Campus de São Carlos

**IDENTIFICAÇÃO DA DINÂMICA NÃO
LINEAR DE UMA PÁ DE HELICÓPTERO
VIA REDES NEURAIS**

**LUCIANE DE FÁTIMA RODRIGUES DE
SOUZA**

Orientador: Prof. Dr. Eduardo Morgado Belo

UNIVERSIDADE DE SÃO PAULO

**ESCOLA DE ENGENHARIA
DE SÃO CARLOS**

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA MECÂNICA

IDENTIFICAÇÃO DA DINÂMICA NÃO LINEAR DE UMA PÁ
DE HELICÓPTERO VIA REDES NEURAIS

Luciane de Fátima Rodrigues de Souza

Orientador: Prof. Dr. Eduardo Morgado Belo

Serviço de Pós-Graduação EESC/USP

EXEMPLAR REVISADO

Data de entrada no Serviço: 20.1.12.1.02

Ass.: *Luciane de Fátima Rodrigues de Souza*

Dissertação apresentada à Escola de Engenharia de São Carlos, Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia Mecânica

São Carlos
- 2002 -

DEDALUS - Acervo - EESC

31100037144

Class. TESE-EESC
Cutt. 5829
Tombo TQ27/03
Sysno 1296551

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

S729i Souza, Luciane de Fátima Rodrigues de
Identificação da dinâmica não linear de uma pá de
helicóptero via redes neurais / Luciane de Fátima
Rodrigues de Souza. -- São Carlos, 2002.

Dissertação (Mestrado) -- Escola de Engenharia de São
Carlos-Universidade de São Paulo, 2002.
Área: Engenharia Mecânica.
Orientador: Prof. Dr. Eduardo Morgado Belo.

1. Pá de helicóptero. 2. Identificação. 3. Sistemas
não lineares. 4. Redes neurais. I. Título.

FOLHA DE JULGAMENTO

Candidata: Licenciada **LUCIANE DE FATIMA RODRIGUES DE SOUZA**

Dissertação defendida e julgada em 28-10-2002 perante a Comissão Julgadora:

Prof. Assoc. **EDUARDO MORGADO BELO** (Orientador)
(Escola de Engenharia de São Carlos/USP)

aprovada

Prof. Dr. **ZHAO LIANG**
(Instituto de Ciências Matemáticas e de Computação/USP)

aprovada

Prof. Dr. **ATAIR RIOS NETO**
(Instituto Nacional de Pesquisas Espaciais/INPE)

aprovada

Prof. Associado **ARTHUR JOSÉ VIEIRA PORTO**
Coordenador do Programa de Pós-Graduação
em Engenharia Mecânica

Profa. Assoc. **MARIA DO CARMO CALIJURI**
Presidente da Comissão de Pós-Graduação

*Aos meus pais João Batista e Yolanda,
que tanto amo, dedico esta minha conquista.*

AGRADECIMENTOS

Agradeço a Deus por tudo que fez e faz por mim, em todos os momentos de minha vida.

Ao meu orientador Prof. Dr. Eduardo Morgado Belo pela orientação e pela paciência nos momentos difíceis.

Ao Prof. Dr. Flávio pela paciência, amizade e pelas importantes contribuições no desenvolvimento do trabalho todo.

Ao Prof. Dr. Zhao Liang pelas contribuições valiosas ao trabalho.

Aos meus pais novamente e aos meus irmãos José Ariovaldo, Fernanda e João Paulo pelo estímulo e carinho dedicados.

A Kelen, a Gi, desde a graduação, e a Luciana por toda amizade e dedicação. Não tenho palavras para agradecê-las, por tudo que fizeram por mim desde quando retornei. Que Deus as abençoe muito! Agradeço também a Kelen pelas importantes dicas dadas no trabalho com as redes neurais.

Ao Carlos Demarqui por toda a amizade, paciência e ajuda quando precisei.

Ao Édson por toda a amizade e ajuda no trabalho com as redes neurais e no *Matlab*.

A todos os meus amigos do laboratório: Alexandre, Cláudio, Daniela, Elizângela, Gasparini, Guilherme, Luís Fernando, Márcio, Naga, Valdinei, Werner por todos os momentos que passamos. A todos os meus amigos: Fabiana, Jaque, Rodrigo, Geraldo, Toddy e todos os amigos da USP.

À Ana Paula e a Bete por sempre terem me ajudado com tudo que precisei.

À minha tia Simíria e à minha prima Valéria por tudo.

Aos meus amigos de Cerqueira César: Alessandra, Angelinha, Bico, Cláudia, Dani, Joyce, Kiko, Luciana, Magda, Marcinha, Mazinha, Myleny, Myrely, Viviane e Zinho pela amizade.

Ao Fabiano por todo incentivo e apoio que me deu, principalmente no início.

Ao Luís Carlos e ao Ricardo por todo o incentivo que me deram.

SUMÁRIO

LISTA DE FIGURAS.....	iv
LISTA DE TABELAS.....	vii
LISTA DE SÍMBOLOS	viii
RESUMO.....	xiii
ABSTRACT.....	xiv
CAPÍTULO 1	
Introdução e revisão da literatura	
1.1 - Objetivos do Trabalho.....	3
1.2 - Identificação de Sistemas não lineares e Redes Neurais Artificiais.....	3
1.3 - Sistemas Bilineares	13
1.4 - Organização da Dissertação	14
CAPÍTULO 2	
Redes Neurais Multicamadas	
2.1 - Introdução.....	16
2.2 - Redes Neurais artificiais.....	17
2.3 - Modelo de um neurônio	17
2.3.1 - Tipos de Função de Ativação	19
2.4 - Redes Neurais Multicamadas	21
2.4.1 - Uma camada de neurônios	23

2.4.2 - Múltiplas camadas de neurônios.....	24
2.5 - Aprendizagem de uma rede neural	26
2.5.1 - Aprendizado Supervisionado.....	26
2.6 - Algoritmo <i>Backpropagation</i> (Retropropagação)	27
2.7 - O Algoritmo de Levenberg-Marquardt	30
2.7.1 - Algoritmo básico	30
2.7.2 - Cálculo da Jacobiana.....	34
2.7.3 - Resumo do algoritmo de Levenberg -Marquardt.....	38
2.8 - Conclusão	38

CAPÍTULO 3

O Modelo Matemático da pá

3.1 - Introdução.....	39
3.2 - Sistemas de Coordenadas	39
3.3 - Energia de Deformação e Energia Cinética.....	42
3.3.1 - Energia de Defomação	42
3.3.2 - Energia Cinética	45
3.4 - Método dos Elementos Finitos	47
3.5 - Carregamento Aerodinâmico.....	50
3.6 - Sistemas Bilineares	52
3.6.1 - Aplicação da teoria de sistemas bilineares à modelagem da pá	54
3.7 - Representação do Sistema no espaço de estados	57
3.8 - Conclusões	59

CAPÍTULO 4

Simulação e discussão dos resultados

4.1 - Introdução.....	60
4.2 – Simulação do modelo	61
4.3 – Detecção de não linearidades - Testes no Domínio do Tempo	63

4.3.1 - Aplicação do teste ao modelo da pá.....	63
4.4 – Rede Neural na modelagem	67
4.5 - Modelagem Da Pá Utilizando Rede Neural	69
4.6 - Resultados e discussões.....	71
4.7 - Uma análise qualitativa da identificação realizada	77
4.7.1 - Mapas iterativos.....	77
4.7.2 - Mapa de Retorno	78
4.8 - Uma análise quantitativa da identificação realizada	80
CAPÍTULO 5	
Considerações Finais	
5.1 - Conclusões finais	86
5.2 – Sugestões para trabalhos futuros	87
REFERÊNCIAS BIBLIOGRÁFICAS.....	88

LISTA DE FIGURAS

CAPÍTULO 1

Figura 1.1 - Identificação de um sistema dinâmico	4
---	---

CAPÍTULO 2

Figura 2.1 - Modelo não linear de um neurônio	18
Figura 2.2 - Função <i>Degrau</i>	19
Figura 2.3 - Função linear por partes.....	20
Figura 2.4 - Função sigmoidal logística.	20
Figura 2.5 - Função tangente hiperbólica.	21
Figura 2.6 - Função linear.....	21
Figura 2.7 - Camada de S neurônios (HAGAN et al., 1996)	23
Figura 2.8 - Camada de S neurônios, notação abreviada (HAGAN et al., 1996).....	23
Figura 2.9 - Rede com três camadas (HAGAN et al., 1996)	24
Figura 2.10 - Rede com três camadas, notação abreviada (HAGAN et al., 1996).....	25
Figura 2.11- Diagrama de blocos que representa um sistema com aprendizado supervisionado	27

CAPÍTULO 3

Figura 3.1 - Sistemas de coordenadas principais da pá (MARQUES, 1993).....	40
--	----

Figura 3.2 - Deslocamentos elásticos da pá (MARQUES, 1993).	41
Figura 3.3 - Coordenadas da seção transversal da pá antes (a) e depois de deformada (b) (MARQUES, 1993).	39
Figura 3.4 - Pá discretizada mostrando um elemento finito (MARQUES, 1993).	48
Figura 3.5 - Diagrama de blocos de um sistema bilinear (MOHLER, 1991).	54

CAPÍTULO 4

Figura 4.1 - Esquema do diagrama de blocos utilizado pelo <i>simulink</i>	62
Figura 4.2 – Gráfico mostrando o valor de $\delta(k)=1,0215 (\neq \gamma)$ para os pontos da saída de torção.....	64
Figura 4.3 - Gráfico mostrando o valor de $\delta(k)$ constante e igual a 2 para a entrada.	64
Figura 4.4 - Gráfico mostrando o valor de $\delta(k)=0.99862 (\neq \gamma)$ para a saída de batimento na ponta da pá	65
Figura 4.5 - Gráfico mostrando o valor $\delta(k)=2.06 (\neq \gamma)$ para a saída de torção.	66
Figura 4.6 - Gráfico mostrando o valor de $\delta(k)=1.2941 (\neq \gamma)$ para a saída correspondente ao batimento na ponta da pá.	66
Figura 4.7 - Gráfico mostrando o valor de $\delta(k)$ constante e igual a 2 para a entrada.	67
Figura 4.8 – Esquema de treinamento supervisionado do modelo série-paralela	69
Figura 4.9 - Esquema da estrutura da rede neural utilizada	71
Figura 4.10 - Sinal de entrada usado na simulação.	72
Figura 4.11 - Gráfico do decaimento do erro da rede neural durante o treinamento.....	73
Figura 4.12 - Resultados da identificação para o batimento na ponta da pá.....	73
Figura 4.13 - Resultados da identificação da torção na ponta da pá.....	74
Figura 4.14 – Comparação das frequências dos sinais do batimento na ponta da pá gerados pela simulação e pela rede neural.....	74

Figura 4.15 - Comparação das frequências dos sinais de torção na ponta da pá gerados pela simulação e pela rede neural.	75
Figura 4.16 – Resultados de um segundo teste de generalização da rede neural para a torção na ponta da pá com novos resultados de simulação	76
Figura 4.17 – Resultados de um segundo teste de generalização da rede neural para o deslocamento na ponta da pá com novos resultados de simulação	76
Figura 4.18 – Resultados de um terceiro teste de generalização da rede neural para o batimento e para a torção, ambos na ponta da pá em função do tempo (segundos).	77
Figura 4.19 - Comparação dos mapas de Retorno construídos com os resultados simulados e emulados pela rede do batimento na ponta da pá.	79
Figura 4.20 - Comparação dos mapas de Retorno construídos com os resultados simulados e emulados pela rede da torção na ponta da pá.	79
Figura 4.21 - Gráfico mostrando a curva de Koch	80
Figura 4.22 - Gráfico de $\text{Log}(N(s))$ contra $\text{Log}(1/s)$ aplicando regressão linear para os resultados da simulação apresentados na Figura 4.12 para batimento na ponta da pá	81
Figura 4.23 - Gráfico de $\text{Log}(N(s))$ contra $\text{Log}(1/s)$ aplicando regressão linear para os resultados apresentados pela rede na Figura 4.12 para batimento na ponta da pá.....	82
Figura 4.24 - Gráfico de $\text{Log}(N(s))$ contra $\text{Log}(1/s)$ aplicando regressão linear para os resultados da simulação apresentados na Figura 4.13 para a torção na ponta da pá	82
Figura 4.25 – Gráfico de $\text{Log}(N(s))$ contra $\text{Log}(1/s)$ aplicando regressão linear para os resultados apresentados pela rede na Figura 4.13 para a torção na ponta da pá.....	83
Figura 4.26 – Gráfico mostrando a curva de Koch usada para validação do método usado no cálculo das dimensões das figuras.....	84
Figura 4.27 – Gráfico de $\text{Log}(N(s))$ contra $\text{Log}(1/s)$ aplicando regressão linear para a curva de Koch mostrada na Figura 4.26 para a validação do método usado.....	84

LISTA DE TABELAS

CAPÍTULO 4

Tabela 4.1 - Dados numéricos utilizados da pá do helicóptero HUDGES 300 retirados de MARQUES (1993) utilizados neste trabalho.	61
Tabela 4.2 - Comparação entre número de camadas intermediárias e soma dos erros quadráticos.	70
Tabela 4.3 - Resultados dos cálculos das dimensões fractais realizados para os sistemas emulados e simulados.	81

SÍMBOLOS

A	matriz de estado ($2n \times 2n$);
A_1	matriz de estado do sistema com carregamento aerodinâmico dependente das variáveis de estado ($2n \times 2n$);
B	matriz de entrada ($2n \times 2n$);
B_1	matriz de entrada do sistema com carregamento aerodinâmico dependente das variáveis de estado ($2n \times l$);
c	corda do perfil aerodinâmico da pá (m);
C_D	coeficiente de arrasto;
C_L	coeficiente de sustentação;
$C_{M_{CA}}$	coeficiente de momento no centro aerodinâmico;
C	matriz de saída ($p \times 2n$);
D	matriz de realimentação direta ($p \times l$);
D	força de arrasto dinâmico (N);
D_e	parcela da força de arrasto aerodinâmico que não depende das coordenadas generalizadas (N);
D_d	parcela da força de arrasto aerodinâmica linearizada que depende das coordenadas generalizadas (N);

D_e	parcela da força de arrasto aerodinâmica que depende das coordenadas generalizadas, e contém os termos bilineares (N);
e	distância entre o CG e o eixo elástico na seção transversal da pá (m);
E	módulo de elasticidade ou módulo de Yung (N/m^2);
E_c	energia cinética (Nm);
E_d	energia de deformação (Nm);
$F(q)$	função não linear das coordenadas generalizadas que compõe as parcelas L_d , D_d e M_d ;
F_c	efeito centrífugo distribuído pela pá (N/m);
g_i	número de graus de liberdade de cada nó do elemento finito de pá;
G	módulo de cisalhamento (N/m^2);
$H_1(x), \dots, H_6(x)$	funções de forma para a discretização da pá por elementos finitos;
$\vec{i}, \vec{j}, \vec{k}$	versores do sistema Oxyz;
$\vec{I}, \vec{J}, \vec{K}$	versores do sistema OXYZ;
I_y	momento de inércia em torno do eixo y (m^4);
I_z	momento de inércia em torno do eixo z (m^4);
l	número de entradas do sistema;
l_e	comprimento de cada elemento finito (m);
L	força de sustentação aerodinâmica (N);
L_c	parcela da força de sustentação aerodinâmica que não depende das coordenadas generalizadas (N);
L_d	parcela da força de sustentação aerodinâmica linearizada que depende das coordenadas generalizadas (N);
L_e	parcela da força de sustentação aerodinâmica que depende das coordenadas generalizadas e contém os termos bilineares (N);
m	massa por unidade de comprimento (Kg/m);

m_{ij}, g_{ij}, k_{ij}	coeficientes das matrizes de massa, giroscópica e de rigidez do elemento finito respectivamente;
M	momento aerodinâmico, positivo quando o bordo de ataque da pá fica para cima (Nm);
M_c	parcela do momento aerodinâmico que não depende das coordenadas generalizadas (Nm);
M_{CA}	momento aerodinâmico em torno do centro aerodinâmico (Nm);
M_d	parcela do momento aerodinâmico que depende das coordenadas generalizadas (Nm);
M_e	parcela do aerodinâmico que depende das coordenadas generalizadas, e contém os termos bilineares (N);
$M, G \text{ e } K$	matrizes de massa, giroscópica e de rigidez globais, respectivamente ($g_l(N+1) \times g_l(N+1)$);
$M_e, G_e \text{ e } K_e$	matrizes de massa, giroscópica e rigidez do elemento finito, respectivamente ($2g_l, 2g_l$);
N	número de graus de liberdade total do sistema;
N_k	matriz de termos bilineares
$Oxyz$	sistema de coordenadas fixas na pá (m);
$OXYZ$	sistema de coordenadas fixas num sistema inercial (m);
p	número de saídas do sistema;
q	vetor de coordenadas generalizadas do sistema ($n \times 1$);
Q	vetor de carregamento aerodinâmico ($n \times 1$);
Q_1	matriz de coeficientes de $L_d, D_d \text{ e } M_d$ ($n \times 2n$);
Q_2	matriz de coeficientes de $L_e, D_e \text{ e } M_e$ ($2n \times 1$);
Q_3	matriz de coeficientes de $L_e, D_e \text{ e } M_e$ ($n \times 2n$);
$\vec{r}$	vetor de posição de um ponto P arbitrário na seção transversal da pá (m);

$\vec{r}_A$	vetor de posição do centro aerodinâmico da seção transversal da pá (m);
R	comprimento total ou envergadura da pá (m);
t	tempo (s);
u, v e w	deslocamentos elásticos nas direções x, y e z , respectivamente (m);
$u(t)$	vetor de entrada ($l \times 1$);
$\vec{V}_{ar}$	vetor da velocidade do fluxo livre (m/s);
x_A, y_A e z_A	coordenadas do centro aerodinâmico da seção transversal (m);
x_l, y_l e z_l	coordenadas de um ponto f arbitrário na seção transversal (m);
$\mathbf{x}(t)$	vetor das variáveis de estado ($2n \times 1$);
$\mathbf{y}(t)$	vetor de saída ($2n \times p$);
α	ângulo de ataque de uma seção transversal arbitrária da pá (rad);
γ_{xy}, γ_{yz} e γ_{xz}	distorções em relação aos planos xy, yz e xz , respectivamente (rad);
$\gamma_{x\eta}$ e $\gamma_{y\zeta}$	distorções em relação aos planos $x\eta$ e $y\zeta$, respectivamente (rad);
$\varepsilon_x, \varepsilon_y$ e ε_z	deformações em relação aos eixos x, y e z , respectivamente (m);
η e ζ	sistema de eixos da seção transversal da pá (m);
θ_p	ângulo de passo do comando (rad);
θ_t	ângulo de torção geométrica da seção transversal da pá (rad);
Θ_0	valor central (nominal) da faixa de trabalho do ângulo de passo da pá (rad);
ρ	densidade do material da pá (kg/m^3);
ρ_{ar}	densidade do ar ao nível do mar (ISO) (Kg/m^3);
σ_x, σ_y e σ_z	tensões axiais nas direções dos eixos x, y e z , respectivamente (N/m^2);
τ_{xy}, τ_{yz} e τ_{xz}	tensões de cisalhamento nos planos xy, yz e xz , respectivamente (N/m^2);

$\tau_{x\eta}$ e $\tau_{y\zeta}$	tensões de cisalhamento nos planos $x\eta$ e $y\zeta$, respectivamente (rad);
ϕ	torção angular da seção transversal da pá deformada (rad);
$\vec{\omega}$	vetor velocidade angular da pá (rad/s);
Ω	rotação da pá (rad/s);
$(')$	d/dx (1/m);
$(.)$	d/dt (1/s);

RESUMO

SOUZA, L. F. R. (2002). *Identificação da dinâmica não linear de uma pá de helicóptero via Redes Neurais*. São Carlos, 2002. 92p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

Este trabalho apresenta uma abordagem para a identificação da dinâmica não linear do modelo matemático de uma pá de helicóptero em rotação. Durante a simulação, foi considerado o helicóptero em voo pairado. Foi usada a representação bilinear para o modelo, uma das formas mais simples de representação de um modelo não linear. O modelo matemático da pá foi implementado através do método de elementos finitos e simulado em ambiente Matlab[®]. Foi usada na identificação, uma rede neural artificial como técnica não convencional, já que demonstra grande capacidade de aproximação de modelos não lineares, grande desempenho em análise da dinâmica de sistemas flexíveis e implementação e rapidez consideráveis. As redes neurais com processamento temporal são usadas para aproximar componentes da dinâmica não linear sobre um conjunto de entradas prescritas, e são usadas em simulação como meio rápido de obter a resposta no tempo. Para capturar a natureza recursiva dos componentes dinâmicos do sistema foi usada uma rede *feedforward* com processamento temporal, com uma camada intermediária de neurônios e com entradas na rede atrasadas. Para verificar o desempenho da rede, foi feita a comparação entre os dados de simulação obtidos originalmente e os dados resultantes de simulação da rede. Este trabalho foi desenvolvido visando futuramente aplicação de técnicas de controle de vibrações em pás rotativas.

