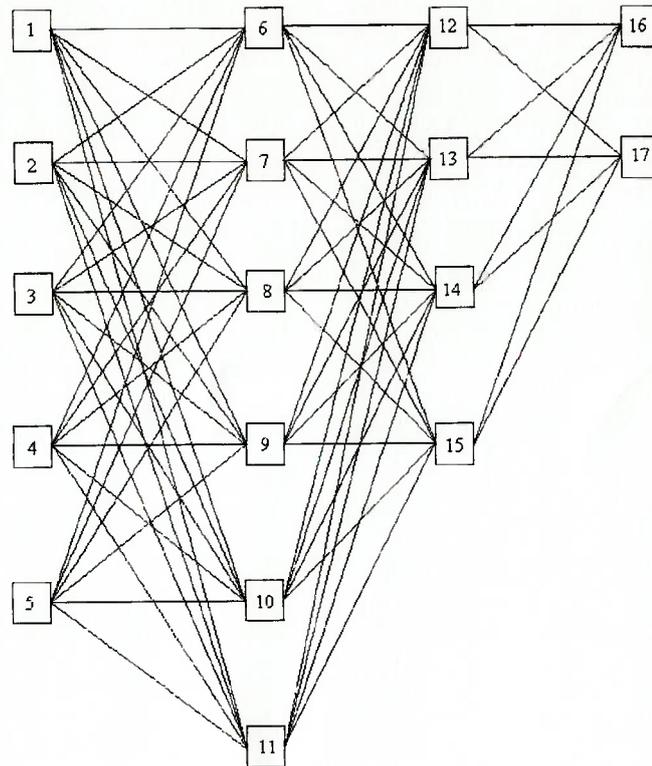
## TWXXX9.DAT

SET	TIME/FREQ
1	19,158
2	22,911
3	39,824
4	40,945
5	40,950
6	45,171
7	46,777
8	48,695
9	49,438
10	50,528

**APÊNDICE II**

### Pesos da rede neural do exemplo - I

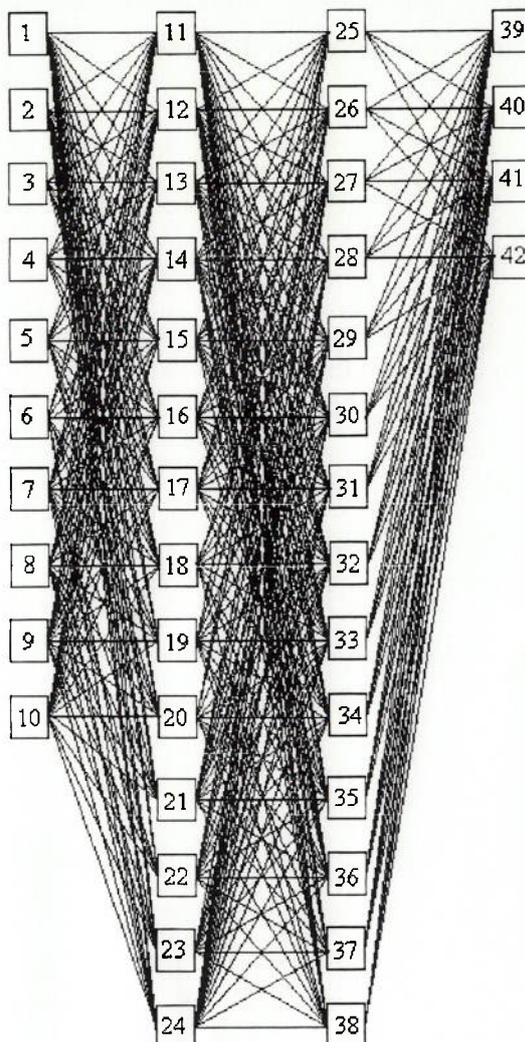
Os pesos listados seguem a convenção do desenho abaixo. Por exemplo, o peso  $w-1-7$  corresponde a ligação entre o neurônio 1, localizado na camada de entrada, e o neurônio 7, da primeira camada oculta.