Palavras-chave: pá de helicóptero, identificação, sistemas não lineares, redes neurais.

ABSTRACT

SOUZA, L. F. R. (2002). *Identification of the dynamics non linear of a blade of helicopter through neural networks*. São Carlos, 2002. 92p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

This work presents an approach for non-linear dynamics identification of a rotating helicopter blade mathematical model. During simulation, the helicopter was considered in hovering flight. A bilinear representation was used for the model, since it is known as one of the simplest forms of representation of a non linear model. mathematical model of blade was implemented using finite elements method and simulated in Matlab. A neural network is used in the identification process as a non conventional technique, since it demonstrates good capacity for approximation non-linear models and good performance in terms of the analysis of the dynamics of flexible systems. It also presents good performance in terms of implementation and processing speed. The neural networks with time dependent processing are used to approximate the components of the non-linear dynamics over a prescribed inputs set, and they are used in simulation as a rapid way of obtaining the time response. In order to capture the recursive nature of the dynamic components of the system, a feedforward network with time dependent processing, with an intermediate layer of neurons and delayed inputs is used. The performance of the net was verified comparing the results obtained originally by simulation with those resulting from the network emulation. This work was developed in order to apply vibration control techniques to rotating blades.

Keywords: helicopter blade, identification, non linear systems, neural networks.

CAPÍTULO 1

INTRODUÇÃO

Equacionar e resolver os problemas relacionados às vibrações de helicópteros e de outras aeronaves tem sido um desafio constante para os construtores e procurando a redução dessas vibrações, uma grande quantidade de trabalhos tem sido desenvolvida. Considerável progresso já foi alcançado, porém os helicópteros modernos tendem a apresentar velocidades cada vez maiores, o que certamente representará mais vibração. Desta forma, a utilização de técnicas de redução de vibrações, particularmente através do uso da teoria de controle, é de grande interesse nessa área (MARQUES, 1993).

Para o sucesso do desenvolvimento de helicópteros, é necessário o envolvimento da maioria das disciplinas de engenharia aeronáutica, por ser uma aeronave mais complexa. Um exemplo é o desenvolvimento do rotor principal. As pás são finas e flexíveis e então até em condições de operação normal, podem sofrer deformações elásticas e torções que podem ir além dos limites teóricos considerados por hipótese para tratamento como vigas lineares. A flexibilidade das pás e os efeitos aerodinâmicos relacionados influenciam também as tensões no rotor, podendo causar fadiga e diminuir o tempo de vida da aeronave. Há também interagindo com estes fatores, o carregamento aerodinâmico adicional devido à deformação das pás que modificam substancialmente o ângulo de ataque de cada uma delas (CELLI, 1999). Devido a esses muito outros fatores, o interesse da comunidade científica em estudar controle de helicópteros cresce cada vez mais. Um importante passo, antes da aplicação de técnicas de controle, é a identificação do sistema dinâmico correspondente.

Fenômenos aeroelásticos muitas vezes apresentam não linearidades de difícil modelagem por métodos convencionais. Para contornar esta dificuldade pode-se empregar técnicas não convencionais (lógica difusa, redes neurais e algoritmos genéticos entre outros) aliadas ao uso de sistemas computacionais de razoável desempenho e rapidez.

Com a crescente disponibilidade de computadores cada vez mais rápidos e com maior capacidade de processamento, o uso de modelos matemáticos tem aumentado em praticamente todas as áreas do conhecimento humano. Devido à necessidade de modelos matemáticos que representem resultados cada vez mais próximos dos reais, tem havido grande interesse no uso de representações não lineares ao invés de representações lineares, pois permitem analisar e reproduzir fenômenos e comportamentos dinâmicos mais complexos. Mas, a obtenção de modelos não lineares é significativamente mais trabalhosa e demorada. Simulações usando métodos tradicionais podem ser caras computacionalmente e de difícil implementação. Por isso, o uso de métodos não convencionais, já citados anteriormente, que apresentam menor custo computacional, capacidade de aprendizagem e tolerância maior a erro, entre outras vantagens, estão sendo muito explorados ultimamente.

Segundo CASSINI (1999), existem diversos tipos de modelos, como por exemplo: os modelos físicos, os mentais, os gráficos e os matemáticos ou analíticos. A escolha do tipo de modelo para representar o sistema dependerá das características estáticas e/ou dinâmicas do sistema, bem como do propósito para o qual o modelo está sendo determinado. A modelagem pela física do processo é uma abordagem analítica realizada através do conhecimento de leis físicas do processo, exigindo conhecimento detalhado do mesmo, o que às vezes o torna inviável. A identificação, por outro lado é uma abordagem experimental, que consiste no desenvolvimento de técnicas para a estimação de modelos de sistemas dinâmicos através de dados experimentais, não exigindo conhecimento prévio do processo.

As Redes Neurais Artificiais (RNA's) surgem como mais uma ferramenta poderosa, permitindo a modelagem dos processos através de identificação, utilizando-se dos dados de entrada e saída dos mesmos. Também têm-se mostrado eficazes quando utilizadas para reconhecimento de padrões ou na área de controle de processos entre outras aplicações, como otimização ou previsão devido à sua capacidade de aprendizado assim como o seu processamento paralelo (CRUZ, 1998).

Além do mais, Redes Neurais artificiais têm sido exploradas para análise e controle de sistemas dinâmicos não lineares, por possuírem, além de todas as vantagens dos métodos não convencionais, velocidade de processamento razoavelmente alta comparada a outros métodos convencionais e capacidade de aprendizagem que se aproximam do desempenho humano. Para capturar a dinâmica de componentes, redes neurais com atrasos no tempo são exploradas.

1.1. Objetivos do Trabalho

Os modelos matemáticos de pás de helicópteros são utilizados, basicamente, para a análise da resposta estrutural e para o estudo de instabilidades aeroelásticas, como o “flutter”, ressonância com o ar, etc (MARQUES, 1993).

Nesse contexto, a proposta deste trabalho é apresentar uma investigação sobre identificação da dinâmica bilinear de uma pá de helicóptero em vôo pairado, a qual se encontra em rotação sujeita a movimentos de batimento, torção e estiramento axial, para poder futuramente propor uma técnica de redução e controle de suas vibrações. A identificação será através de uma rede neural *feedforward* que consiste de uma camada oculta sendo treinada usando uma variação do algoritmo *backpropagation* que utiliza o algoritmo de otimização de Levenberg-Marquardt. Os dados para treinamento são obtidos de simulações realizadas com um modelo de elementos finitos, usando-se o “software” *Simulink* do *Matlab*®.

1.2 Identificação de Sistemas não lineares e Redes Neurais Artificiais

Existem diversas maneiras de modelar um sistema, tais como a modelagem pela física do processo e a modelagem através de técnicas de identificação. A modelagem pela física do processo é uma abordagem analítica realizada através de conhecimento de leis físicas, exigindo um conhecimento detalhado do mesmo, o que torna, em alguns casos, difícil a sua aplicação (CASSINI, 1999). A identificação, por outro lado, é uma abordagem experimental, que consiste no desenvolvimento de técnicas para a estimação de modelos de sistemas dinâmicos através de dados experimentais, não exigindo conhecimento prévio do processo, embora tal conhecimento possa ser utilizado se disponível (AGUIRRE et al. 1998).

Segundo CASSINI (1999), há vantagens e desvantagens nos dois tipos de modelagem. Uma vantagem da modelagem matemática, que é realizada com base na física do processo, é que ela permite visualizar características intrínsecas ao sistema, sendo que, na maioria dos casos, todos os parâmetros têm significado físico. Uma desvantagem deste tipo de modelagem é que a determinação do modelo pode ser uma tarefa bastante difícil ou inviável a curto prazo. Em casos em que o processo a ser modelado é muito complexo, a obtenção de modelos usando apenas conhecimento físico do mesmo é praticamente impossível. Nestes casos, é mais apropriado utilizar técnicas de identificação de sistemas.

O problema geral de identificação de sistemas dinâmicos está representado abaixo, na Figura 1.1:

Figura 1.1 - Identificação de um sistema dinâmico.

A princípio, deve-se assumir uma planta em tempo discreto que tem como entradas e saídas as sequências $\{u(k)\}$ e $\{y(k)\}$ e dinâmica desconhecida (KÓVACS, 1996). Como um primeiro passo, deve-se assumir alguma classe parametrizada $P(\hat{W})$ de modelos de identificação e dentro desta classe mediante algum algoritmo apropriado, procurar identificar a parametrização que mais adequadamente represente a dinâmica da planta. A adequação desta parametrização é realizada através da minimização do erro de reconstrução da saída $e(k)$ o qual é também usado para reajustar os parâmetros do modelo de identificação. Portanto, com a identificação de um sistema dinâmico procura-se determinar um modelo de identificação $P(\hat{W})$, tal que, para algum ε e K se tenha:

$$\|y(k) - \hat{y}(k)\| = \|y(k) - P(\hat{W}; u(k))\| < \varepsilon, \text{ para } k \geq K \text{ e } \forall \{u(k)\} \in U,$$

onde ε é o erro admissível, W o conjunto de parâmetros, $\hat{W}$ o conjunto de parâmetros estimados pelo modelo de identificação e U o conjunto de entradas.

Os problemas de identificação de sistemas dinâmicos lineares de ordem conhecida e em condições perfeitas de observação são relativamente elementares, reduzindo-se em geral a problemas algébricos de determinação de parâmetros. Quando as observações são imprecisas envolvendo ruídos e excitações aleatórias, passa a ser um problema de estimação de parâmetros, em geral adequadamente resolvido por métodos estatísticos. Para dinâmicas não lineares, o problema de identificação dispõe de poucas ferramentas. Uma abordagem clássica tem sido identificar uma dinâmica linear aproximada e de ordem pré-especificada para uma faixa de operação suficientemente estreita do sistema não linear. Tal método de aproximação necessariamente envolve algum critério de erro mínimo: melhor resposta para uma entrada específica melhor resposta em média para uma classe de entradas possíveis, melhor comportamento assintótico, etc. Esta abordagem pode ser generalizada para a identificação de dinâmicas não lineares sobre classes parametrizadas arbitrárias: dinâmicas bilineares ou que envolvam não linearidades quadráticas, polinomiais, etc. A literatura sobre identificação de dinâmicas não lineares é atualmente bastante extensa. O que interessa neste trabalho é a aplicação da classe paramétrica particular das redes neurais multicamadas na identificação de dinâmicas não lineares.

A história de redes neurais tem progredido através de inovações de conceitos e desenvolvimento de implementações. Alguns dos trabalhos práticos no campo de redes neurais ocorreram no fim do século XIX e início do século XX. Do ponto de vista moderno, os estudos sobre redes neurais foram iniciados em 1943 com o trabalho de Warren McCulloch e Walter Pitts (HAYKIN, 1994). McCULLOCH & PITTS (1943), mostraram que redes de neurônios artificiais podem, em princípio, calcular qualquer função aritmética ou lógica. McCulloch e Pitts foram seguidos por Donald Hebb, o qual propôs um mecanismo de aprendizado em neurônios biológicos. Donald Hebb propôs um algoritmo de aprendizado baseado no reforço de ligações sinápticos, ou seja, variação de pesos de entrada (LIANG, 2002).

A primeira aplicação prática de redes neurais artificiais foi em 1950, com a invenção da rede *perceptron* associada à regra de aprendizado de Frank Rosenblatt (HAGAN, 1996). Rosenblatt construiu uma rede *perceptron* e demonstrou sua habilidade de reconhecimento de padrões. Os primeiros sucessos estimularam um grande número de

pesquisadores a desenvolver trabalhos com redes neurais. Posteriormente foi mostrado que a base da rede *perceptron* resolve somente uma classe limitada de problemas. Ao mesmo tempo, Bernard Widrow e Tedd Hoff introduziram um novo algoritmo de aprendizado e o usaram para treinar rede neural linear adaptativa, o qual era similar na estrutura e capacidade ao *perceptron* de Rosenblatt. A regra de aprendizado de Widrow-Hoff, conhecida como Regra Delta ainda é usada hoje.

Segundo HAGAN (1996), as redes de Rosenblatt e Widrow sofrem as mesmas limitações e, Rosenblatt e Widrow, cientes dessas limitações, propuseram uma nova rede que poderia superar essas limitações. De qualquer modo, eles não foram capazes de modificar com sucesso o algoritmo de aprendizado para treinar redes mais complexas.

Muitas pessoas, influenciadas por Minsky e Papert, acreditavam que não existiriam mais pesquisas sobre redes neurais. Isto combinou com o fato que não existiam computadores com capacidade para executarem os programas de redes neurais, fazendo com que muitos pesquisadores deixassem de fazer pesquisas nesta área. Pesquisas em redes neurais foram praticamente suspensas durante décadas.

Alguns trabalhos importantes continuaram durante os anos 70. Em 1972 Teuvo Kohonen e James Anderson independentemente desenvolveram novas redes neurais que poderiam atuar como a memória humana.

Durante os anos 80 a pesquisa em redes neurais aumentou substancialmente. Novos computadores e *workstations*, os quais aumentaram rapidamente sua capacidade, tornam-se amplamente úteis. Além disso, novas concepções importantes foram introduzidas.

Duas novas concepções foram responsáveis pelo renascimento de redes neurais. A primeira foi o uso de mecânica estatística para explicar a operação de uma certa classe de redes recorrentes, as quais podem ser usadas como uma memória associativa.

O segundo desenvolvimento chave dos anos 80 foi o algoritmo *backpropagation* (retropropagação) para treinar redes *perceptron* com várias camadas, o qual foi desenvolvido independentemente por vários pesquisadores. A publicação de maior influência deste algoritmo foi feita por David Rumelhart e James McClelland (HAGAN, 1996). Este

algoritmo foi a resposta para as críticas feitas por Minsky e Papert nos anos 60 (HAGAN et al., 1996).

Esses novos desenvolvimentos revigoraram o campo de redes neurais. Muitos artigos foram escritos, e redes neurais têm encontrado muitas aplicações (HAGAN et al., 1996).

Foi a publicação de NARENDRA e PARTHASARATHY que colocou de forma clara o problema da aplicabilidade de redes multicamada à identificação e posterior controle de dinâmicas não lineares.

NARENDRA & PARTHASARATHY (1990), disseram que esta identificação pode ser feita através de dois modelos: *série paralela* e *paralela*. Na identificação *série-paralela*, é apresentado à rede um conjunto de entradas e saídas obtidas da simulação numérica do sistema. A rede é alimentada com as entradas no tempo corrente, seus respectivos atrasos e os atrasos das saídas. É utilizada para isto uma rede *feedforward* com treinamento supervisionado para prever a saída no tempo corrente, isto é, saídas anteriores medidas do sistema são usadas como entradas correntes na rede. HAYKIN (2001) chama este modelo de Rede Alimentada Adiante Focada Atrasada no Tempo, pois como visto anteriormente, não depende apenas de valores presentes da rede, mas também de seus valores passados. Para o treinamento deste tipo de rede pode ser utilizado o algoritmo *backpropagation*. Na identificação paralela é apresentado à rede um conjunto de entradas e saídas obtidas da simulação numérica do sistema. A rede é alimentada com as entradas no tempo corrente, seus respectivos atrasos e os atrasos das saídas estimadas. É utilizada uma rede recorrente neste tipo de modelo.

As redes neurais artificiais, depois da publicação de NARENDRA e PARTHASARATHY, têm sido amplamente utilizadas na identificação de sistemas, pois demonstram grande capacidade de aproximação de modelos não lineares, grande desempenho em análise da dinâmica de sistemas flexíveis e são fáceis de implementar. Uma grande vantagem de sua aplicação é a geração de dados em tempo real, visto que os meios convencionais de simulação de dinâmica não linear não podem gerar dados em tempo real (SARAVANAN & DUYAR, 1993), além de outras desvantagens.

HUNT & ZBIKOWSKI (1992) apresentaram uma revisão em redes neurais artificiais no campo da modelagem, identificação e controle de sistemas não lineares. As idéias básicas e técnicas de redes neurais artificiais foram apresentadas com linguagem e notação simples. Aplicações de uma variedade de arquiteturas de redes neurais aplicadas à identificação e controle foram apresentadas. Também foram exploradas ligações existentes entre ciências que estudam controle e redes neurais de forma unificada e identificou áreas-chaves para futuras pesquisas a serem realizadas.

CHEN et al. (1990) apresentaram uma investigação de identificação de sistemas discretos usando redes neurais com apenas uma camada de neurônios. Novos algoritmos de estimação de parâmetros são derivados de modelos de redes neurais, baseados em formulações de previsão de erro, e a aplicação em dados tanto simulados como reais, demonstraram o sucesso de redes neurais.

SJOBERG (1994), apresentou alguns exemplos de identificação de sistemas não lineares utilizando redes neurais artificiais. Todos os sistemas identificados tinham tipos diferentes de não linearidades, desde bem conhecidas a menos conhecidas. Os exemplos mostraram que essas não linearidades podem ser muito bem modeladas por modelos não lineares baseados em redes neurais.

Em sistemas complexos de engenharia, relações empíricas são frequentemente exploradas para estimar o valor de parâmetros e propriedades de sistemas. São um domínio complexo caracterizado por um número de fatores interagindo e suas relações não são na maioria das vezes, necessariamente conhecidas. Além do mais, muitos dados associados a alguns parâmetros podem ser incompletos ou errados. O desenvolvimento destas relações empíricas é uma tarefa formidável que requer técnicas de modelagem sofisticadas como intuição humana e experiência. GOH (1995), analisa o uso de redes neurais com o algoritmo de treinamento *backpropagation* para ajudar a resolver este problema. Primeiramente fez uma revisão na metodologia de redes neurais e em seguida forneceu algumas diretrizes práticas para implementação de redes neurais com o uso do algoritmo de aprendizagem *backpropagation* e apresentou dois exemplos que demonstram o potencial destas redes em capturar não linearidades entre variáveis em sistemas complexos em engenharia. O trabalho demonstrou a viabilidade de usar redes neurais para capturar interações não lineares entre vários parâmetros nos sistemas complexos de engenharia civil. Aprendendo a partir de um

conjunto de padrões selecionados, os modelos de redes neurais buscaram produzir previsões razoavelmente precisas.

JUDITSKY et al. (1995), apresentaram diversos aspectos dos fundamentos matemáticos do problema de identificação caixa preta de sistemas não lineares. Foi mostrado que a qualidade do procedimento da identificação é resultado de uma determinada diferença entre o poder expressivo do modelo que se quer identificar (quanto maior o número dos parâmetros usados ao descrever o modelo, mais flexível será a aproximação), e o erro estocástico (proporcional ao número dos parâmetros). A consequência desta diferença é o simples fato de que uma boa técnica pode ser a base de um bom algoritmo de identificação. Foram considerados diferentes métodos de aproximação e dada atenção especial às aproximações adaptativas, em especial para as redes neurais e *wavelets*. Foram testadas as aproximações desenvolvidas como experiência em estimação de problemas. Finalmente considerou algumas implicações dos desenvolvimentos teóricos em aplicações práticas de implementação de novas versões de algoritmos. Segundo JUDITSKY et al. (1995), a modelagem caixa-preta tem como base relacionar os dados passados de entradas e saídas com dados atuais/futuros de saídas, com muito pouco conhecimento do sistema disponível.

RIVALS & PERSONNAZ (1996) mostraram que há algumas vantagens no uso de modelos não lineares em espaço de estados, os quais constituem uma classe grande de modelos dinâmicos não lineares. Foram recordados os fundamentos de modelagem caixa preta e espaço de estados e foi mostrado que as redes neurais são bem mais potentes que os esquemas entrada-saída convencionais. Foi dado um exemplo usando um atuador hidráulico de um braço de robô.

PATRIKAR & PROVENCE (1996) apresentaram um modelo de rede chamada polinomial que foi proposta para identificação e controle de sistemas dinâmicos não lineares. A rede polinomial pode identificar tanto sistemas de Volterra, sistemas bilineares ou sistemas polinomiais recursivos de grau elevado de uma maneira eficiente. A teoria de Volterra e de sistemas de bilineares foi desenvolvida razoavelmente bem e uma gama extensiva de resultados teóricos existe para estes sistemas. Simulações foram feitas para comparar as propriedades de aproximação de redes polinomiais com propriedades de aproximação de redes sigmoidais. Ambas as redes identificaram satisfatoriamente sistemas não lineares complexos. Observou-se que o número de parâmetros necessários para identificar um dado sistema não linear usando uma rede polinomial é muito menor que de

uma rede sigmoidal e conseqüentemente o tamanho de uma rede polinomial usada para modelar sistemas não lineares é também bem menor.

ZHANG et al. (1997) apresentaram uma revisão do estado da arte de aplicação de redes neurais artificiais. O objetivo do trabalho desenvolvido foi apresentar uma síntese das pesquisas feitas nessa área, direções de pesquisas futuras e introspecções na modelagem via redes neurais. A revisão foi feita detalhadamente, mas não de forma exaustiva. Foram abordados também os principais pontos de redes neurais, como origem, principais características, vantagens, etc.

GREENWOOD (1997) mostrou o desempenho das redes neurais multicamadas aplicadas a problemas de estimar comportamento de sistemas dinâmicos a longo tempo. Foi usada uma estratégia de evolução no treinamento de uma das redes e depois foram comparados os resultados com os de uma rede treinada com o *backpropagation* usual. Resultados indicaram que estratégias de evolução ajudam as redes a produzirem melhor desempenho. Foram descritas também duas técnicas: uma para se fazer análise qualitativa da identificação do sistema através dos mapas de Hénon, e uma técnica que pode ser usada para fazer análise quantitativa da reconstrução do sistema através do cálculo de dimensão de correlação.

SPARKS & MAGHAMI (1998), apresentaram um exemplo específico de aplicação de redes neurais artificiais para identificação rápida e eficiente da dinâmica e controle de sistemas flexíveis. Redes neurais *feedforward* foram usadas para aproximar a dinâmica dos componentes para uso em simulações, como meio mais rápido de análise do processo. Para capturar a natureza recursiva dos componentes, foram usadas redes recorrentes, as quais possuem atrasos nos estados e número apropriado de entradas atrasadas. Depois de devidamente treinadas, as redes usadas forneceram boas aproximações para a dinâmica não linear dos componentes. Para ilustrar o bom desempenho de redes, um modelo existente de reação de um sistema de roda de uma aeronave foi simulado, primeiro convencionalmente e depois com uma rede neural no lugar. Simulações indicaram que, pelo menos na presença de não linearidades significantes ou entradas que causam rápidas variações nas respostas, o modelo baseado em redes neurais forneceu respostas precisas, com aumento de velocidade computacional de aproximadamente 180 vezes. Os parâmetros das redes *feedforward* foram estimados usando o algoritmo *backpropagation* e com a técnica de otimização de *Levenberg-Marquardt*.

TAKAHASHI (1999) apresentou uma rede neural com múltiplas camadas, treinada com o algoritmo *backpropagation* para detectar o carregamento crítico para o *flutter* e as condições limite na estrutura. Foi considerado para o estudo um modelo de viga de Timoshenko. Exemplos numéricos foram apresentados para demonstrar a capacidade das redes neurais. Dos resultados dos exemplos numéricos, pôde-se chegar às seguintes conclusões. Primeiro: o carregamento crítico para que ocorra o *flutter* e as condições-limite da estrutura podem ser preditos com sucesso através de redes neurais multicamadas. Segundo: apenas uma camada intermediária na rede neural é suficiente para identificação deste sistema.

TISCHLER (1999) abordou o uso de métodos de identificação em sistemas no domínio de frequência para desenvolvimento e controle de vôo de sistemas de aeronaves. Foram realizadas análises de modelos com variada complexidade de respostas em frequências não paramétricas para funções de transferência e representações em espaço de estados de alta ordem. Foram apresentados resultados de testes feitos com dados de numerosos vôos e programas para o *Ames Research Center* incluindo helicóptero, avião com asas fixas, experimentos com rotor em túnel de vento, etc. O artigo mostrou a importância do papel da tecnologia de identificação de sistemas no desenvolvimento de aeronaves, através de simulações e testes, e na otimização de vôo.

ANDERSON & KAUFMANN (2000), utilizaram uma aproximação diferente onde é considerada a existência de mínimos locais como uma vantagem explorável, já que eles podem ser considerados como descrições diferentes do conteúdo de informação. O trabalho focaliza um método para combinar estas descrições diferentes, obtidas de várias redes neurais aperfeiçoadas, em uma rede neural generalizada. Demonstraram o desenvolvimento de redes neurais generalizadas usando dois conjuntos de dados da vida real. Os resultados mostraram que as redes neurais generalizadas melhoram o Erro Quadrático calculado antes em pelo menos 23%.

CELI & JONES (2000), analisaram alguns aspectos de análise do domínio de frequência baseados em identificação de sistemas de um sistema dinâmico grande: um modelo acoplado de rotor-fuselagem de um helicóptero, descrito na forma de espaço de estados. O objetivo era identificar alguns elementos das matrizes de estado e controle. As identificações foram desenvolvidas usando dados de teste de simulação obtidos de um modelo linearizado, portanto apresentou algumas limitações que devem ser lembradas ao

generalizar todos os resultados e conclusões deste estudo. Os resultados indicaram que o uso de respostas em frequência podem ajudar a automatizar algumas decisões tipicamente complicadas ao julgamento do analista.

MAGHAMI & SPARKS (2000) apresentaram um novo procedimento para desenvolvimento e treinamento de redes neurais artificiais, podendo ser usado para controle rápido, eficiente e também para análise de sistemas espaciais flexíveis. Redes neurais artificiais foram exploradas para avaliar rapidamente o efeito de mudanças bruscas. Especificamente, duas redes neurais foram desenvolvidas para aproximar as relações funcionais entre componentes de aeronaves e medidas de seu desempenho. Um algoritmo de treinamento, baseado em teorias estatísticas foi apresentado, garantindo uma boa aproximação de funções e uma convergência rápida ao objetivo da rede. Foi apresentado também um exemplo de aplicação em aeronaves demonstrando a praticabilidade da abordagem proposta.

RICHE et al. (2001) aplicaram duas técnicas baseadas em redes neurais para identificação de sistemas estruturais dinâmicos não lineares. Primeiro a massa do sistema foi identificada baseada em dados de acelerações usando redes neurais *feedforward*. Foram usados testes estatísticos para limitar o número de dados. Depois, redes neurais recorrentes foram montadas e testadas para simulações de carga, sendo conhecidas as acelerações dos componentes específicos do sistema, mostrando então, serem bem mais robustas. A robustez do método em simulação de dados reais, mostra o potencial de identificação de sistemas através de redes neurais para modelagem rápida de sistemas estruturais dinâmicos.

GIANNAKIS & SERPEDIN (2001) apresentaram uma abrangente lista de referências bibliográficas em identificação de sistemas não lineares e suas aplicações. Foram fornecidas referências sobre sistemas bilineares, processamento de sinais, séries de Volterra, modelos de Wiener, polinômios de Hermite e engenharia biomédica entre outros. Várias referências em biomedicina e tópicos em redes neurais artificiais também foram incluídos. A lista está bastante completa, incluindo um bom número de referências existentes até o ano de 2000, podendo auxiliar o leitor em sua pesquisa bibliográfica.

1.3 Sistemas Bilineares

Modelos de sistemas bilineares surgem numa variedade de problemas reais, como em vários campos da engenharia, biologia, economia, etc. É sabido que grande classe desses problemas pode ser mapeado por sistemas bilineares em espaço de estados. Em particular, a aplicação de sistemas bilineares na forma de espaço de estados, pode ser muito útil na resolução de problemas envolvendo questões naturais (MOHLER, 1991). Sistemas bilineares apresentam grandes vantagens, mas também algumas limitações, e por essa razão são de grande interesse atualmente.

Na teoria de controle automático, o conceito de sistemas bilineares foi introduzido em 1960 (ELLIOTT, 1998). Esta teoria pode ser aplicada em várias áreas da ciência e tecnologia e pode ser desenvolvida a partir das teorias de sistemas lineares invariantes no tempo e dos grupos de matrizes de Lie.

KERSCHEN & GOLINVAL (2001) compararam resultados numéricos e resultados de identificação experimental de vigas simétricas (rigidez linear) e não simétricas (rigidez não linear). A viga pôde ser considerada com um grau de liberdade, pois o procedimento de filtragem podia descartar as contribuições dos outros modos de vibrar. Para a parte com rigidez linear, ou seja, simétrica, dois modelos foram aplicados e deram resultados similares. Um modelo polinomial identificou significativa rigidez cúbica, com erro de 1,7%, e o modelo não polinomial conduziu a um erro de 1,8%. Para a parte com rigidez bilinear, ou seja, caso assimétrico, o erro e o ajuste da força permitiram concluir que o método de identificação usado é altamente confiável. Concluíram então que o método é uma importante ferramenta de identificação de sistemas lineares e bilineares.

ELLIOTT (1998) apresentou a definição de sistemas bilineares e algumas aplicações em engenharia; uma revisão da teoria dos métodos de sistemas bilineares, incluindo álgebra de Lie e suas contribuições; aplicação em problemas como estabilização, observabilidade e controlabilidade e breves comentários sobre as áreas onde sistemas bilineares dão sua maior contribuição para teoria de sistemas. Foram fornecidos também alguns exemplos e aplicações de sistemas bilineares.

MARQUES & BELO (1999) apresentaram uma metodologia para identificação de sistemas bilineares. Do modelo linear em espaço de estados, utilizou-se um algoritmo

genético para determinar a matriz que caracteriza a representação bilinear. Foi utilizada para identificação a dinâmica longitudinal não linear de uma aeronave. O modelo bilinear forneceu respostas com uma boa aproximação do comportamento não linear da aeronave. Os resultados da generalização mostraram que dado somente um conjunto de entradas-saídas, a aproximação é capaz de produzir representações bilineares que executam boas previsões de respostas de sistemas não lineares.

VERDULT & VERHAEGEN (2000) discutiram identificação de sistemas bilineares em espaço de estados com múltiplas entradas, múltiplas saídas e discretizados no tempo. Foram considerados dois problemas de identificação. No primeiro caso, a entrada do sistema era um sinal ruído branco conhecido. Foi mostrado que neste caso é possível identificar o sistema resolvendo um problema de otimização não linear. O número de parâmetros no problema de otimização foi reduzido explorando o princípio de separação de mínimos quadrados. No segundo caso a entrada não era conhecida. Isto tornou o problema de identificação mais difícil do que no caso com entradas conhecidas. Não foi encontrada uma solução geral. No entanto, para uma classe de sistemas bilineares em espaço de estados com única entrada e única saída, mostrou-se que é possível gerar um algoritmo simples de identificação.

1.4 Organização da Dissertação

No seguimento deste capítulo 1 de Introdução e Revisão Bibliográfica, o Capítulo 2 apresenta uma introdução às redes neurais artificiais multicamadas, com alguns conceitos básicos teóricos que estão envolvidos no trabalho, incluindo o modelo artificial e as principais arquiteturas de redes neurais usadas em problemas de identificação. Apresenta-se também uma base do algoritmo *backpropagation* e o algoritmo de Levenberg-Marquardt, usado no processo de treinamento de redes neste trabalho.

No Capítulo 3 são abordados os conceitos básicos relacionados a sistemas bilineares e suas aplicações. Mostra-se também o desenvolvimento do modelo matemático da pá do helicóptero HUDGES 300. A partir do carregamento aerodinâmico determina-se a matriz de acoplamento entre estados e entradas, chegando-se então à equação diferencial matricial de movimento da pá. Visando técnicas futuras de controle, transforma-se então essas equações para a representação no espaço de estados.

No capítulo 4 são apresentados os resultados obtidos a partir da simulação do modelo matemático da pá, os resultados da identificação feita pela rede neural e também uma análise das frequências dos modelos simulados e emulados. Apresenta-se também uma breve análise das não linearidades do modelo, os mapas de Retorno objetivando fazer uma análise qualitativa da identificação e o cálculo de dimensão dos sistemas simulado e identificado através do método conhecido como “contagem de caixas” para poder fazer uma análise quantitativa da identificação.

No capítulo 5 são apresentadas as principais conclusões do trabalho e sugestões para trabalhos futuros.

CAPÍTULO 2

REDES NEURAIIS MULTICAMADAS

2.1 Introdução

O uso de redes neurais artificiais tem crescido durante a última década. Sua capacidade para identificar não linearidades, tem contribuído muito em muitas áreas. Dentre as principais áreas de aplicação de redes neurais artificiais estão identificação de sistemas, controle de sistemas e aproximação de funções. Embora existam inúmeras arquiteturas de redes neurais, redes neurais multicamadas são as mais freqüentemente encontradas na literatura.

Neste capítulo serão apresentados os aspectos teóricos de redes neurais artificiais, incluindo o modelo do neurônio artificial e as principais arquiteturas de redes neurais usadas em problemas de identificação. Serão apresentados também uma base do algoritmo *backpropagation* (retropropagação) e depois o algoritmo de Levenberg-Marquardt, sendo este uma variação do método de Newton que foi designado para minimizar funções que são soma dos quadrados de outras funções não lineares.

Será utilizada uma notação padrão onde escalares serão representados por letra minúscula em *itálico*, vetores serão representados por letra minúscula em **negrito**, matrizes serão representadas por letra maiúscula em **negrito**.

2.2 Redes Neurais artificiais

Redes Neurais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neuronal de organismos inteligentes e que adquirem conhecimento através da experiência. Uma grande Rede Neural Artificial pode ter centenas ou milhares de unidades de processamento, que também são chamadas de neurônios, e devem funcionar de acordo com os elementos nos quais foram inspirados, recebendo e retransmitindo informações. Já o cérebro de um mamífero pode ter bilhões de neurônios.

Podem ser utilizadas na resolução de uma grande classe de problemas encontrados em várias áreas aplicadas: classificação, identificação, diagnóstico, análise de sinais e de imagens, otimização e controle. As Redes Neurais têm-se mostrado de fácil implementação, robustas no tratamento de dados com ruído e eficientes, especialmente nos problemas em que não se tem formulação analítica ou não se tem um conhecimento explícito acessível ou os dados "estão contaminados" por ruído ou o próprio problema modifica-se com o tempo. O emprego de uma Rede Neural depende da habilidade em adaptá-la ao problema considerado, por meio de mudanças nos pesos sinápticos (aprendizagem), de modo a aumentar sua eficiência.

O estudo das Redes Neurais é um dos ramos da Inteligência Artificial que mais se desenvolve, atraindo pessoas de diversas áreas do conhecimento.

2.3 Modelo de um neurônio

Um *neurônio* é a unidade fundamental de processamento de informação de uma rede neural (HAYKIN, 1999), e a Figura 2.1 mostra o modelo de um *neurônio* artificial. Podem ser identificados três elementos básicos no modelo:

Figura 2.1 - Modelo não linear de um neurônio.

1. Um conjunto de *sinapses* ou *conexões* de entrada, sendo cada entrada ponderada por um *peso sináptico*. Sendo assim, um sinal p_k na entrada ponderada ao neurônio j é multiplicado pelo seu peso sináptico w_{jk} . A ordem adotada para os índices se refere primeiramente ao neurônio em questão e o segundo índice ao terminal de entrada da sinapse ao qual o peso se refere.
2. Uma *junção de soma*, responsável pela soma dos sinais de entrada, ponderados pelos respectivos pesos das sinapses do neurônio.
3. Uma *função de ativação*, geralmente não linear, para limitar a amplitude da saída do neurônio. Tipicamente, a função de ativação do neurônio é limitada ao intervalo $[0, 1]$ ou $[-1, 1]$.

A saída a do neurônio é calculada como

$$a = f(wp + b), \quad (2.1)$$

onde f é uma função de transferência ou função de ativação do neurônio, p é a entrada da rede, w é o valor do peso que pondera a entrada p e b é o *bias*, ou seja, o peso relativo à entrada fixa -1 .

Uma vez escolhida a função de ativação, os valores dos pesos e do *bias* podem ser ajustados de forma a obter uma relação satisfatória entrada/saída.

2.3.1 Tipos de Função de Ativação

A função de ativação define a saída do neurônio em termos do seu nível de ativação interna.

A função de ativação utilizada no algoritmo *Backpropagation* deve ser contínua e diferenciável. Além disso, para maior eficiência computacional, é desejável que sua derivada seja facilmente calculada. Podem ser identificadas três classes principais de função de ativação (IYODA, 2000):

1. *Função Degrau*. Para este tipo de função, mostrada na Figura 2.2 tem-se:

$$a = \begin{cases} 1 & \text{se } n \geq 0 \\ 0 & \text{se } n < 0 \end{cases}$$

Figura 2.2 – Função Degrau.

2. *Função Linear por Partes*. Para a função linear por partes, mostrada na Figura 2.3, tem-se:

$$a = \begin{cases} -1, & \text{se } n \leq -b \\ n/b, & \text{se } -b < n \leq b \\ 1, & \text{se } n \geq b \end{cases}$$

Figura 2.3 - Função linear por partes

3. Função Sigmoidal. É a função mais utilizada em redes neurais multicamadas que são treinadas com o algoritmo *backpropagation*. Ela é definida como uma função monotônica crescente que apresenta propriedades assintóticas e de suavidade. Um exemplo de função sigmoidal é a chamada *função sigmoidal logística*. A função é definida por

$$a = \frac{1}{1 + e^{-\alpha n}} \quad (2.2)$$

e é mostrada na Figura 2.4 abaixo:

Figura 2.4 - Função sigmoidal logística.

O parâmetro α é o parâmetro de inclinação da função sigmoidal, portanto, quando varia, podemos obter funções sigmoidais com diferentes inclinações. Em muitas situações, é desejável termos a função sigmoidal variando entre -1 e 1 , e neste caso, uma função normalmente empregada é a *tangente hiperbólica*, dada por:

$$a = \tanh\left(\frac{n}{2}\right) = \frac{1 - e^{-n}}{1 + e^{-n}} \quad (2.3)$$

As funções de transferência que foram usadas neste trabalho são: função não linear *tangente hiperbólica* como a apresentada e a função linear, cujos gráficos estão apresentados a seguir nas Figuras 2.5 e 2.6.

Figura 2.5 - Função tangente hiperbólica.

Figura 2.6 - Função linear.

2.4 Redes Neurais Multicamadas

A rede neural artificial multicamadas é um sistema de neurônios ligados por conexões sinápticas, dividida em neurônios de entrada, que recebem estímulos do meio externo, neurônios intermediários ou ocultos e neurônios de saída, que se comunicam com o exterior. A forma mais simples deste tipo de rede neural consiste de um único neurônio na camada de saída, sendo conhecido como *perceptron*.

A forma de arranjar perceptrons em camadas é denominada Perceptron multicamadas (*Multilayer Perceptron*) e representada por MLP. O perceptron multicamadas foi concebido para resolver problemas mais complexos, os quais não poderiam ser resolvidos pelo modelo de neurônio básico. Os neurônios intermediários são de suma importância na

rede neural. Foi provado que sem eles torna-se impossível a resolução de problemas linearmente não separáveis. Em outras palavras pode-se dizer que uma rede é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede.

A maioria dos modelos de redes neurais possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. Em outras palavras, elas aprendem através de exemplos.