PESO	VALOR	PESO	VALOR	PESO	VALOR	PESO	VALOR
w-1-6	-6.0199	w-1-7	-10.5316	w-1-8	-0.0757	w-1-9	11.7229
w-2-6	0.3845	w-2-7	7.8426	w-2-8	2.4741	w-2-9	2.4629
w-3-6	1.0496	w-3-7	-1.3403	w-3-9	-1.1950	w-3-9	-3.2076
w-4-6	2.8573	w-4-7	1.9025	w-4-9	-1.6480	w-4-9	-7.6751
w-5-6	1.5584	w-5-7	1.7457	w-5-9	0.4204	w-5-9	-3.4517
w-1-10	-4.4753	w-1-11	6.4500				
w-2-10	0.8516	w-2-11	-2.5909				
w-3-10	0.0960	w-3-11	-0.0736				
w-4-10	0.3026	w-4-11	-3.0305				
w-5-10	2.5707	w-5-11	-1.1081				

PESO	VALOR	PESO	VALOR	PESO	VALOR	PESO	VALOR
w-6-12	-0.7138	w-6-13	0.2039	w-6-14	0.4524	w-6-15	-0.9563
w-7-12	-2.0497	w-7-13	0.0501	w-7-14	0.2978	w-7-15	-0.9138
w-8-12	-0.3098	w-8-13	1.1695	w-8-14	0.9331	w-8-15	0.5925
w-9-12	1.2428	w-9-13	0.3091	w-9-14	-0.1916	w-9-15	2.2228
w-10-12	0.0521	w-10-13	-0.3004	w-10-14	0.3144	w-10-15	-1.2527
w-11-12	1.1194	w-11-13	0.6111	w-11-14	0.4355	w-11-15	0.7922
w-12-16	-10.5002	w-12-17	-1.3548				
w-13-16	-0.3201	w-13-17	-0.9611				
w-14-16	0.2840	w-14-17	-1.0111				
w-15-16	0.5632	w-15-17	1.3585				

### Pesos da rede neural do exemplo - II



PESO	VALOR	PESO	VALOR	PESO	VALOR	PESO	VALOR
w-1-11	0,1776	w-1-12	-0,2164	w-1-13	-0,2172	w-1-14	-5,6536
w-2-11	0,2386	w-2-12	-0,0530	w-2-13	-0,1879	w-2-14	-9,5692
w-3-11	0,9501	w-3-12	1,1762	w-3-13	-1,6471	w-3-14	-1,4913
w-4-11	-0,0917	w-4-12	0,7383	w-4-13	1,4443	w-4-14	-0,5338
w-5-11	0,6340	w-5-12	-1,0521	w-5-13	-0,6504	w-5-14	-9,2826
w-6-11	-0,5156	w-6-12	-0,4881	w-6-13	-0,6124	w-6-14	2,1805
w-7-11	-2,8135	w-7-12	3,2284	w-7-13	-0,9194	w-7-14	3,5245
w-8-11	1,5341	w-8-12	-1,4742	w-8-13	3,6450	w-8-14	-0,2053
w-9-11	0,4611	w-9-12	-0,4891	w-9-13	1,6628	w-9-14	-4,1079
w-10-11	-0,2532	w-10-12	0,6377	w-10-13	-2,3965	w-10-14	0,0322

PESO	VALOR	PESO	VALOR	PESO	VALOR	PESO	VALOR
w-1-15	-2,1246	w-1-16	1,7286	w-1-17	0,0783	w-1-18	-0,8522
w-2-15	-3,3702	w-2-16	2,8854	w-2-17	0,2814	w-2-18	-1,3097
w-3-15	0,2570	w-3-16	-0,0045	w-3-17	1,4575	w-3-18	1,0455
w-4-15	0,2083	w-4-16	-0,0553	w-4-17	-1,1037	w-4-18	-0,6146
w-5-15	-3,0660	w-5-16	1,4942	w-5-17	-0,4728	w-5-18	-1,3814
w-6-15	-0,0334	w-6-16	0,6487	w-6-17	0,5988	w-6-18	0,1714
w-7-15	-0,4442	w-7-16	-2,2580	w-7-17	0,4712	w-7-18	0,2111
w-8-15	2,8614	w-8-16	0,8279	w-8-17	-2,3091	w-8-18	-0,4108
w-9-15	-1,9830	w-9-16	2,0979	w-9-17	-1,3553	w-9-18	-1,6548
w-10-15	-1,0098	w-10-16	-0,7759	w-10-17	1,4968	w-10-18	0,6960
w-1-19	-0,1505	w-1-20	-0,2049	w-1-21	-1,7860	w-1-22	0,0102
w-2-19	-0,2121	w-2-20	-0,4305	w-2-21	-1,7990	w-2-22	0,2113
w-3-19	0,0051	w-3-20	0,4305	w-3-21	3,6905	w-3-22	-0,6969
w-4-19	1,7497	w-4-20	0,0594	w-4-21	0,9425	w-4-22	0,2478
w-5-19	-1,1728	w-5-20	-0,3682	w-5-21	-5,3650	w-5-22	1,0050
w-6-19	0,5785	w-6-20	0,1844	w-6-21	-4,9219	w-6-22	-0,3341
w-7-19	-0,8335	w-7-20	-0,2547	w-7-21	-2,5670	w-7-22	-1,5197
w-8-19	4,2679	w-8-20	0,7192	w-8-21	0,9381	w-8-22	0,8875
w-9-19	1,0884	w-9-20	-0,6795	w-9-21	-1,4074	w-9-22	1,1566
w-10-19	-1,6498	w-10-20	-0,2151	w-10-21	-1,8196	w-10-22	-0,3980
w-1-23	0,4707	w-1-24	-1,7509				
w-2-23	1,4759	w-2-24	-2,1771				
w-3-23	-0,8380	w-3-24	0,5055				
w-4-23	-0,0320	w-4-24	0,3678				
w-5-23	-1,5650	w-5-24	4,5630				
w-6-23	0,5133	w-6-24	-0,9840				
w-7-23	-4,2969	w-7-24	-1,0786				
w-8-23	3,3622	w-8-24	-8,4528				
w-9-23	2,4018	w-9-24	-0,6624				
w-10-23	-1,7765	w-10-24	4,8037				