A rede neural passa por um processo de treinamento a partir dos casos reais conhecidos, adquirindo, a partir daí, a sistemática necessária para executar adequadamente o processo desejado dos dados fornecidos. Sendo assim, a rede neural é capaz de extrair regras básicas a partir de dados reais, diferindo da computação programada, onde é necessário um conjunto de regras rígidas pré-fixadas e algoritmos.

As redes neurais multicamadas são arquiteturas onde os neurônios são organizados em duas ou mais camadas de processamento, já que existirão pelo menos uma camada de entrada e uma camada de saída.

Normalmente, as camadas são classificadas em três grupos:

Camada de Entrada: onde os padrões são apresentados à rede;

Camadas Intermediárias ou Ocultas: onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;

Camada de Saída: onde o resultado final é concluído e apresentado.

2.4.1 - Uma camada de neurônios

Uma rede neural com uma camada simples constituída por S neurônios é mostrada na Figura 2.7.

É comum o número de entradas para uma determinada camada ser diferente do número de neurônios da camada seguinte (isto é, $R \neq S$).

Figura 2.7 - Camada de S neurônios (HAGAN et al., 1996)

A Figura 2.8 a seguir mostra a representação simplificada para uma rede neural com uma camada simples constituída por S neurônios.

Figura 2.8 - Camada de S neurônios, notação abreviada (HAGAN et al., 1996).

Aqui novamente, p é um vetor de comprimento R , W é uma matriz $S \times R$, e a e b são vetores de comprimento S .

2.4.2 - Múltiplas camadas de neurônios

Seja agora uma rede com várias camadas. Cada camada tem sua própria matriz peso W , seu próprio vetor *bias* b , um vetor de entrada n na função de ativação f e um vetor de saída a . Para distinguir entre cada uma delas usar-se-á índices sobrescritos. Assim, a matriz peso para a primeira camada é escrita como W^1 e a matriz peso para a segunda camada é escrita como W^2 . Esta notação é usada até a terceira camada da rede mostrada na Figura 2.9.

Na rede neural abaixo, as saídas da primeira e segunda camadas são respectivamente as entradas da segunda e terceira camadas, ou seja, o sinal de entrada se propaga da camada de entrada para a camada de saída mas não vice-versa. Em outras palavras, esta rede neural é estritamente do tipo *feedforward*. Assim, a segunda camada pode ser vista como uma rede com uma camada com $R = S^1$ entradas, $S = S^2$ neurônios e uma $S^2 \times S^1$ matriz peso W^2 . A entrada para a segunda camada é a^1 e a saída é a^2 .

Figura 2.9 - Rede com três camadas (HAGAN et al., 1996).

Os neurônios que recebem diretamente as entradas da rede constituem a chamada camada de entrada. Os neurônios que recebem como entradas as saídas daqueles da camada de entrada constituem a segunda camada e assim sucessivamente até a última camada, chamada de camada de saída. As camadas internas, entre as camadas de entrada e de saída, são geralmente referidas como camadas intermediárias ou escondidas (*hidden layers*). Desta forma, a rede mostrada acima possui uma camada de saída (terceira camada) e duas camadas intermediárias (primeira e segunda).

A mesma rede com 3 camadas discutida previamente também pode ser escrita usando uma notação matricial, como mostra a Figura 2.10.

Uma rede neural com duas camadas, sendo uma primeira com a função de ativação sigmóide e uma segunda com a função de ativação linear, pode ser treinada para realizar o mapeamento de qualquer função não linear. Já redes neurais com apenas uma camada realizam um mapeamento apenas para um número limitado de funções (HAGAN et al., 1996).

Figura 2.10 - Rede com três camadas, notação abreviada (HAGAN et al., 1996).

O número de entradas e o número de saídas da rede são definidos de acordo com o problema específico. Se existem quatro variáveis externas para serem usadas como entradas, então existem quatro entradas para a rede. Similarmente, se existem sete saídas para a rede, devem existir sete neurônios na camada de saída. As características desejadas para as saídas da rede podem também ajudar a selecionar a função de ativação para a camada de saída. Se uma saída deve assumir os valores 0 ou 1, então a função de ativação *sinál* deve ser usada.

Assim, a arquitetura de uma rede neural de camada simples é quase completamente determinada por problemas específicos, incluindo um número determinado de entradas e saídas e as características particulares da saída desejada.

2.5 Aprendizagem de uma rede neural

Entre as diversas propriedades importantes de uma rede neural, a propriedade mais atraente em problemas de identificação é a capacidade da rede em aprender sobre o seu meio ambiente e melhorar seu desempenho através de aprendizado. Uma rede neural aprende sobre o seu meio ambiente através de um processo iterativo de ajuste aplicado aos seus pesos e *bias*. Idealmente, a rede vem a conhecer mais sobre o seu meio ambiente depois de cada iteração do processo de aprendizagem.

O processo de aprendizagem, definido no contexto de redes neurais, é um processo pelo qual seus pesos são adaptados através de um processo contínuo de simulações no meio ambiente em que a rede está inserida. Existem muitos tipos de regras de aprendizado de redes neurais (HAYKIN, 1994). Elas dividem-se em três grandes categorias: *aprendizado supervisionado*, *aprendizado não supervisionado* e *aprendizado por reforço*. Neste trabalho discutir-se-á apenas o aprendizado supervisionado, pois foi o utilizado.

2.5.1 Aprendizado Supervisionado

Este tipo de aprendizagem é caracterizado pela presença de um ‘professor’ externo, cuja função é suprir a rede com a *resposta desejada* a um determinado estímulo. Define-se erro como a diferença entre a *resposta desejada* e a *resposta observada*. Abaixo é mostrado um esquema que representa um sistema com aprendizado supervisionado.

Em outras palavras, para uma rede neural com entradas e saídas denotadas respectivamente por p e t , onde p representa um vetor de entrada e t um vetor com as saídas desejadas, é fornecido o seguinte conjunto de treinamento:

$$\{(p_1, t_1)\}, \{(p_2, t_2)\}, \dots, \{(p_Q, t_Q)\},$$

Figura 2.11 - Diagrama de blocos que representa um sistema com aprendizado supervisionado.

2.6 - Algoritmo Backpropagation (Repropagação)

Dado um neurônio j , o sinal de erro do neurônio j é definido por:

$$e_j = d_j - a_j, \quad (2.4)$$

onde d_j é a saída desejada e a_j é a saída obtida.

Assumindo o valor instantâneo do erro quadrático:

$$\frac{1}{2} e_j^2, \quad (2.5)$$

correspondentemente, a soma dos erros quadráticos instantâneos para todos os neurônios de saída da rede, na iteração (ou época) m é definida por:

$$F(x) = \frac{1}{2} \sum_{j \in C} e_j^2(m) \quad (2.6)$$

onde C é o conjunto de todos os neurônios na camada de saída da rede.

A soma dos erros quadráticos instantâneos $F(x)$ é uma função de todos os parâmetros livres da rede (pesos e *bias*), para um dado conjunto de treinamento. $F(x)$ representa a função de custo (ou objetivo) como a medida do desempenho do conjunto de treinamento. O objetivo do processo de aprendizagem é ajustar os parâmetros livres da rede, ou seja, minimizar $F(x)$.

Dado um neurônio j , alimentado pelas ativações de todos os neurônios da camada imediatamente anterior, seu nível de ativação interno é dado por:

$$n_j(n) = \sum_{k=0}^p w_{jk}(n) p_k(n) \quad (2.7)$$

onde p indica o número total de entradas no neurônio j . Portanto, a saída do j -ésimo neurônio na n -ésima iteração é dada por:

$$a_j(n) = f(n_j(n)) \quad (2.8)$$

O algoritmo *backpropagation* permite o cálculo do incremento $\Delta w_{jk}(n)$ a ser dado ao peso $w_{jk}(n)$ na iteração $(n+1)$. Neste trabalho, serão usados dois procedimentos juntamente com o algoritmo *backpropagation*, que são o método do gradiente e o método de Levenberg-Marquardt.

Para o método da descida do gradiente, o incremento $\Delta w_{jk}(n)$ a ser dado ao peso pode ser definido como:

$$\Delta w_{jk}(n) = -\eta \frac{\partial F(n)}{\partial w_{jk}(n)} \quad (2.9)$$

onde η é uma constante que determina a taxa de aprendizagem do algoritmo *backpropagation*. Assim, tem-se:

$$\Delta w_{jk}(n) = \eta e_j(n) f'(n_j(n)) a_k(n) \quad (2.10)$$

e portanto, a atualização nos pesos é dada por:

$$w_{jk}(m+1) = w_{jk}(m) + \Delta w_{jk}(m) \quad (2.11)$$

Pode-se observar que o um fator fundamental para o cálculo do incremento é sinal do erro do j -ésimo neurônio de saída $e_j(m)$. Pode-se então identificar dois casos distintos que irão depender da localização do neurônio na rede, ou seja, se na camada de saída como na discussão anterior ou se na camada intermediária.

Caso 1: neurônio na camada de saída

Quando o neurônio está localizado na camada de saída, pode-se aplicar diretamente a equação (2.4) para calcular o erro $e_j(m)$ associado, pois sabe-se qual é a saída desejada. Determinado $e_j(m)$, utiliza-se a equação (2.10) para calcular $\Delta w_{jk}(m)$.

Caso 2: neurônio na camada intermediária

Para um neurônio na camada, não há nenhuma saída desejada especificada. Assim, o sinal de erro de um neurônio da camada intermediária, deve ser calculado em termos dos sinais de erro de todos os neurônios aos quais este está diretamente conectado. Dado um neurônio k pertencente a uma camada intermediária e na iteração $(n+1)$, tem-se:

$$\Delta w_{ki}^{(1)}(m) = \eta \sum_j e_j(m) f'(n_j^{(2)}(m)) w_{ki}^{(1)} a_i^1(m) \quad (2.12)$$

O valor para a taxa de aprendizagem η pode ser escolhido de forma a acelerar o treinamento. Quanto menor seu valor, menor será o ajuste dado aos pesos de uma iteração para outra. Entretanto, para muitos casos, se η for selecionado como sendo muito pequeno, o número de iterações para se chegar na solução pode ser muito grande. Por outro lado, o processo de treinamento pode ser acelerado para valores grandes de η . Em alguns casos porém, a rede pode tornar-se instável e uma forma de evitar esta instabilidade é selecionar valores adequados para η ($0 < \eta < 1$) e incluir um termo adicional chamado “momentum” α ($0 < \alpha < 1$). Este termo permite à rede ignorar pequenos mínimos locais na superfície do erro (SANTORO, 1999). Portanto, a equação (2.12) para um neurônio na camada intermediária, resulta em:

$$\Delta w_{ki}^{(1)}(m) = \Delta w_{ki}^{(1)} a_i^1(m-1) + \eta \sum_j e_j(m) f'(v_j^{(2)}(m)) w_{ki}^{(1)} a_i^1(m) \quad (2.13)$$

2.7 O Algoritmo de Levenberg-Marquardt

O algoritmo de Levenberg-Marquardt é uma variação do método de Newton que foi designado para minimizar funções que são soma dos quadrados de outras funções não lineares. Este método é muito importante no treinamento de redes neurais onde o índice do desempenho é dado como erro quadrático médio (HAGAN et al., 1996).

Neste trabalho optou-se por utilizar o algoritmo já implementado no *software* Matlab®.

2.7.1 Algoritmo básico

Por questões didáticas, optou-se por considerar primeiramente a forma do método de Newton, onde o índice de desempenho é dado na forma de soma dos quadrados. O método de Newton para otimizar um índice de desempenho de $F(\mathbf{x})$ é apresentado a seguir. Sendo,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k \quad (2.14)$$

onde $\mathbf{A}_k \equiv \nabla^2 F(\mathbf{x}) \big|_{\mathbf{x}=\mathbf{x}_k}$ e $\mathbf{g}_k \equiv \nabla F(\mathbf{x}) \big|_{\mathbf{x}=\mathbf{x}_k}$.

Assumido que $F(\mathbf{x})$ é do tipo:

$$F(\mathbf{x}) = \sum_{i=1}^N v_i^2(\mathbf{x}) = \mathbf{v}^T(\mathbf{x}) \mathbf{v}(\mathbf{x}), \quad (2.15)$$

então o j -ésimo elemento do gradiente poderá ser

$$[\nabla F(\mathbf{x})]_j = \frac{\partial F(\mathbf{x})}{\partial x_j} = 2 \sum_{i=1}^N v_i(\mathbf{x}) \frac{\partial v_i(\mathbf{x})}{\partial x_j}. \quad (2.16)$$

Pode-se escrever então o gradiente na forma matricial:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x}) \mathbf{v}(\mathbf{x}), \text{ onde} \quad (2.17)$$

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial v_1(\mathbf{x})}{\partial x_1} & \frac{\partial v_1(\mathbf{x})}{\partial x_2} \dots & \frac{\partial v_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial v_2(\mathbf{x})}{\partial x_1} & \frac{\partial v_2(\mathbf{x})}{\partial x_2} \dots & \frac{\partial v_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial v_N(\mathbf{x})}{\partial x_1} & \frac{\partial v_N(\mathbf{x})}{\partial x_2} \dots & \frac{\partial v_N(\mathbf{x})}{\partial x_n} \end{bmatrix}, \text{ é a matriz Jacobiana.} \quad (2.18)$$

Em seguida quer-se encontrar a matriz Hessiana. O elemento k, j da matriz Hessiana pode ser definido como:

$$[\nabla^2 F(\mathbf{x})]_{k,j} = \frac{\partial^2 F(\mathbf{x})}{\partial x_k \partial x_j} = 2 \sum_{i=1}^N \left\{ \frac{\partial v_i(\mathbf{x})}{\partial x_k} \frac{\partial v_i(\mathbf{x})}{\partial x_j} + v_i(\mathbf{x}) \frac{\partial^2 v_i(\mathbf{x})}{\partial x_k \partial x_j} \right\} \quad (2.19)$$

A matriz Hessiana pode então ser escrita da seguinte forma:

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}), \quad (2.20)$$

onde,

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^N v_i(\mathbf{x}) \nabla^2 v_i(\mathbf{x}). \quad (2.21)$$

Se $\mathbf{S}(\mathbf{x})$ for assumido que seja pequeno, pode-se aproximar a matriz Hessiana da seguinte forma:

$$\nabla^2 F(\mathbf{x}) \cong 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) \quad (2.22)$$

Se forem substituídas (2.22) e (2.17) em (2.14), será obtido o método de Gauss-Newton, com:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - [2\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1} 2\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \\ &= \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \end{aligned} \quad (2.23)$$

Uma grande vantagem do método de Gauss-Newton sobre o método padrão de Newton é que este não requer o cálculo das derivadas segundas.

Um problema com o método de Gauss-Newton é que a matriz $H = J^T J$ pode não ser inversível. Mas isto pode ser superado usando a seguinte modificação para aproximar a matriz Hessiana:

$$G = H + \mu I. \quad (2.24)$$

Para ver como esta matriz pode ser invertida, suponha-se que os autovalores e autovetores de H são $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ e $\{z_1, z_2, \dots, z_n\}$. Então

$$Gz_i = [H + \mu I]z_i = Hz_i + \mu z_i = \lambda_i z_i + \mu z_i = (\lambda_i + \mu)z_i, \quad (2.25)$$

Conseqüentemente os autovetores de G são os mesmos que os autovetores de H , e os autovalores de G são $(\lambda_i + \mu)$. G pode ser definida positiva acrescentando μ até $(\lambda_i + \mu) > 0$ para todo i , e conseqüentemente a matriz será inversível.

Isto conduz ao algoritmo Levenberg-Marquardt, um método mais potente e adequado para o treinamento de redes neurais, onde o índice de desempenho é dado na forma da soma dos quadrados dos erros (HAGAN, 1996). A regra de aprendizagem para a atualização dos pesos é dada por:

$$x_{k+1} = x_k - [J^T(x_k)J(x_k) + \mu_k I]^{-1} J^T(x_k)v(x_k), \quad (2.26)$$

ou,

$$\Delta x_k = -[J^T(x_k)J(x_k) + \mu_k I]^{-1} J^T(x_k)v(x_k) \quad (2.27)$$

onde $J(x_k)$ é a matriz Jacobiana das derivadas de cada erro com relação a cada peso e *bias*, μ_k é um escalar e $v(x_k)$ é o vetor de erros na camada de saída.

Este algoritmo de aprendizagem possui a característica muito útil de que se o valor do escalar μ_k for selecionado como sendo muito grande, a equação (2.27) se aproxima do algoritmo da descida do gradiente com taxa de aprendizagem pequena, ou seja:

$$\mathbf{x}_{k+1} \cong \mathbf{x}_k - \frac{1}{\mu_k} \mathbf{J}^T(\mathbf{x}_k) \mathbf{v}(\mathbf{x}_k) = \mathbf{x}_k - \frac{1}{2\mu_k} \nabla F(\mathbf{x}) \quad (2.28)$$

Caso contrário, para valores pequenos de μ_k , a equação (2.27) torna-se o método de Gauss-Newton, que converge mais rapidamente e é mais preciso quando perto do erro mínimo (SANTORO, 1999).

Assim, o objetivo é alterar μ_k de forma a trabalhar ou com o Gauss-Newton ou com o método da descida do gradiente. Neste processo, quando o erro aumenta, μ_k é aumentado e o termo $[\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]$ torna-se desprezível. O treinamento é então realizado de acordo com o termo $\left(\frac{1}{\mu_k} \mathbf{J}^T(\mathbf{x}_k) \mathbf{v}(\mathbf{x}_k) \right)$, e que é o método da descida do gradiente. Da outra forma, μ_k é diminuído.

O índice de desempenho de treinamento de uma rede multicamadas, como dito anteriormente, é dado na forma da média quadrática dos erros. Se cada saída desejada (objetivo) ocorre com igual probabilidade, o erro quadrático médio é proporcional à soma dos quadrados dos erros para todas as Q saídas desejadas no conjunto de treinamento:

$$F(\mathbf{x}) = \sum_{q=1}^Q (\mathbf{t}_q - \mathbf{a}_q)^T (\mathbf{t}_q - \mathbf{a}_q) \quad (2.29)$$

$$= \sum_{q=1}^Q \mathbf{e}_q^T \mathbf{e}_q = \sum_{q=1}^Q \sum_{j=1}^{S^M} (e_{j,q})^2 = \sum_{i=1}^N (v_i)^2$$

onde $e_{j,q}$ é o j -ésimo elemento do vetor erro para o q -ésimo par entrada-saída desejada, S^M é o número de neurônios da última camada, Q representa o número total de padrões e o vetor de erros é dado por:

$$\mathbf{v}^T = [v_1 \ v_2 \ \dots v_N] = [e_{1,1} \ e_{2,1} \ \dots e_{S^M,1} \ e_{1,2} \ \dots e_{S^M,Q}], \quad (2.30).$$

onde

$$N = Q \times S^M \text{ e } n = S^1(R+1) + S^2(S^1+1) + \dots + S^M(S^{M-1}+1).$$

Esta equação é equivalente ao índice de desempenho da equação (2.15), para o qual o Levenberg-Marquardt foi designado.

2.7.2 Cálculo da Jacobiana

Um passo muito importante no algoritmo de Levenberg-Marquardt é o cálculo da matriz Jacobiana. Para executar este cálculo, será usada uma variação do algoritmo *backpropagation*. No algoritmo *backpropagation* padrão, é preciso calcular as derivadas dos erros quadráticos com respeito aos pesos e *bias* da rede. Para criar a matriz Jacobiana será necessário calcular as derivadas dos erros ao invés das derivadas dos quadrados dos erros.

É conceitualmente uma questão simples modificar o algoritmo *backpropagation* para calcular os elementos da matriz Jacobiana. Embora o conceito básico seja simples, os detalhes da implementação podem apresentar alguns problemas. Antes de apresentar o procedimento para calcular a Jacobiana, olhe-se para a equação (2.18).

O vetor de parâmetros é

$$\mathbf{x}^T = [x_1 \ x_2 \ \dots x_n] = [w_{1,1}^1 \ w_{1,2}^1 \ \dots w_{S^1,R}^1 \ b_1^1 \ \dots b_{S^1}^1 \ w_{1,1}^2 \ \dots b_{S^M}^M], \quad (2.31)$$

Então, fazendo-se essas substituições na equação (2.18), a matriz Jacobiana para treinamento de uma rede neural multicamadas poderá ser escrita como:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_{1,1}^I} & \frac{\partial e_{1,1}}{\partial w_{1,2}^I} & \dots & \frac{\partial e_{1,1}}{\partial w_{S^I,R}^I} & \frac{\partial e_{1,1}}{\partial b_1^I} & \dots \\ \frac{\partial e_{2,1}}{\partial w_{1,1}^I} & \frac{\partial e_{2,1}}{\partial w_{1,2}^I} & \dots & \frac{\partial e_{2,1}}{\partial w_{S^I,R}^I} & \frac{\partial e_{2,1}}{\partial b_1^I} & \dots \\ \vdots & \vdots & & \vdots & \vdots & \dots \\ \frac{\partial e_{S^M,1}}{\partial w_{1,1}^I} & \frac{\partial e_{S^M,1}}{\partial w_{1,2}^I} & \dots & \frac{\partial e_{S^M,1}}{\partial w_{S^I,R}^I} & \frac{\partial e_{S^M,1}}{\partial b_1^I} & \dots \\ \frac{\partial e_{1,2}}{\partial w_{1,1}^I} & \frac{\partial e_{1,2}}{\partial w_{1,2}^I} & \dots & & \frac{\partial e_{1,2}}{\partial b_1^I} & \dots \\ \vdots & \vdots & & & \vdots & \dots \end{bmatrix}. \quad (2.32)$$

Os termos desta matriz Jacobiana podem ser calculados como sendo uma simples modificação do algoritmo *backpropagation*. O *backpropagation* padrão calcula os termos como:

$$\frac{\partial \hat{F}(\mathbf{x})}{\partial x_l} = \frac{\partial \mathbf{e}_q^T \mathbf{e}_q}{\partial x_l} \quad (2.33)$$

Para os elementos da matriz Jacobiana que são necessários para o algoritmo de Levenberg-Marquardt, os termos serão calculados como:

$$[\mathbf{J}]_{hl} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial x_l}. \quad (2.34)$$

Relembrando que na derivação do algoritmo *backpropagation*:

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m}, \quad (2.35)$$

o primeiro termo do lado direito foi definido como a sensibilidade:

$$s_i^m = \frac{\partial \hat{F}}{\partial n_i^m}. \quad (2.36)$$

O algoritmo *backpropagation* calcula as sensibilidades partindo da última camada em direção à primeira. Pode-se usar este conceito para calcular os termos necessários para a matriz Jacobiana se for definida a nova sensibilidade de Marquardt:

$$\tilde{s}_{i,h}^m = \frac{\partial v_h}{\partial n_{i,q}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \quad (2.37)$$

onde da equação (2.31), $h = (q-1)S^M + k$.

Agora pode-se calcular os elementos da matriz Jacobiana podem ser calculados como:

$$[J]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial w_{i,j}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = \tilde{s}_{i,h}^m \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = \tilde{s}_{i,h}^m \times a_{j,q}^{m-1}, \quad (2.38)$$

ou se x_l é um bias,

$$[J]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial w_i^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \times \frac{\partial n_{i,q}^m}{\partial b_i^m} = \tilde{s}_{i,h}^m \times \frac{\partial n_{i,q}^m}{\partial b_i^m} = \tilde{s}_{i,h}^m. \quad (2.39)$$

As sensibilidades de Marquardt podem ser calculadas através de algumas relações de recorrências como as sensibilidades padrões com uma modificação para a camada final. Para as sensibilidades para a camada final tem-se:

$$\tilde{s}_{i,h}^M = \frac{\partial v_h}{\partial n_{i,q}^M} = \frac{\partial e_{k,q}}{\partial n_{i,q}^M} = \frac{\partial (t_{k,q} - a_{k,q}^M)}{\partial n_{i,q}^M} = - \frac{\partial a_{k,q}^M}{\partial n_{i,q}^M} \quad (2.40)$$

$$= \begin{cases} -\dot{f}^M(n_{i,q}^M) & \text{para } i = k \\ 0 & \text{para } i \neq k \end{cases}$$

$$\text{onde } \dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m}.$$

Então enquanto aplica-se a entrada p_q à rede e calcula-se a correspondente saída a_q^p , o algoritmo *backpropagation* Levenberg-Marquardt é inicializado com:

$$\tilde{S}_q^M = -\dot{F}^M(n_q^M), \quad (2.41)$$

onde $\dot{F}^m(n^m)$ é definido como:

$$\dot{F}^m(n^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \dot{f}^m(n_{S^m}^m) \end{bmatrix}.$$

Cada coluna da matriz $\tilde{S}_q^M$ pode ser calculada através da rede usando

$$\begin{aligned} s^m &= \frac{\partial \hat{F}}{\partial n^m} = \left(\frac{\partial n^{m+1}}{\partial n^m} \right)^T \frac{\partial \hat{F}}{\partial n^{m+1}} = \dot{F}^m(n^m) (W^{m+1})^T \frac{\partial \hat{F}}{\partial n^{m+1}} \\ &= \dot{F}^m(n^m) (W^{m+1})^T s^{m+1} \end{aligned}$$

para produzir uma linha da matriz Jacobiana. As colunas podem ser produzidas usando:

$$\tilde{S}_q^m = \dot{F}^m(n_q^m) (W^{m+1})^T \tilde{S}_q^{m+1}. \quad (2.42)$$

O total de matrizes de sensibilidades de Marquardt para cada camada é então criado aumentando as matrizes calculadas para cada entrada:

$$\tilde{S}^m = \begin{bmatrix} \tilde{S}_1^m & \tilde{S}_2^m & \dots & \tilde{S}_Q^m \end{bmatrix}. \quad (2.43)$$

Note-se que para cada entrada apresentada à rede, serão retornados S^M vetores sensibilidade. Isto deve-se ao fato de se ter que calcular a derivada de cada erro individualmente, ao invés de calcular a soma dos quadrados dos erros. Para toda a entrada aplicada à rede, haverá S^M erros (um para cada elemento da saída da rede). Para cada erro

haverá uma linha da matriz Jacobiana. Depois das sensibilidades terem sido calculadas, a matriz Jacobiana é calculada através da equação (2.38) e (2.39).

2.7.3 Resumo do algoritmo de Levenberg-Marquardt

1. Apresentar todas as entradas à rede e calcular as correspondentes saídas e os erros $\mathbf{e}_q = \mathbf{t}_q - \mathbf{a}_q^M$. Calcular a soma dos quadrados dos erros sobre todas as entradas, $F(\mathbf{x})$ usando a equação (2.29).
2. Calcular a matriz Jacobiana, usando a equação (2.29). Calcular as sensibilidades com as relações de recorrência (2.42), depois de inicializar com (2.41). Aumentar as matrizes de sensibilidade Marquardt usando (2.43). Calcular os elementos da matriz Jacobiana com (2.38) e (2.39).
3. Resolver a equação (2.17) para obter $\Delta \mathbf{x}_k$.
4. Calcular novamente a soma dos quadrados dos erros usando $\mathbf{x}_k + \Delta \mathbf{x}_k$. Se esta nova soma dos quadrados é menor que a calculada no passo 1, então dividir μ por α , manter $\mathbf{x}_k + \Delta \mathbf{x}_k$ e voltar ao passo 1. Se a soma dos quadrados não diminuiu, então multiplicar μ por α e voltar ao passo 3.

É assumido que o algoritmo convergiu quando a norma do gradiente, da equação (2.17), for menor que algum valor pré-determinado, ou quando a soma dos quadrados dos erros se reduzir ao objetivo.

2.8 Conclusão

Neste capítulo foram apresentados os aspectos teóricos de redes neurais artificiais, incluindo o modelo do neurônio, as principais arquiteturas de redes neurais usadas em problemas de identificação, a base do algoritmo *backpropagation* e o algoritmo de Levenberg-Marquardt. Este capítulo é bastante importante para os outros capítulos do trabalho e fornece ao leitor uma base teórica em redes neurais.

CAPÍTULO 3

MODELO BILINEAR DE UMA PÁ DE HELICÓPTERO

3.1. Introdução

Os modelos matemáticos permitem aplicações mais avançadas, sendo os mais utilizados, seja na engenharia, biologia, medicina, economia ou outras áreas, pois na maioria dos casos, como já citado anteriormente, permitem visualizar características intrínsecas ao sistema. A modelagem matemática de sistemas dinâmicos é uma tarefa de vital importância para o desenvolvimento da ciência e da tecnologia. Na engenharia, os modelos matemáticos são representações dos sistemas físicos, que permitem análise e prever seus comportamentos sob diversas condições de operação (CASSINI, 1999).

Neste capítulo será apresentada a modelagem matemática de uma pá de helicóptero em vôo pairado. A pá será tratada como uma viga em rotação sujeita aos movimentos de batimento, torção e estiramento axial. Será aplicada a discretização através do método de elementos finitos. Serão apresentadas também a definição de sistemas bilineares e a aplicação ao modelo da pá.

3.2 Sistemas de coordenadas

Como a pá de helicóptero é um sistema dinâmico se movimentando no espaço, para a modelagem matemática é necessário adotar alguns sistemas de coordenadas que simplificam o equacionamento. Os sistemas de coordenadas adotados para determinação do

modelo matemático da pá são descritos a seguir. Estes sistemas estão baseados nos adotados por MARQUES (1993).

A Figura 3.1 abaixo mostra os sistemas de eixos $OXYZ$, para a pá sem deformação, cujos respectivos versores $\bar{I}, \bar{J}$ e $\bar{K}$, definem um sistema de eixos fixos com a origem O na raiz da pá, sendo que O coincide com o ponto onde o eixo elástico intercepta a seção transversal da pá. O outro sistema de eixos $Oxyz$ e seus respectivos versores $\bar{i}, \bar{j}$ e $\bar{k}$ estão fixos e movem-se com a pá e tendo a mesma origem que o sistema $OXYZ$. O eixo x se estende da raiz da pá para a ponta da pá e coincide com o eixo elástico quando não há deformação. O eixo y estende-se do bordo de fuga para o bordo de ataque ao longo da corda da seção transversal da pá na raiz. A rotação Ω , a que a pá está sujeita ocorre em torno do eixo Z , e os eixos x e y , pertencentes ao plano XY , apresentam um giro Ωt , com respeito aos eixos fixos X e Y . Nessas condições os eixos Z e z são sempre coincidentes.

Figura 3.1 - Sistemas de coordenadas principais da pá. (MARQUES, 1993).

A Figura 3.2, mostra a pá deformada e os deslocamentos elásticos u , v e w , funções do tempo e da posição x em relação aos eixos $Oxyz$, respectivamente.

Nas Figuras 3.3(a) e 3.3(b), outro sistema de coordenadas é definido. Trata-se do formado pelos eixos η e ζ que estão contidos numa seção transversal arbitrária da pá. Quando a pá não está deformada, esses eixos descrevem planos paralelos ao plano yz , como

mostra a figura 3.3(a) e o eixo η apresenta um giro θ_t com relação ao eixo y no plano yz , chamado *ângulo de torção geométrica*. Para o caso da pá deformada, o posicionamento dos eixos é mostrado na figura 3.3(b), onde se pode ver o giro de deformação ϕ sofrido pela seção transversal da pá e os deslocamentos de deformação v e w nas direções y e z respectivamente.

Figura 3.2 - Deslocamentos elásticos da pá (MARQUES, 1993).

Figura 3.3 - Coordenadas da seção transversal da pá antes (a) e depois de deformada (b) (MARQUES, 1993).

3.3 Energia de deformação e energia cinética

Serão mostradas as equações de energia de deformação e energia cinética da pá, pois serão importantes na obtenção das deformações sofridas pela pá. O equacionamento completo pode ser encontrado no trabalho de MARQUES, (1993).

3.3.1 Energia de deformação

Serão agora discutidas as expressões de energia de deformação e energia cinética utilizadas para a pá de helicóptero em estudo.

A expressão de energia de deformação é dada por:

$$E_d = \frac{1}{2} \int_{vol} (\sigma_x \epsilon_x + \sigma_y \epsilon_y + \sigma_z \epsilon_z + \tau_{xy} \gamma_{xy} + \tau_{yz} \gamma_{yz} + \tau_{xz} \gamma_{xz}) dvol, \quad (3.1)$$

a qual é a expressão que associa todos os efeitos do estado de tensão tridimensional com a energia de deformação. Mas, para o caso da pá estudada aqui, a energia de deformação é dada por:

$$E_d = \frac{1}{2} \int_0^R \int \int_A (\sigma_x \epsilon_x + \tau_{x\eta} \gamma_{x\eta} + \tau_{x\zeta} \gamma_{x\zeta}) d\eta d\zeta dx, \quad (3.2)$$

pois são desprezados os efeitos das tensões nas direções y e z, pois as deformações ϵ_y e ϵ_z correspondentes são muito pequenas comparadas a ϵ_x e da mesma forma, a distorção γ_{yz} é também muito menor quando comparada às distorções γ_{xy} e γ_{xz} e portanto também pode ser desprezada. Observe-se que os subscritos y e z foram trocados por η e ζ , afim de compatibilizar com a Figura 3.3(a).

Considerando a integração na área da seção transversal dos termos de tensão e cisalhamento, integrando-se ao longo da envergadura, aplicando a lei de Hooke e considerando as simplificações abaixo, que são as simplificações que levam (3.1) a (3.2),

$$\begin{aligned}
\sigma_x &= E\varepsilon_x \\
\sigma_y &= 0 \\
\sigma_z &= 0 \\
\tau_{x\eta} &= 0 \\
\tau_{x\zeta} &= G\gamma_{x\zeta} \\
\tau_{x\eta} &= G\gamma_{x\eta} \\
\tau_{\eta\zeta} &= G\gamma_{\eta\zeta}
\end{aligned} \tag{3.3}$$

obtém-se:

$$E_d = \frac{1}{2} \int_0^R \left[\iint_A (E\varepsilon_x^2 + G\gamma_{x\eta}^2 + G\gamma_{x\zeta}^2) d\eta d\zeta \right]. \tag{3.4}$$

Como, foram adotados os eixos particulares η e ζ para as seções transversais da pá, deverá então obter-se as deformações em termos de η , ζ e de θ_1 . Considerando-se um ponto arbitrário na seção transversal da pá e supondo pequenos deslocamentos, primeiramente são encontradas as coordenadas para o ponto no caso da pá sem deformação e logo após as coordenadas do ponto para o caso da pá já deformada.

Considerando-se uma fibra elementar da pá passando pelo ponto arbitrário adotado, calculam-se, primeiramente, o comprimento diferencial da fibra e em seguida seu comprimento inicial, bastando para isso fazer as deformações u , v , w e ϕ , iguais a zero. Encontra-se então a deformação devido à tração da fibra. As distorções na seção transversal da pá também são consideradas. Substituindo todos esses dados na equação (3.4), desenvolvendo e desprezando os termos que contêm θ_1' , pois é considerado pequeno demais e observando que:

$$\iint_A d\eta d\zeta = A$$

$$\iint_A \eta d\eta d\zeta = 0$$

$$\iint_A \zeta d\eta d\zeta = 0$$

$$\iint_A \eta \zeta d\eta d\zeta = 0$$

$$\iint_A \eta^2 d\eta d\zeta = I_z$$

$$\iint_A \zeta^2 d\eta d\zeta = I_y$$

$$\iint_A (\eta^2 + \zeta^2) d\eta d\zeta = J$$

obtém-se:

$$E_d = \frac{1}{2} \int_0^R \{ E A u'^2 + E I_z (v'' \cos \theta_t + w'' \sin \theta_t)^2 + E I_y (-v'' \sin \theta_t + w'' \cos \theta_t)^2 + G J \phi'^2 \} dx \quad (3.5)$$

Considerando-se também o efeito centrífugo distribuído ao longo da pá, já que a pá apresenta movimento de rotação, o valor desse efeito é dado por:

$$F_c = \int_x^R \Omega^2 m x dx \quad (3.6)$$

e suas componentes nas direções vertical e horizontal $F_c(v')^2$ e $F_c(w')^2$, respectivamente. Acrescentando estes termos na equação (3.5), obtém-se a expressão final da energia de deformação, que é dada por:

$$E_d = \frac{1}{2} \int_0^R \{ EAu'^2 + EI_z (v'' \cos \theta_t + w'' \sin \theta_t)^2 + EI_y (-v'' \sin \theta_t + w'' \cos \theta_t)^2 + GJ\phi'^2 + F_c v'^2 + F_c w'^2 \} dx. \quad (3.7)$$

3.3.2 Energia cinética

A energia cinética para o caso da pá assumida como uma viga em rotação, pode ser definida como:

$$E_c = \frac{1}{2} \int_0^R \{ \int \int \int_A (\rho d\eta d\zeta) \left(\frac{d\vec{r}}{dt} \cdot \frac{d\vec{r}}{dt} \right) \} dx, \quad (3.8)$$

onde $\frac{d\vec{r}}{dt}$, que é a velocidade de um ponto P arbitrário na seção transversal da pá é dado por

$$\frac{d\vec{r}}{dt} = \vec{\omega} \times \vec{r} + \dot{\vec{r}} \quad (3.9)$$

e $\dot{\vec{r}}$ é a derivada parcial com relação ao tempo do vetor posição $\vec{r}$.

A velocidade angular na direção z, será dada por

$$\vec{\omega} = \Omega \vec{k}.$$

Já as velocidades angulares nas direções x e y, serão desprezadas, pois são muito menores que a na direção z.

Como:

$$\vec{r} = x_I \vec{i} + y_I \vec{j} + z_I \vec{k}$$

então,

$$\begin{aligned}\bar{\omega} \times \bar{r} &= \begin{bmatrix} \bar{i} & \bar{j} & \bar{k} \\ 0 & 0 & \Omega \\ x_1 & y_1 & z_1 \end{bmatrix} = \begin{bmatrix} 0 & \Omega \\ y_1 & z_1 \end{bmatrix} \bar{i} + \begin{bmatrix} \Omega & 0 \\ z_1 & x_1 \end{bmatrix} \bar{j} + \begin{bmatrix} 0 & 0 \\ x_1 & y_1 \end{bmatrix} \bar{k} \\ &= -y_1 \Omega \bar{i} + x_1 \Omega \bar{j} .\end{aligned}\tag{3.10}$$

Sabe-se também que:

$$\bar{r} = \dot{x}_1 \bar{i} + \dot{y}_1 \bar{j} + \dot{z}_1 \bar{k} .\tag{3.11}$$

Substituindo em (3.2), as equações (3.3) e (3.4), e logo em seguida em (3.8), obtém-se:

$$E_c = \frac{1}{2} \int_0^R \int \int_A (\rho \, d\eta \, d\zeta) [(\dot{x}_1 - y_1 \Omega)^2 + (\dot{y}_1 + x_1 \Omega)^2 + \dot{z}_1^2] dx .\tag{3.12}$$

Em seguida, substituindo as derivadas com relação ao tempo, $\dot{x}_1$, $\dot{y}_1$ e $\dot{z}_1$, e assumindo-se que o eixo elástico não coincide com o eixo do CG, obtém-se a expressão final da energia cinética:

$$\begin{aligned}
E_c = & \frac{1}{2} \int_0^R \{ m [\Omega^2 x^2 + 2\Omega^2 xu + \Omega^2 u^2 + \dot{u}^2 - 2\Omega u \dot{v} + \\
& + \Omega^2 v^2 + 2\Omega x \dot{v} + 2\Omega u \dot{v} + \dot{v}^2 + \dot{w}^2] + \\
& + m e \cos \theta_t [-2\Omega \dot{u} + 2\Omega^2 v - 2\dot{u} \dot{v}' + 2\Omega v \dot{v}' + 2\phi \dot{w} - \\
& - 2\Omega^2 x v' - 2\Omega^2 u v' - 2\Omega \dot{v} v'] + \\
& + m (k_{m_2}^2 \cos^2 \theta_t + k_{m_1}^2 \sin^2 \theta_t) [\Omega^2 + \phi^2 + 2\Omega \dot{v}' + \dot{v}'^2 + \Omega^2 v'^2] + \\
& + m e \sin \theta_t [-2\Omega x \phi - 2\Omega \phi u + 2\Omega \phi \dot{u} - 2\Omega^2 \phi v - 2\phi \dot{v} - 2\dot{u} \dot{v}' + \\
& + 2\Omega \dot{w}' v - 2\Omega \dot{v} w' - 2\Omega^2 x w' - 2\Omega^2 u w'] + \\
& + m \cos \theta_t \sin \theta_t (k_{m_2}^2 - k_{m_1}^2) [-2\Omega^2 \phi - 2\Omega \phi \dot{v}' + 2\Omega \dot{w}' + 2\dot{v}' \dot{w}' + \\
& + 2\Omega \phi v' + 2\Omega^2 v' w'] + \\
& + m (k_{m_2}^2 \sin^2 \theta_t + k_{m_1}^2 \cos^2 \theta_t) [\Omega^2 \phi^2 + \phi^2 - 2\Omega \phi \dot{w}' + \dot{w}'^2 + \\
& + 2\Omega \phi w' + \Omega^2 w'^2] \} dx
\end{aligned} \tag{3.13}$$

3.4 Método dos elementos finitos

No caso da pá de helicóptero analisada neste trabalho, a discretização em elementos finitos ocorre em termos de elementos de viga com nós em cada uma das extremidades desses elementos, possuindo seis graus de liberdade por nó: deslocamentos nas direções x, y e z e giros nos planos xy, xz e no plano da seção transversal da pá.