PESO	VALOR	PESO	VALOR	PESO	VALOR	PESO	VALOR
w-11-25	-0,1035	w-11-26	-0,0715	w-11-27	-0,1682	w-11-28	0,2656
w-12-25	-0,0927	w-12-26	-0,0907	w-12-27	0,6640	w-12-28	-0,0252
w-13-25	-0,2078	w-13-26	-0,1830	w-13-27	0,0679	w-13-28	0,4109
w-14-25	-0,0388	w-14-26	-0,2579	w-14-27	0,2203	w-14-28	0,5221
w-15-25	0,0756	w-15-26	0,1436	w-15-27	-0,1533	w-15-28	-0,0521
w-16-25	0,9660	w-16-26	-0,3168	w-16-27	0,3598	w-16-28	0,3211
w-17-25	0,4595	w-17-26	0,0929	w-17-27	0,1533	w-17-28	-0,2797
w-18-25	-0,4132	w-18-26	0,2022	w-18-27	-0,0134	w-18-28	-0,2360
w-19-25	0,9130	w-19-26	-0,0522	w-19-27	-0,3431	w-19-28	0,4073
w-20-25	0,6361	w-20-26	-0,0649	w-20-27	0,2001	w-20-28	0,0953
w-21-25	-0,9997	w-21-26	-0,1331	w-21-27	0,1538	w-21-28	0,3166
w-21-25	-0,8422	w-21-26	0,0338	w-21-27	-0,2793	w-21-28	0,1182
w-23-25	-0,9119	w-23-26	-0,1837	w-23-27	0,1040	w-23-28	0,3531
w-24-25	-1,0000	w-24-26	-0,0671	w-24-27	0,3379	w-24-28	0,4665
w-11-29	-0,2039	w-11-30	-0,4270	w-11-31	0,1770	w-11-32	-0,1183
w-12-29	-1,3067	w-12-30	1,6960	w-12-31	1,2102	w-12-32	-0,9855
w-13-29	0,8788	w-13-30	0,1422	w-13-31	0,3032	w-13-32	0,6743
w-14-29	-1,5032	w-14-30	1,8097	w-14-31	0,0466	w-14-32	0,1913
w-15-29	-1,1067	w-15-30	-0,1838	w-15-31	-0,1157	w-15-32	0,6110
w-16-29	0,9550	w-16-30	0,2433	w-16-31	0,0538	w-16-32	-0,1430
w-17-29	-0,7870	w-17-30	-0,0078	w-17-31	-0,1180	w-17-32	-0,4646
w-18-29	-0,9294	w-18-30	-0,2567	w-18-31	-0,0267	w-18-32	-0,0149
w-19-29	0,3006	w-19-30	0,5406	w-19-31	0,2857	w-19-32	-0,6414
w-20-29	-0,3015	w-20-30	-0,2293	w-20-31	-0,0598	w-20-32	0,0826
w-21-29	-0,9476	w-21-30	0,6330	w-21-31	-0,6618	w-21-32	1,9445
w-21-29	0,4062	w-21-30	0,0711	w-21-31	0,0887	w-21-32	0,1056
w-23-29	0,6718	w-23-30	0,5438	w-23-31	-0,3039	w-23-32	0,3897
w-24-29	-0,6750	w-24-30	-1,9767	w-24-31	-0,7404	w-24-32	-1,1375