Figura 3.4 – Pá discretizada mostrando um elemento finito (MARQUES, 1993).

São usadas funções de forma para transformar os deslocamentos elásticos da pá nos deslocamentos nodais, ou seja:

$$u = H_1(x) u_1(t) + H_2(x) u_2(t)$$

$$v = H_3(x) v_1(t) + H_4(x) v_1'(t) + H_5(x) v_2(t) + H_6(x) v_2'(t)$$

(3.14)

$$w = H_3(x) w_1(t) + H_4(x) w_1'(t) + H_5(x) w_2(t) + H_6(x) w_2'(t)$$

$$\phi = H_1(x) \phi_1(t) + H_2(x) \phi_2(t)$$

onde $H_1(x), H_2(x), H_3(x), H_4(x), H_5(x)$ e $H_6(x)$ são os polinômios de *Hermite* e as variáveis $u_1(t), u_2(t), v_1(t), v_1'(t), v_2(t), v_2'(t), w_1(t), w_1'(t), w_2(t), w_2'(t), \phi_1(t)$ e $\phi_2(t)$ formam o conjunto dos deslocamentos nodais que podem ser considerados como coordenadas generalizadas do sistema, ou seja,

$$\mathbf{q} = [u_1(t) u_2(t) v_1(t) v_1'(t) v_2(t) v_2'(t) w_1(t) w_1'(t) w_2(t) w_2'(t) \phi_1(t) \phi_2(t)]^T.$$

Os deslocamentos u , v , w e ϕ discretizados pelas equações são substituídos nas expressões de energia e carregamento aerodinâmico para que as equações de movimento sejam obtidas.

Resumidamente, os coeficientes m_{ij} , g_{ij} e k_{ij} para $i, j = 1, 2, \dots, 2g_l$, onde g_l é o número de graus de liberdade de cada nó do elemento finito da pá, foram obtidos através da substituição das deformações u , v , w e ϕ , em termos das coordenadas generalizadas (deslocamentos nodais), nas expressões de energia de deformação (3.7) e de energia cinética (3.13). Como tais coeficientes são não lineares com relação às coordenadas generalizadas q , foram adotados pequenos deslocamentos nas vizinhanças da posição de equilíbrio para obter-se a linearização, resultando em:

$$m_{ij} = \left. \frac{\partial^2 E_c}{\partial \dot{q}_i \partial \dot{q}_j} \right|_{q=0} \quad (3.15)$$

$$g_{ij} = \left. \frac{\partial^2 E_c}{\partial \dot{q}_j \partial q_i} \right|_{q=0} - \left. \frac{\partial^2 E_c}{\partial q_i \partial \dot{q}_j} \right|_{q=0} \quad (3.16)$$

$$k_{ij} = \left. \frac{\partial^2 E_d}{\partial q_j \partial q_i} \right|_{q=0} - \left. \frac{\partial^2 E_c}{\partial q_i \partial q_j} \right|_{q=0} \quad (3.17)$$

Considerando agora cada um dos elementos finitos separadamente e substituindo os respectivos parâmetros nas expressões de energia, os resultados das equações 3.15 a 3.17 formam as matrizes M_e , G_e e K_e correspondentes a cada elemento finito.

A montagem final das matrizes do sistema, ou matrizes globais, M , G e K é feita pela superposição de cada uma das matrizes dos elementos finitos.

Depois das matrizes M , G e K serem montadas e obedecendo às condições de contorno, o vetor de carregamento Q deve ser obtido, o que é mostrado a seguir.

3.5 Carregamento Aerodinâmico

As principais condições de voo a que os helicópteros estão sujeitos são: voo pairado, voo na vertical, voo para frente, para trás ou para os lados e o carregamento aerodinâmico para cada uma delas é diferente. É adotado neste trabalho o caso de voo pairado.

Toma-se um elemento dx da pá a uma distância x do eixo de rotação, considerando-se as deformações u , v , w e ϕ , mostradas nas Figuras 3.2 e 3.3(a) e (b). Adota-se também a velocidade do fluxo do ar paralela ao plano xy , o que implica em estar desprezando o efeito do fluxo descendente (“*inflow*”), e para efeito de cálculo aerodinâmico foi suposto que a seção transversal permanece sempre paralela ao plano yz , mesmo após deformações. Isso fará com que as forças de sustentação L e de arrasto D sejam respectivamente paralelas aos eixos z e y respectivamente.

As forças elementares de sustentação e arrasto, para um elemento dx da pá, são expressas respectivamente por:

$$\begin{aligned} dL &= \frac{1}{2} \rho_{ar} C_L \|\vec{V}_{ar}\|^2 c dx \\ dD &= \frac{1}{2} \rho_{ar} C_D \|\vec{V}_{ar}\|^2 c dx \end{aligned} \quad (3.18)$$

A velocidade do fluxo livre pode ser dada por:

$$\vec{V}_{ar} = [\dot{u} - \Omega v] \vec{i} + [\dot{v} + \Omega(x + u)] \vec{j} + \dot{w} \vec{k} \quad (3.19)$$

cujo módulo pode ser dado por:

$$\|\vec{V}_{ar}\| = \{[\dot{u} - \Omega v]^2 + [\dot{v} + \Omega(x + u)]^2 + \dot{w}^2\}^{\frac{1}{2}} \quad (3.20)$$

Sabe-se que os coeficientes C_L e C_D são funções do ângulo de ataque α . Supondo-se que a faixa de trabalho da pá está na região linear de C_L x α ,

$$C_L = a \alpha. \quad (3.21)$$

Já o coeficiente de arrasto, é aproximadamente uma função quadrática do ângulo de ataque, e como deseja-se ter $C_D \times \alpha$ linear, faz-se uma linearização dessa função, considerando-se uma pequena faixa de trabalho.

Adotando-se um ângulo de passo Θ_0 de 10 graus, que define um valor (nominal) da faixa de trabalho da pá, então:

$$\alpha = \theta_p + \Theta_0 + \theta_t + \phi \quad (3.22)$$

Supondo-se que a faixa de trabalho da pá admite uma variação de α entre 5 e 15 graus, lineariza-se a função $C_D \times \alpha$ nesse trecho de α , e obtém-se:

$$C_D = b \alpha \quad (3.23)$$

Substituindo-se (3.23), (3.21) e (3.20) em (3.18), e desenvolvendo para toda a envergadura da pá obtém-se:

$$L = \frac{1}{2} \rho_{ar} c a \int_x^R \alpha \{ [\dot{u} - \Omega v]^2 + [\dot{v} + \Omega(x+u)]^2 + \dot{w}^2 \} dx \quad (3.24)$$

e

$$D = \frac{1}{2} \rho_{ar} c b \int_x^R \alpha \{ [\dot{u} - \Omega v]^2 + [\dot{v} + \Omega(x+u)]^2 + \dot{w}^2 \} dx. \quad (3.25)$$

Finalmente, desenvolve-se a equação referente ao momento aerodinâmico na pá. O momento aerodinâmico total numa seção transversal da pá é dado por:

$$M = -M_{CA} + L e. \quad (3.26)$$

Considerando-se um elemento dx da pá, (3.26) pode ser escrita como:

$$dM = \frac{l}{2} \rho_{ar} \|\vec{V}_{ar}\|^2 c (C_{M_{ca}} - C_{L_e}) dx \quad (3.27)$$

Como se trata de um perfil simétrico, então $C_{M_{ca}} = 0$.

Substituindo-se (3.20) e (3.21) na equação (3.27), obtém-se:

$$M = \frac{l}{2} \rho_{ar} c a e \int_x^R \alpha \{ [\dot{u} - \Omega v]^2 + [\dot{v} + \Omega(x+u)]^2 + \dot{w}^2 \} dx. \quad (3.28)$$

Substituindo (3.22) em (3.24), obtém-se:

$$L = \frac{l}{2} \rho_{ar} c a \int_x^R (\theta_p + \theta_o + \theta_t + \phi) \left\{ [\dot{u} - \Omega v]^2 + [\dot{v} + \Omega(x+u)]^2 + \dot{w}^2 \right\} dx. \quad (3.29)$$

Separando as parcelas, obtém-se:

$$\begin{aligned} L = & \frac{l}{2} \rho_{ar} c a \int_x^R (\theta_p + \theta_o + \theta_t) \left\{ [\dot{u} - \Omega v]^2 + [\dot{v} + \Omega(x+u)]^2 + \dot{w}^2 \right\} dx + \\ & + \int_x^R \phi \left\{ [\dot{u} - \Omega v]^2 + [\dot{v} + \Omega(x+u)]^2 + \dot{w}^2 \right\} dx \end{aligned} \quad (3.30)$$

Para a força de arrasto e o momento aerodinâmico, os procedimentos são análogos.

Como, o que se pretende neste trabalho, é conseguir um modelo matemático bilinear para essa pá de helicóptero, será então dada agora a definição de sistemas bilineares e continuar-se-á o equacionamento para obter-se então o modelo.

3.6 Sistemas Bilineares

Sistemas bilineares (BLS) compreendem talvez a classe mais simples de sistemas não lineares (NLS). Sistemas bilineares são sistemas que são lineares em controle e lineares nos estados, mas não são lineares em controle e estados juntamente (MOHLER, 1991). Em

outras palavras, eles envolvem produtos de estados e controle. Pode ser interpretado que superposição não se aplica.

Pode-se usar como exemplo de sistema bilinear qualquer sistema linear, pois os sistemas lineares são uma classe especial de sistemas bilineares. Mas, um exemplo bem simples de sistema bilinear propriamente dito é dado por:

$$\frac{dx}{dt} = ux,$$

onde x é uma função escalar do tempo e u é função escalar de controle. Sistemas bilineares, invariantes no tempo, têm a seguinte forma:

$$\frac{dx(t)}{dt} = A x(t) + B u(t) + \sum_{k=1}^m N_k u_k(t) x(t), \quad (3.31)$$

onde os estados $x \in \mathcal{R}^n$, o controle $u \in \mathcal{R}^m$ com componentes $u_1, u_2, \dots, u_m$, e $N_k, k=1, \dots, m$, matrizes $n \times n$, isto é, $N_k \in \mathcal{R}^{n \times n}$. Algumas vezes, o último termo em (3.31) pode ser escrito também como $N(u, x)$. O diagrama de blocos correspondente está mostrado na Figura (3.4). A saída pode ser gerada a partir de $x(t)$ por $y(t) = Cx(t)$.

Em quase todas as aplicações, o controle aditivo é independente do controle multiplicativo, e as matrizes B e $N_k, k=1, \dots, m$, são assim construídas. Matematicamente, algumas vezes é conveniente assumir que B é uma matriz nula, de modo que (3.31) é um sistema bilinear homogêneo o qual teoricamente pode ser assumido meramente pela adição de estados com $x_{n+1}=1$, apropriadamente adicionando zeros extras em A e em N_k e apropriadamente incluindo B na nova N_k . Sistemas não lineares que são lineares em controle e em estados são pequenas generalizações de BLS, como definido em (3.31).

Em alguns sistemas práticos, como por exemplo, reatores nucleares e sistemas imunológicos, sistemas bilineares são acoplados junto, de modo que o sistema todo é mais altamente não linear. Tais sistemas acoplados podem ser lineares ou não lineares, dependendo do processo particular investigado. Por exemplo, a saída de um subsistema, dito potência do reator, pode tornar-se uma entrada a um outro subsistema, tal como transferência de calor. Num reator de potência nuclear, a temperatura apropriada da troca de calor

transforma-se numa variável de realimentação através da temperatura de reatividade para o processo de fusão nuclear que gera sistemas de potência. Tais sistemas são chamados, sistemas bilineares acoplados.

Figura 3.4 - Diagrama de blocos de um sistema bilinear (MOHLER, 1991).

3.6.1 Aplicação da teoria de sistemas bilineares à modelagem da pá

O procedimento adotado para a construção do modelo é descrito a seguir.

Expandindo (3.31), obtém-se:

$$\begin{aligned}
 L = \frac{I}{2} \rho_{ar} c a \int_x^R (\theta_p + \theta_o + \theta_t) & \left\{ \dot{u}^2 - 2\dot{u}\Omega v + \Omega^2 v^2 + \dot{v}^2 + 2\dot{v}\Omega(x+u) + \Omega^2(x+u)^2 + \right. \\
 & \left. + \dot{w}^2 \right\} dx + \int_x^R \phi \left\{ \dot{u}^2 - 2\dot{u}\Omega v + \Omega^2 v^2 + \dot{v}^2 + 2\dot{v}\Omega(x+u) + \Omega^2(x+u)^2 + \dot{w}^2 \right\} dx
 \end{aligned} \quad (3.32)$$

ou seja,

$$\begin{aligned}
L = & \frac{1}{2} \rho_{ar} c a \int_x^R (\theta_p + \theta_o + \theta_t) \dot{u}^2 - 2(\theta_p + \theta_o + \theta_t) \dot{u} \Omega v + (\theta_p + \theta_o + \theta_t) \Omega^2 v^2 + (\theta_p + \theta_o + \theta_t) \dot{v}^2 + \\
& + 2(\theta_p + \theta_o + \theta_t) \dot{v} \Omega x + 2(\theta_p + \theta_o + \theta_t) \dot{v} \Omega u + (\theta_p + \theta_o + \theta_t) \Omega^2 x^2 + 2(\theta_p + \theta_o + \theta_t) \Omega^2 x u + \\
& + (\theta_p + \theta_o + \theta_t) \Omega^2 u^2 + (\theta_p + \theta_o + \theta_t) \dot{w}^2 \} dx + \int_x^R \phi \dot{u}^2 - 2\phi \dot{u} \Omega v + \phi \Omega^2 v^2 + \phi \dot{v}^2 + 2\phi \dot{v} \Omega x + \\
& + 2\phi \dot{v} \Omega u + \phi \Omega^2 x^2 + 2\phi \Omega^2 x u + \phi \Omega^2 u^2 + \phi \dot{w}^2 \} dx
\end{aligned} \quad (3.33)$$

Desprezando-se os termos de acoplamento entre $(\theta_p + \theta_o + \theta_t)$ e os deslocamentos elásticos u , v , w e ϕ , porém mantendo-se os termos de acoplamento entre os estados e entradas apresentados abaixo, e aplicando a discretização dada na equação (3.18) em (3.19), obtém-se:

$$\begin{aligned}
L = & \frac{1}{2} \rho_{ar} c a \int_x^R (\theta_p + \theta_o + \theta_t) (\Omega^2 x^2) dx + \int_x^R \phi (2 \Omega x + \Omega^2 x^2) dx + \\
& + 2 \Omega \int_x^R (\theta_p + \theta_o + \theta_t) (\dot{v} x + \Omega u x) dx
\end{aligned} \quad (3.34)$$

Usando o fato de que $F(q)$ é uma função não linear das coordenadas generalizadas q , pode-se escrever (3.34) como:

$$L = \frac{1}{2} \rho_{ar} c a \left\{ \int_x^R (\theta_p + \theta_o + \theta_t) \Omega^2 x^2 dx + \int_x^R F(q) dx + 2 \Omega \int_x^R (\theta_p + \theta_o + \theta_t) (\dot{v} + \Omega u) x dx \right\}. \quad (3.35)$$

Fazendo,

$$L_c = \frac{1}{2} \rho_{ar} c a \int_x^R (\theta_p + \theta_o + \theta_t) \Omega^2 x^2 dx, \quad (3.36)$$

$$L_d = \frac{1}{2} \rho_{ar} c a \int_x^R F(q) dx, \quad (3.37)$$

$$\begin{aligned}
L_e &= \frac{1}{2} \rho_{ar} c a \int_x^R (\theta_P + \theta_0 + \theta_t) (\dot{v} + \Omega u) x dx \\
&= \frac{1}{2} \rho_{ar} c a \int_x^R (\theta_P + \theta_0 + \theta_t) \dot{v} x dx + \int_x^R (\theta_P + \theta_0 + \theta_t) \Omega u x dx,
\end{aligned} \tag{3.38}$$

então a equação (3.35) torna-se:

$$L = L_c + L_d + L_e. \tag{3.39}$$

Observe-se que a parcela L_e é a parcela que contém os acoplamentos entre entradas e estados.

Linearizando a parcela L_d na equação (3.39) para manter a coerência entre o modelo estrutural linear e o de carregamento aerodinâmico, são adotados pequenos deslocamentos nas vizinhanças da posição de equilíbrio. Portanto L_d torna-se:

$$L_d = \frac{1}{2} \rho_{ar} c a \int_x^R \left\{ \sum_{i=1}^n \frac{\partial F(q)}{\partial q_i} \bigg|_{q=0} q_i + \sum_{i=1}^n \frac{\hat{c} F(q)}{\hat{c} \dot{q}_i} \bigg|_{q=0} \dot{q}_i \right\} dx. \tag{3.40}$$

Assumindo o equilíbrio estático como o ponto de equilíbrio, então são nulos os valores de q_i em $\frac{\partial F(q)}{\partial q_i}$ e em $\frac{\hat{c} F(q)}{\hat{c} \dot{q}_i}$ para $i=1, 2, \dots, n$. Substituindo (3.40) em (3.39) obtém-se então a expressão final da força de sustentação contendo os termos de acoplamento bilineares, discretizada em termos do ângulo de passo da pá, das coordenadas generalizadas e em termos do acoplamento entre o ângulo de passo e coordenadas generalizadas.

Para a força de arrasto e o momento aerodinâmico, os procedimentos são análogos.

De posse das matrizes de massa, giroscópica, de rigidez e do vetor de carregamento aerodinâmico, chega-se à equação matricial do movimento da pá:

$$M\ddot{q} + G\dot{q} + Kq = Q(q, \dot{q}, \theta_P, q\theta_P) \tag{3.41}$$

3.7. Representação do sistema no espaço de estados

A equação (3.41) será transformada para a representação no espaço de estados, visando futuramente o controle da estrutura. Como as coordenadas generalizadas podem ser consideradas como variáveis de estado, o vetor $2n$ -dimensional será introduzido:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \quad (3.42)$$

Substituindo (3.42) em (3.41), e permutando por $\mathbf{M}^{-1}$ chega-se então à equação de estado:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{Q}, \quad (3.43)$$

onde as matrizes $\mathbf{A}$ ($2n \times 2n$) e $\mathbf{B}$ ($2n \times 2n$) são dadas por:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{G} \end{bmatrix} \text{ e } \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1} \end{bmatrix}. \quad (3.44)$$

Relacionando as entradas e as variáveis de estado do sistema às suas saídas, obtém-se a seguinte equação de saída:

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t). \quad (3.45)$$

O vetor $\mathbf{u}(t)$ (1×1), é o vetor formado pelas entradas do sistema e aparece implicitamente na equação (3.43) através de $\mathbf{Q}$. A matriz $\mathbf{D}$ ($1 \times p$) é nula para o caso estudado neste trabalho.

A equação (3.43) não está na forma mais usual, pois não aparece o vetor de entradas $\mathbf{u}(t)$, e isso se deve ao fato de que $\mathbf{Q}$ é função das coordenadas e velocidades generalizadas (variáveis de estado) e da entrada do sistema, que é o ângulo de passo da pá.

Explicitando $\mathbf{Q}$ em termos de $\mathbf{x}(t)$ e $\mathbf{u}(t)$, obtém-se:

$$\mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, \theta_p) = \mathbf{Q}_1 \mathbf{x}(t) + \mathbf{Q}_2 \mathbf{u}(t) + \mathbf{Q}_3 \mathbf{u}(t) \mathbf{x}(t). \quad (3.46)$$

A matriz $\mathbf{Q}_1$ ($n \times 2n$) é composta pelos coeficientes originários das parcelas L_d , D_d e M_d das equações de força de sustentação, força de arrasto e momento. O vetor $\mathbf{Q}_2$ ($2n \times 1$) é formado pelos coeficientes provenientes de L_c , D_c e M_c . Finalmente a matriz $\mathbf{Q}_3$ é formada pelos coeficientes provenientes de L_e , D_e e M_e .

Na montagem destas matrizes é preciso observar a direção de atuação das forças aerodinâmicas. A direção de atuação da força ou momento determinará a posição dos coeficientes em $\mathbf{Q}_1$, $\mathbf{Q}_2$ e $\mathbf{Q}_3$. A direção das forças aerodinâmicas determina que os coeficientes de $\mathbf{Q}_1$, $\mathbf{Q}_2$ e $\mathbf{Q}_3$ devem estar nas linhas correspondentes aos deslocamentos nodais v para a força de arrasto, w para a força de sustentação e ϕ para o momento. Isso se deve ao fato de que a força de arrasto atua na horizontal, a força de sustentação atua na vertical e o momento é em torno do eixo longitudinal da pá. Deve ser lembrado também que os cinco primeiros graus de liberdade devem ser retirados para atender às condições de contorno definidas para a pá e que também os deslocamentos axiais devem ser retirados na montagem de $\mathbf{Q}_1$, $\mathbf{Q}_2$ e $\mathbf{Q}_3$, pela retirada das linhas e colunas correspondentes.

Tendo $\mathbf{Q}_1$, $\mathbf{Q}_2$ e $\mathbf{Q}_3$ montadas e substituindo-se (3.46) em (3.41), obtém-se:

$$\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B}(\mathbf{Q}_1 \mathbf{x}(t) + \mathbf{Q}_2 \mathbf{u}(t) + \mathbf{Q}_3 \mathbf{u}(t) \mathbf{x}(t)).$$

Agrupando, tem-se:

$$\dot{\mathbf{x}}(t) = \mathbf{A}_1 \mathbf{x}(t) + \mathbf{B}_1 \mathbf{u}(t) + \mathbf{N} \mathbf{u}(t) \mathbf{x}(t), \quad (3.47)$$

onde:

$$\mathbf{A}_1 = \mathbf{A} + \mathbf{B} \mathbf{Q}_1$$

$$\mathbf{B}_1 = \mathbf{B} \mathbf{Q}_2$$

$$\mathbf{N} = \mathbf{B} \mathbf{Q}_3$$

e, como dito anteriormente, $\mathbf{x}$ é o vetor de estados, $\mathbf{u}$ é o vetor de entradas de controle, $\mathbf{A}$ é a matriz de estados, $\mathbf{B}$ é a matriz de controle e $\mathbf{N}$ é a matriz de acoplamentos entre estados e entradas de controle, que é a matriz de termos bilineares.

3.8 Conclusão

Neste capítulo, foram abordados os conceitos básicos relacionados a sistemas bilineares e suas aplicações, mostrando-se também o desenvolvimento do modelo matemático da pá. A partir do carregamento aerodinâmico determinou-se a matriz de acoplamento entre estados e entradas, chegando-se então à equação diferencial matricial de movimento da pá e finalmente essas equações foram transformadas para a representação no espaço de estados.

CAPÍTULO 4

SIMULAÇÃO E DISCUSSÃO DOS RESULTADOS

4.1 Introdução

Entre diversas representações para modelagem de sistemas não lineares, pode-se citar as séries de Volterra, séries de Wiener, modelos polinomiais, bilineares (MOHLER, 1991) e redes neurais artificiais (HAYKIN, 1994). Como cada algoritmo é melhor para representar um tipo de sistema de acordo com o tipo de não linearidade presente nos dados, não se pode dizer qual das abordagens é a mais adequada de imediato. Neste trabalho, para a modelagem da pá de helicóptero, foi adotada a representação bilinear para a simulação do modelo e, para a identificação foram utilizadas redes neurais artificiais.

Neste capítulo serão discutidos a simulação do modelo matemático da pá e em seguida os procedimentos e resultados da identificação utilizando redes neurais. Serão mostrados também os resultados e procedimentos de algumas investigações realizadas objetivando análise qualitativa e quantitativa da identificação. Para analisar a qualidade da identificação, serão mostrados os mapas de retorno dos sistemas simulado e emulado, será apresentada também uma análise das frequências de ambos os sistemas e para analisar quantitativamente a identificação, serão calculadas as dimensões dos sistemas através do método de contagem de caixas.

4.2 Simulação do modelo

O desenvolvimento do modelo matemático da pá, como já foi dito e mostrado, foi realizado através do método dos elementos finitos, devido à eficiência mostrada em modelagens de pás de helicópteros e comprovada aplicabilidade.

Os dados numéricos usados neste trabalho são baseados na pá metálica do helicóptero **HUDGES 300**, retirados de MARQUES (1993) e serão mostrados a seguir.

DADOS NUMÉRICOS UTILIZADOS	UNIDADES
Comprimento da pá:	$R=4,09\text{m};$
Massa da pá por unidade de comprimento:	$m = 2,3 \text{ Kg/m};$
Velocidade angular de rotação da pá:	$\Omega = 37,698 \text{ rad/s};$
Rigidez ao batimento:	$EI_y=3,2256 \times 10^3 \text{ N/m};$
Rigidez à flexão no plano de rotação:	$EI_z = 1,188 \times 10^5 \text{ N/m};$
Rigidez axial:	$EA = 5,09184 \times 10^7 \text{ N/m};$
Rigidez à torção:	$GJ=2,2869 \times 10^4 \text{ N/m};$
Raio de giração em torno do eixo y:	$Km_1=0,008;$
Raio de giração em torno do eixo z:	$Km_2=0,04;$
Corda do perfil da pá:	$c=0,179\text{m};$
Densidade do ar ao nível do mar (ISO):	$\rho_{ar}=1,225\text{Kg/m}^3;$
Distância entre o CG e o eixo elástico na seção transversal da pá:	$e=-0,010132073\text{m};$
Ângulo de torção geométrica na ponta da pá:	$\theta_t = -10 \text{ graus.}$

Tabela 4.1 – Dados numéricos utilizados da pá do helicóptero **HUDGES 300** retirados de MARQUES (1993) utilizados neste trabalho.

Resumindo, a pá foi modelada no capítulo 3 como sendo uma viga em rotação, sujeita aos movimentos de batimento, flexão no plano de rotação e torção. Foram apresentados os sistemas de coordenadas utilizados, as equações de energia de deformação e energia cinética e aplicado o método de elementos finitos, considerando-se pequenos deslocamentos nas vizinhanças do ponto de equilíbrio estático. O carregamento aerodinâmico foi também determinado e compatibilizado ao modelo de elementos finitos pela suposição de pequenos deslocamentos. Constrói-se a matriz contendo os termos bilineares, ou seja, a matriz que possui acoplamento de entradas e estados e chega-se então, à

equação diferencial matricial de movimento da pá. Visando futuramente a aplicação de técnicas modernas de controle, essas equações foram transformadas para a representação no espaço de estados.

Como sistemas não lineares não possuem solução direta, o procedimento habitual neste caso é, fazer uma simulação do sistema para obter as respostas no tempo, por integração das equações de movimento dos sistemas (BELO & SOUZA, 2001). Há vários pacotes de simulação disponíveis, como o *simulink*[®], que podem gerar as respostas para a análise no domínio do tempo que forem requeridas. Neste trabalho, foi utilizado o *simulink*[®] para simular batimento e torção na ponta da pá, o qual apresentou um bom desempenho.

No início, a pá foi discretizada utilizando 10 elementos finitos, mas a simulação apresentou problemas devido ao tempo de simulação que se mostrou ser muito grande. Como uma técnica de redução de modelo não linear seria difícil de se conseguir de imediato, o modelo foi simulado utilizando apenas 5 elementos finitos. Foram feitas comparações entre os resultados de simulações dos dois modelos, com 5 e com 10 elementos finitos, e não houve perdas consideráveis de precisão, tendo o custo computacional diminuído bastante. Foi considerada uma faixa de trabalho para a pá de aproximadamente 5 graus. A Figura 4.1 apresenta um esquema do diagrama de blocos utilizado pelo *simulink*[®].

Figura 4.1 - Esquema do diagrama de blocos utilizado pelo *simulink*[®]

4.3 Detecção de não Linearidades – Testes no Domínio do Tempo

Estes testes consistem em excitar o sistema com um sinal de entrada $u_1(k)$, obtendo na saída o sinal $y_1(k)$ correspondente. Repetir o procedimento para um sinal de entrada $u_2(k) = \gamma u_1(k)$, onde γ é uma constante, e obtendo na saída $y_2(k)$. Esse teste foi baseado no usado por CASSINI (1999).

Medir a saída y_0 ou termo constante, para uma entrada nula. Subtrair este valor de $y_1(k)$ e $y_2(k)$ e encontrar a seguinte razão:

$$\delta(k) = \frac{y_2(k) - y_0}{y_1(k) - y_0} \quad (4.1)$$

O sistema será linear se $\delta(k)$ for constante e igual a γ , caso contrário está-se na presença de efeitos não lineares.

4.3.1 Aplicação do teste ao modelo da pá.

Foi aplicado o teste no modelo da pá, utilizando a entrada $u_1(k)$ senoidal, com amplitude 0.01 rad, frequência de 5 Hertz e obtida a saída $y_1(k)$, para a torção. Em seguida, a amplitude da entrada foi multiplicada por $\gamma=2$, aplicada ao modelo e medida a saída de torção $y_2(k)$. O valor encontrado de $\delta(k)$ para os pontos desta saída foi de 1.0215, o que demonstra haver não linearidades envolvidas na torção. A Figura 4.2 abaixo mostra o gráfico de número de pontos usados na análise contra valores de $\delta(k)$ para torção.

Figura 4.2 – Gráfico mostrando o valor de $\delta(k) = 1.0215$ ($\neq \gamma$) para os pontos da saída de torção.

Obviamente aplicando o teste às entradas, o valor de $\delta(k)$ encontrado foi 2 pois não há não linearidades envolvidas. A Figura 4.3 a seguir mostra os resultados de $\delta(k)$ para os pontos da entrada amplificada sobre os pontos da entrada não amplificada.

Figura 4.3 – Gráfico mostrando o valor de $\delta(k)$ constante e igual a 2 para a entrada.

Com relação ao deslocamento na ponta, o valor encontrado para $\delta(k)$ foi 0,99862, o que indica que as não linearidades influenciaram mais esta saída do que a saída de torção. A Figura 4.4 mostra os valores de $\delta(k)$ encontrados para os pontos usados na análise do deslocamento na ponta da pá.

Figura 4.4 - Gráfico mostrando o valor de $\delta(k) = 0,99862 \neq \gamma$ para a saída de batimento na ponta da pá.

Um segundo teste foi feito também usando como entrada um sinal senoidal de amplitude 0,01 rad, e frequências de 5 Hz e 20 Hz juntamente. Também foi usado $\gamma = 2$ e o valor encontrado para $\delta(k)$ foi de 2,06 para a torção. A Figura 4.5 mostra os valores de $\delta(k)$ encontrados para os pontos usados na análise.

Para o batimento na ponta foi encontrado $\delta(k) = 1,2941$, também diferente de 2, o que mostra haver não linearidades envolvidas. A Figura 4.6 abaixo mostra o resultado encontrado.

Figura 4.5 - Gráfico mostrando o valor de $\delta(k)=2.06$ ($\neq \gamma$) para a saída de torção.

Figura 4.6 - Gráfico mostrando o valor de $\delta(k)=1.2941$ ($\neq \gamma$) para a saída correspondente ao batimento na ponta da pá.

Aplicando novamente o teste às entradas o valor de $\delta(k)$ encontrado foi 2 e o resultado está mostrado na Figura 4.7 a seguir.

Figura 4.7 - Gráfico mostrando o valor de $\delta(k)$ constante e igual a 2 para as entradas.

Com a aplicação deste teste, pôde-se verificar que os termos bilineares realmente influenciaram nas saídas do sistema, ou seja este tem comportamento não linear nesta faixa de trabalho adotada.

4.4 Rede Neural na modelagem

A motivação para o uso de redes neurais artificiais é o aumento substancial de velocidade alcançada na análise de processos e devido à habilidade de aproximar relações funcionais, especificamente relações não lineares. Aproximações de dinâmica usando redes neurais podem ser feitas através do uso apropriado de atrasos no tempo (SPARKS, 1998).

Existem diversos algoritmos para treinar uma rede neural, porém como já mencionado, um dos mais utilizados é o *backpropagation*, que segue o paradigma de aprendizagem supervisionada, no qual a rede neural deve reproduzir a saída gerada pelo supervisor, para uma determinada entrada (SOUZA et al., 2002). Neste trabalho foi utilizado o algoritmo *backpropagation* com a técnica de otimização de Levenberg-Marquardt para treinar as redes neurais. Optou-se por utilizar o pacote *Matlab*® ao invés de implementar o

algoritmo, pois este *software* possui uma caixa de ferramentas bem rica para trabalhar com redes neurais.

Depois de treinada, pode-se supor que a rede armazenou o conhecimento a ela fornecido. Entretanto, o conhecimento dentro de uma rede neural não é armazenado em uma localização particular. Depende de sua topologia e da importância dos pesos na camada de entrada.

A generalização de uma rede neural artificial é a capacidade de reproduzir respostas desejadas, condizentes às respostas simuladas da planta, para diferentes sinais de entrada que não foram utilizados no treinamento da rede (SARAVANAN & DUYAR, 1994). Ou seja, a identificação de um sistema dinâmico deverá reproduzir a dinâmica da planta, em outras palavras, emular a planta.

A rede é alimentada com as entradas no tempo corrente, seus respectivos atrasos e os atrasos das saídas, como na Figura 4.8 a seguir. É utilizada para isto uma rede *feedforward* com treinamento supervisionado para predizer a saída no tempo corrente, isto é, saídas anteriores medidas do sistema são usadas como entradas correntes na rede. Este modelo é também chamado Rede Alimentada Adiante Focada Atrasada no Tempo (HAYKIN, 1994). Na Figura 4.8 a seguir, é mostrado um esquema de treinamento supervisionado, onde $u(t)$ é a entrada, $u(t-1), \dots, u(t-m)$ são as entradas em tempos anteriores ou atrasadas, $y(t)$ a saída, $y(t-1), \dots, y(t-m)$ as saídas atrasadas e $\hat{y}(t)$ a saída desejada. O erro é calculado como sendo a diferença entre a saída desejada e a saída fornecida pela rede neural.

Figura 4.8 - Esquema de treinamento supervisionado do modelo série-paralela.

Segundo HAGAN (1996), para uma boa generalização é necessário que o número de parâmetros flexíveis (pesos e *bias*) seja menor que o número de padrões (número de exemplos que será fornecido à rede). Além disso, a entrada da planta deve ser um sinal aleatório do tipo ruído branco, para que a rede receba todas as informações da dinâmica da planta.

4.5 Modelagem da pá Utilizando Rede Neural

Foram treinadas algumas topologias de redes neurais artificiais do tipo *feedforward*. Primeiramente, foram treinadas algumas topologias de redes com quatro camadas de neurônios, uma camada de entrada, duas camadas intermediárias e uma camada na saída. Os resultados da identificação foram satisfatórios. Depois foram treinadas redes com três camadas, uma na entrada, uma camada intermediária e uma na saída. O desempenho nos dois casos foi semelhante, e levando-se em consideração que o tempo de processamento das redes com uma camada intermediária foi consideravelmente menor que o tempo das redes com quatro camadas, foram adotadas redes com apenas uma camada intermediária. E mais, segundo HAYKIN (1994) redes com apenas uma camada intermediária são suficientes para aproximar uma função contínua de qualquer grau.

Depois de adotadas redes com apenas uma camada intermediária, foram analisados diferentes números de neurônios nesta camada e diferentes números de atrasos no tempo na entrada e nas saídas. A Tabela 4.2 abaixo mostra alguns dos resultados dos testes realizados.

Escolhido o número de neurônios na camada intermediária analisou-se então o número de atrasos na entrada e nas saídas. Foram testados 3, 4, 5, 6, 7 8, 9 e 10 atrasos na entrada e nas saídas e pôde-se concluir que com 4 atrasos ou mais a rede possuía o mesmo desempenho. Cada topologia de rede foi treinada com pesos iniciais selecionados aleatoriamente e a topologia de rede adotada está descrita a seguir.

Depois de realizados os testes descritos acima, a topologia da rede neural adotada para emular a pá foi: *rede com três camadas*: uma camada de entrada, para receber os sinais de impulso com 13 neurônios; uma camada na saída com 2 neurônios e uma camada intermediária (oculta ou escondida) com 10 neurônios, pois como dito anteriormente foi a topologia que apresentou desempenho melhor, tanto na identificação, como em tempo de simulação. Foram fornecidos como entrada para a rede, uma entrada no tempo corrente t , quatro entradas atrasadas no tempo e para cada uma das duas saídas, quatro saídas atrasadas no tempo, conforme modelo série-paralela mencionado anteriormente.

Número de neurônios na camada intermediária	Soma dos Erros Quadráticos
2	5,75632 e -002
5	4,98096 e -003
7	3,3706 e -005
10	1,5394 e -006
11	3,63508 e -006
12	3,11067 e -006
13	2,9102 e -006
14	2,75763 e -006
15	2,08493 e -006
16	2,31931 e -006

Tabela 4.2 – Comparação entre número de camadas intermediárias e soma dos erros quadráticos.

A Figura 4.9 mostra o esquema da estrutura da rede utilizada.

Figura 4.9 - Esquema da estrutura da rede neural utilizada

4.6 Resultados e discussões

Um conjunto de dados entrada-saída foi obtido utilizando o modelo simulado através do *simulink*[®]. O sistema foi alimentado com entradas do tipo ruído (sinal aleatório), para que todos os modos fossem excitados.

O treinamento da rede inicialmente foi realizado com 300 janelas, ou seja, conjuntos de treinamento. Verificou-se que na generalização a rede não aprendia em vários pontos. À medida que se aumentou o número de janelas da rede, verificou-se uma melhor generalização até se chegar a 600 entradas onde se obteve uma generalização razoável. Com o aumento do número de entradas conseguiu se mostrar para a rede durante o treinamento vários pontos que ela não estava entendendo antes.

Portanto o treinamento da rede foi realizado entrando-se com um conjunto de 600 exemplos (pares entrada-saída) definidos aleatoriamente. O desempenho no treinamento atingiu a ordem de 10^{-6} em termos de erro.

A generalização foi realizada da seguinte maneira: utilizou-se conjuntos de entradas diferentes do utilizado para treinamento, e o mesmo conjunto de pesos e *bias* obtidos durante o treinamento.

A Figura 4.10 mostra o sinal de entrada usado na simulação do modelo e a Figura 4.11 mostra o decaimento do erro durante o treinamento.

As Figuras 4.12 e 4.13 mostram respectivamente a identificação feita pela rede para o batimento na ponta da pá e a identificação realizada pela rede para a torção na ponta da pá. Este foi um primeiro teste realizado, para verificar a capacidade de generalização da rede.

Figura 4.10 - Sinal de entrada usado na simulação.

Figura 4.11 – Gráfico do decaimento do erro da rede neural durante o treinamento.

Como pode-se observar na Figura 4.11 acima, durante o treinamento o erro estabilizava depois de 50 épocas, não sendo então vantajoso deixar a rede treinar por um número maior de épocas. O desempenho atingido foi na ordem de 10^{-6} em termos de erro.

Figura 4.12 - Resultados da identificação para o batimento na ponta da pá.