PESO	VALOR	PESO	VALOR	PESO	VALOR	PESO	VALOR
w-11-33	-0.2596	w-11-34	0.0577	w-11-35	0.1020	w-11-36	-0.3164
w-12-33	0.0922	w-12-34	-0.8476	w-12-35	0.2071	w-12-36	0.5642
w-13-33	-0.8711	w-13-34	-0.3025	w-13-35	0.4750	w-13-36	-0.9350
w-14-33	-0.1582	w-14-34	0.4811	w-14-35	-2.4314	w-14-36	-1.8680
w-15-33	-0.5349	w-15-34	-0.3462	w-15-35	-1.0106	w-15-36	-1.5395
w-16-33	-0.2079	w-16-34	0.4818	w-16-35	1.1324	w-16-36	0.5092
w-17-33	0.6263	w-17-34	0.4341	w-17-35	-0.3055	w-17-36	0.6336
w-18-33	-0.3069	w-18-34	0.0412	w-18-35	-0.6512	w-18-36	-0.2683
w-19-33	-0.8002	w-19-34	1.1226	w-19-35	-0.2690	w-19-36	0.0706
w-20-33	-0.3290	w-20-34	-0.0724	w-20-35	-0.1967	w-20-36	-0.2503
w-21-33	0.4598	w-21-34	-0.0535	w-21-35	-0.6233	w-21-36	-0.1288
w-21-33	-0.0490	w-21-34	-0.2340	w-21-35	0.2886	w-21-36	-0.1619
w-23-33	-0.2043	w-23-34	0.4550	w-23-35	0.9689	w-23-36	-0.3519
w-24-33	0.7130	w-24-34	-0.2769	w-24-35	-0.0187	w-24-36	0.0559
w-11-37	-0.0266	w-11-38	0.4943				
w-12-37	-1.2787	w-12-38	-0.3095				
w-13-37	-0.0243	w-13-38	0.1375				
w-14-37	-1.4031	w-14-38	-0.4172				
w-15-37	-0.6317	w-15-38	0.1217				
w-16-37	0.1008	w-16-38	-0.1918				
w-17-37	-0.1787	w-17-38	-0.1977				
w-18-37	-0.2823	w-18-38	-0.0298				
w-19-37	0.5604	w-19-38	0.4735				
w-20-37	-0.2028	w-20-38	-0.0951				
w-21-37	1.1263	w-21-38	0.2547				
w-21-37	0.4455	w-21-38	0.2356				
w-23-37	0.3861	w-23-38	0.0430				
w-24-37	0.1308	w-24-38	-0.0698				

PESO	VALOR	PESO	VALOR	PESO	VALOR	PESO	VALOR
w-25-39	-0.0476	w-11-40	0.0141	w-11-41	0.1114	w-11-42	-0.0811
w-26-39	-0.2505	w-12-40	0.1039	w-12-41	0.1826	w-12-42	-0.0911
w-27-39	0.0254	w-13-40	-0.4695	w-13-41	0.0891	w-13-42	0.5097
w-28-39	0.3830	w-14-40	-0.2432	w-14-41	-0.4686	w-14-42	-0.0151
w-29-39	0.1712	w-15-40	-0.0221	w-15-41	-1.2035	w-15-42	0.0424
w-30-39	0.6901	w-16-40	-0.3835	w-16-41	0.1875	w-16-42	0.2606
w-31-39	-0.6902	w-17-40	-1.2817	w-17-41	-1.0013	w-17-42	-0.5389
w-32-39	0.5910	w-18-40	-0.7108	w-18-41	0.6765	w-18-42	-0.0926
w-33-39	0.5779	w-19-40	-0.7819	w-19-41	0.1022	w-19-42	0.2067
w-34-39	0.8245	w-20-40	-0.1089	w-20-41	-0.1198	w-20-42	-0.6589
w-35-39	0.1435	w-21-40	-0.6430	w-21-41	1.0799	w-21-42	-0.1784
w-36-39	-0.1378	w-21-40	0.1784	w-21-41	0.7226	w-21-42	-0.6290
w-37-39	0.1721	w-23-40	0.9253	w-23-41	-0.9237	w-23-42	-0.2855
w-38-39	-0.1022	w-24-40	0.3863	w-24-41	-0.2489	w-24-42	-0.3293