Figura 4.13 - Resultados da identificação da torção na ponta da pá

Usando os resultados das identificações obtidos neste primeiro teste e objetivando comparar a qualidade da identificação, foi feita uma análise de frequências dos sinais do modelo simulado e do modelo emulado. Os resultados estão apresentados nas Figuras 4.14 e 4.15 a seguir. As transformadas de Fourier (FFTs) estão normalizadas.

Figura 4.14 – Comparação das frequências dos sinais de batimento na ponta da pá gerados pela simulação e pela rede neural.

Figura 4.15 – Comparação das frequências dos sinais de torção na ponta da pá gerados pela simulação e pela rede neural.

Pode-se observar que as frequências contidas na resposta simulada estão contidas nas respostas emuladas.

Para verificar a capacidade de generalização da rede neural, foram feitas novas simulações, obtendo assim novos resultados e usando o mesmo conjunto de pesos e *bias* salvos depois do treinamento, pôde-se conferir a capacidade de generalização com esse outro conjunto de resultados simulados do modelo da pá. Pôde-se então observar que a topologia de rede escolhida foi eficiente na identificação dos movimentos de torção e batimento da pá em questão.

Os resultados obtidos na identificação para um segundo teste realizado estão mostrados nas Figuras 4.16 e 4.17.

Figura 4.16 - Resultados de um segundo teste de generalização da rede neural, para a torção na ponta da pá com novos resultados de simulação.

Figura 4.17 - Resultados de um segundo teste de generalização da rede neural para o deslocamento na ponta da pá com novos resultados de simulação.

Um terceiro teste de generalização da rede foi realizado com novos resultados simulados. Os resultados da identificação do batimento e torção na ponta da pá estão mostrados na Figura 4.18 abaixo.

Figura 4.18 - Resultados de um terceiro teste de generalização da rede neural para o batimento e para a torção, ambos na ponta da pá em função do tempo (segundos)

4.7 – Uma análise qualitativa da identificação realizada

4.7.1 Mapas iterativos

Segundo GREENWOOD (1997), um mapa de uma dimensão é uma função da forma:

$$x^{(n+1)} = M[x^{(n)}]$$

onde $x^{(n)}$ é o valor de x para o tempo n . A sequência $\{x^{(0)} x^{(1)} x^{(2)} \dots\}$ é chamada de órbita de $x^{(n)}$.

$x^{(n+1)}$ pode ser definido recursivamente pela iteração do mapa para n valores de tempo se $x^{(0)}$ for conhecido, podendo ser denotado por $M^n[x^{(0)}]$, onde $M^n[x^{(0)}]$ pode ser calculado como:

$$M^n[x^{(0)}] = \underbrace{M[M(\dots\{M[x^{(0)}]\}\dots)]}_{n \text{ tempos}}.$$

Há vários tipos de órbitas, sendo uma das mais importantes a do ponto fixo. Um ponto fixo satisfaz $x_0 = M(x_0)$. Outro tipo de órbita é a *órbita periódica*. Uma órbita periódica de período p tem pontos que repetem todas as iterações p . O conjunto de pontos para os quais uma órbita arbitrária eventualmente converge é chamado de *atrator*. *Atratores* são chamados estranhos quando possuem geometria intrigante e dimensão não inteira.

4.7.2 Mapa de Retorno

Um exemplo particularmente simples e por esse motivo um dos mais utilizados de um mapa bi-dimensional, é o mapa de Retorno. Dado $x^{(n)}$ pertencente a uma série temporal qualquer, plota-se $x^{(n)}$ contra $x^{(n+1)}$. A evolução mostrará o quão rico e complicado, ou seja, o quão complexo é o comportamento do sistema no tempo mostrando a trajetória dos pontos com o passar do tempo.

Foram plotados os mapas de Retorno para as duas saídas consideradas neste trabalho. A Figura 4.19 apresenta uma comparação entre os mapas de retorno para o batimento na ponta da pá gerados com os resultados da simulação com os resultados gerados pela rede. A Figura 4.20 apresenta os mapas plotados para a saída correspondente à torção, gerados com os resultados da simulação e com os resultados gerados pela rede.

Nas duas figuras pôde-se perceber que os mapas, tanto os plotados com os resultados simulados quanto os com os resultados apresentados pela rede, são bastante semelhantes, o que aponta que a qualidade da identificação foi boa, ou seja, a rede neural identificou satisfatoriamente o sistema.

Figura 4.19 – Comparação dos mapas de Retorno construídos com os resultados simulados e emulados pela rede do batimento na ponta da pá.

Figura 4.20 – Comparação dos mapas de Retorno construídos com os resultados simulados e emulados pela rede da torção na ponta da pá.

4.8 – Uma análise quantitativa dos resultados apresentados na identificação

Matemáticos e geômetras têm caracterizado elementos geométricos como o ponto, as curvas e as superfícies em termos de sua dimensão topológica. Com base nesse conceito, o ponto tem dimensão 0, a reta tem dimensão 1, o plano tem dimensão 2 e o espaço tem dimensão 3. No entanto, seria estranho atribuir à curvas bem elaboradas, como por exemplo a de Koch, dimensão unitária com a de uma reta comum.

Para resolver esse problema, adotou-se então o conceito de dimensão fractal, que pode assumir valores fracionários em vez de apenas números inteiros. Por exemplo, a curva de Koch mostrada abaixo tem dimensão 1,262. Curvas bidimensionais podem apresentar dimensão fractal entre 1 e 2.

Figura 4.21 – Gráfico mostrando a curva de Koch.

Uma interpretação conceitual da dimensão fractal é como sendo uma medida da “complexidade” do objeto de estudo. Na prática pode-se caracterizar dimensão fractal como uma medida da superfície efetiva de contato entre o objeto e seu meio, ou seja, a dimensão de uma curva fractal é um número que caracteriza a maneira na qual a medida do comprimento entre dois pontos aumenta à medida em que a escala diminui. Portanto, quanto maior essa superfície, maior será a dimensão fractal do objeto (COSTA & BIANCHI, 2002).

Pode-se determinar teoricamente a dimensão de diversas curvas fractais, mas normalmente é necessário o uso de métodos numérico-computacionais para estimar a dimensão fractal de objetos encontrados na natureza. Como o objetivo principal deste trabalho não é discutir formas de se calcular dimensões de curvas fractais, não pretende-se aprofundar muito neste assunto.

Pretende-se então calcular a dimensão dos gráficos obtidos com os resultados da simulação e os gerados pela rede para poder analisar “quantitativamente” a identificação.

Um dos métodos numérico-computacionais mais tradicionais utilizados para estimar a dimensão fractal de objetos encontrados na natureza baseia-se no fracionamento do objeto em pequenos quadrados, tradicionalmente chamado de “contagem de caixa” (COSTA & BIANCHI, 2002).

O método envolve a sobreposição no objeto representado por uma figura de malhas com células (ou ‘caixas’) de tamanhos s diferentes. Para cada uma dessas malhas, conta-se o número de caixas ($N(s)$) que contêm parte da imagem. A dimensão fractal D é determinada pela inclinação da parte linear do gráfico de $\log(1/s)$ versus $\log(N(s))$. Ou seja,

$$D \log(1/s) = \log(N(s)) .$$

Foram calculadas as dimensões dos gráficos apresentados nas Figuras 4.12 e 4.13, que mostram os resultados dos primeiros testes de generalização da rede. Os gráficos usados para obter os resultados encontrados serão mostrados a seguir.

Figura 4.22 - Gráfico de $\text{Log}(1/s)$ contra $\text{Log}(N(s))$ aplicando regressão linear para os resultados da simulação apresentados na Figura 4.12 para batimento na ponta da pá.

Figura 4.23 - Gráfico de Log(1/s) contra Log(N(s)) aplicando regressão linear para os resultados apresentados pela rede na Figura 4.12 para o batimento na ponta da pá.

Para os resultados apresentados na Figura 4.13 o mesmo procedimento foi tomado e os resultados estão apresentados logo abaixo:

Figura 4.24 - Gráfico de Log(1/s) contra Log(N(s)) aplicando regressão linear para os resultados simulados apresentados na Figura 4.13 para a torção na ponta da pá.

Figura 4.25 - Gráfico de $\text{Log}(1/s)$ contra $\text{Log}(N(s))$ aplicando regressão linear para os resultados apresentados pela rede na Figura 4.13 para a torção na ponta da pá

Para as demais Figuras 4.16, 4.17 e 4.18, não foram calculadas as dimensões, pois os resultados estão bem próximos também.

Todos os resultados encontrados das dimensões calculadas estão mostrados na Tabela 4.3 abaixo.

RESULTADOS	
Figura	Dimensão
Batimento na ponta da pá apresentado na Figura 4.12 (Resultado apresentado pela simulação)	1,718
Batimento na ponta da pá apresentado na Figura 4.12 (Resultado apresentado pela rede neural)	1,768
Torção na ponta da pá apresentado na Figura 4.13 (Resultado apresentado pela simulação)	1,743
Torção na ponta da pá apresentado na Figura 4.13 (Resultado apresentado pela rede neural)	1,7583

Tabela 4.3 Resultados dos cálculos das dimensões fractais realizados para os sistemas emulados e simulados.

Para validação do algoritmo usado, foi calculada a dimensão da curva de Koch mostrada na Figura 4.26 a seguir, que possui dimensão conhecida. Serão mostrados o gráfico usado neste cálculo e a curva usada para validação respectivamente nas Figuras 4.27 e 4.26.

O resultado obtido foi 1,2463 e a dimensão conhecida como foi dito anteriormente é 1,262. Isso mostra que o algoritmo utilizado foi bastante eficaz.

Figura 4.26 – Gráfico mostrando a curva de Koch usada para validação do método usado no cálculo das dimensões das figuras.

Figura 4.27 - Gráfico de $\text{Log}(1/s)$ contra $\text{Log}(N(s))$ aplicando regressão linear para a curva de Koch mostrada na Figura 4.26 para validação do método usado.

Com os cálculos das dimensões das Figuras, pode-se concluir que a rede neural identificou satisfatoriamente o sistema. As diferenças nos valores das dimensões calculadas foram bem pequenas, tornando-se desprezíveis.

CAPÍTULO 5

CONSIDERAÇÕES FINAIS

5.1 Conclusões finais

Este trabalho apresentou uma aplicação específica de redes neurais artificiais para identificação rápida e eficiente de sistemas dinâmicos flexíveis não lineares. Especificamente, redes neurais *feedforward* com atrasos no tempo foram designadas para aproximar a dinâmica não linear, mais precisamente bilinear de uma pá de helicóptero.

Foi aplicado um teste para detecção de não linearidades para poder analisar a influência dos termos bilineares no comportamento do sistema. Pôde-se verificar que realmente o sistema tem comportamento não linear na faixa de trabalho adotada.

Depois de devidamente treinada, a rede neural forneceu aproximações satisfatórias para as duas saídas desejadas que foram torção e batimento na ponta da pá. Para verificar a eficiência da rede, foram feitos alguns testes de generalização. Todos se mostraram satisfatórios, mas para se ter certeza de que a rede identificou bem o sistema, foram feitas também análises qualitativa e quantitativa da identificação.

Com os resultados obtidos no primeiro teste de generalização da rede, foi feita uma análise de freqüências dos sinais do modelo simulado e do modelo emulado. Pôde-se observar que as freqüências contidas na resposta simulada estavam também contidas na resposta emulada, ou seja, a identificação apresentou boa qualidade.

Ainda com relação à análise qualitativa da identificação, foram construídos os mapas de retorno para as duas saídas do sistema emulado e simulado obtidas no primeiro teste de generalização. Todos os mapas, tanto os plotados com os resultados simulados quanto os com os resultados apresentados pela rede, ficaram bastante semelhantes, o que apontou novamente que a qualidade da identificação foi boa, ou seja, a rede neural identificou satisfatoriamente o sistema.

Com relação à análise quantitativa da emulação feita pela rede, calculou-se as dimensões dos sistemas simulados e emulados através do método de contagem de caixas. Os resultados foram muito próximos, mostrando que os sistemas emulado e simulado são equivalentes.

Concluiu-se finalmente que o modelo da rede neural aproximou-se satisfatoriamente do modelo de elementos finitos. Ou seja, o poder das redes neurais artificiais na identificação de sistemas não lineares é indiscutível, desde que devidamente treinadas.

5.2 Sugestões para trabalhos futuros

A seguir são apresentadas algumas sugestões para trabalhos futuros:

- Verificar o comportamento da rede para condições fora da faixa de operação adotada, prevendo bifurcações e talvez comportamento caótico.
- Análise de condições de estabilidade fora da faixa de trabalho.
- O modelo aerodinâmico é linear. Portanto, tentar produzir um modelo mais próximo do real para a pá do helicóptero em questão.
- Incluir não linearidade no modelo estrutural, visando também deixar o modelo mais realista.
- Com relação ao treinamento, procurar alternativas para otimiza-lo.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGUIRRE, L. A.; RODRIGUES, G.G.; JÁCOME, C. R. F. (1998). Identificação de Sistemas Não-Lineares utilizando modelos NARMAX polinomiais - Uma revisão e novos resultados. *SBA Controle & Automação*, v.9, n.2, p.90-106.
- ANDERSON, G.G.; KAUFMANN, P. (2000). Development of a generalized neural network. *Elsevier*, v.50, p. 101-105.
- BALAKRISHNAN, S. N.; WEIL, R. D. (1996). Neurocontrol: A Literature Survey. *Mathl. Comput. Modelling*, v.23, pp. 101-117.
- BELO, E. M.; SOUZA, L. F. R. (2001). Identificação do comportamento não linear de um aerofólio flexível. Campinas. (CD ROM). 22nd Iberian Latin-American Congress on Computational Methods in Engineering.
- CASSINI, C. C. S. (1999). *Estimação Recursiva de Características Estáticas Não Lineares Utilizando Modelos Polinomiais NARMAX*. Minas Gerais. Dissertação (Mestrado). Universidade Federal de Minas Gerais.
- CELI, R. (1999). Recent applications of design optimization of rotorcraft- A survey. 55nd Annual Forum at the American Helicopter Society. Montréal, Canadá, May 25-27.
- CELI, R.; JONES, C. T. (2000). Frequency Response Sensitivity Functions for Helicopter Frequency Domain System Identification. *IBM J. Res. Develop.*, v.44, n.5, p.748-758, September.
- CHEN, S.; GRANT, P. M.; BILLINGS, S.A. (1990). Non linear system identification using neural networks. *International Journal of Control*, v.51, p.1191-1215.

- COSTA, L.F. & BIANCHI, A. G. C. (2002). Dimensão fractal. *Ciência Hoje*, v.31, n. 183, p.40-47.
- CRIVELARO, K. C. O. (1999). Identificação de regimes de escoamento bifásico horizontal com auxílio de redes neurais. São Carlos Dissertação (Mestrado). Escola de Engenharia de São Carlos, Universidade de São Paulo.
- CRUZ, J.C.G. (1998). *Identificação de Uma Torre de Retificação de Águas Ácidas Usando Redes Neurais Artificiais*. Minas Gerais. Dissertação de Mestrado. Universidade Federal de Minas Gerais.
- ELLIOTT, D. L. (1998). Bilinear Systems. Institute for Systems Research University of Maryland, College Park, MD 20742, June 30.
- ENNS, R.; SI, J. (2002). Apache Helicopter Stabilization Using Neural Dynamic Programming. *Journal of Guidance, Control, and Dynamics*, v.25, n.1, p. 19-25.
- GIANNAKIS, G. B. & SERPEDIN, E. (2001). A bibliography on nonlinear system identification. *Elsevier Science*, v.81, p.533-580.
- GOH, A. T. C. (1995). Back propagation neural networks for modeling complex systems. *Elsevier*, v.19, p. 143-151.
- GREENWOOD, G. W. (1997). Training Multiple-Layer Perceptrons to Recognize Attractors. *IEEE Transactions on evolutionary computational*, v.1, n.4, p. 244-248.
- HAGAN, M. T.; DEMUTH, H. B.; BEALE, M. (1996). *Neural network design*. Boston, PWS Publishing Company.
- HAYKIN, S. (1994). *Neural network a comprehensive foundation*. New York, Macmillan College Publishing Company.
- HAYKIN, S. (2001). *Redes Neurais : Princípios e prática*. Tradução: Paulo Martins Engel. – 2ª ed.-Porto Alegre: Bookman.

- HOVAKIMYAN, N.; RYSDYK, R.; CALISE, A. J. (2001). Dynamic Neural Networks for output feedback control. *Journal of Robust and Nonlinear Control*, v.11, n.1, p. 23-39.
- HUNT, K. J., SBARBARO, D. & ZBIKOWSKI, R. (1992). Neural Networks for Control Systems – A Survey. *Automatica*, v.28, n.6, p.1083-1112.
- IYODA, E. M (2000). *Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas*. Campinas. Dissertação (Mestrado). Universidade de Campinas.
- JUDITSKY, A. et al. (1995). Nonlinear Black Box Models in System Identification: Mathematical Foundations. *Automatica*, v.31, n.12, p.1725-1750.
- JUDITSKY, G.; GOLINVAL, C. (2001). Theoretical and experimental identification of a non-linear beam. *Journal of Sound and Vibration*, v.244, n.4, p.597-613.
- KÓVACS, Z. L.(1996). *Redes Neurais Artificiais: fundamentos e aplicações*. São Paulo, edição acadêmica.
- LIANG, Z. (2002). *Redes neurais: Introdução: Histórico*. Instituto de Ciências Matemáticas e Computação. São Carlos. São Paulo/ Notas de Aula/
- MAGHAMI, P. G.; SPARKS, D. W. (2000). Design of Neural Networks for Fast Convergence and Accuracy. , *IEEE Transactions on Neural networks*, v.11, n.1, p.113-123, Jan.
- MARQUES, F.D.; BELO, E. M. (1999). Bilinear System representation of an aircraft dynamics. *DINAME 99 Applied Mechanics in the Americas*, v.8, p.1393-1396.
- MARQUES, F. D. (1993). *Controle de vibrações em uma pá de helicóptero*. São Carlos Dissertação (Mestrado). Escola de Engenharia de São Carlos, Universidade de. São Paulo.
- McCULLOCH, W. S. & PITTS, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biophysics*, v.5, p.115-133.

- MOHLER, R. R. (1991). *Nonlinear System- Applications to Bilinear Control*, v.2, Prentice Hall, New Jersey, USA.
- NARENDRA, K. S.; PARTHASARATHY (1990). Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks*, v.1, n.1.
- PATRIKAR, A.; PROVENCE, J. (1996). Nonlinear System Identification and Adaptive Control Using Polynomial Networks. *Mathl. Comput. Modelling*, v.23, n.1/2, p.159-173.
- RIVALS, I.; PERSONNAZ, L. (1996). Black Box Modelling with State-Space Neural Networks. In: *Neural Adaptive Control Technology*, World Scientific, p.237-264.
- SANTORO, F. G. (1999). *Estudo de estratégias de controle para uma estação de rastreamento a laser*. São Carlos. 219p. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.
- SARAVANAN, N.; DUYAR, A. (1994). Modeling Space Shuttle Main Engine Using Feed-Forward Neural Networks. *Journal of Guidance, Control and Dynamics*, v. 17, n.4., p.641-648.
- SCOTT, R. (2000). Active Control of Wind-Tunnel Model Aeroelastic Response Using Neural Networks. *Proceedings of SPIE*, v.3991, n.30.
- SJOBERG, J. (1994). Some examples of Identification with Neural Networks. Department of Electrical Engineering, Linköping University, Sweden.
- SOARES, P. P. S.; NADAL, J. (1999). Aplicação de uma Rede Neural *Feedforward* com Algoritmo de Levenberg-Marquardt para Classificação de Alterações do Segmento ST do Eletrocardiograma. *Proceedings of the IV Brazilian Conference on Neural Networks- IV Congresso Brasileiro de Redes Neurais*, p. 384-389
- SOUZA, L. F. R. et al., (2002). Identificação da dinâmica não linear de uma pá de helicóptero via redes neurais. João Pessoa. (CD ROM) In: *II Congresso Nacional de Engenharia Mecânica*.

-
- SPARKS, D. W.; MAGHAMI, P. G. (1998). Neural networks for rapid design and analysis. *AIAA Journal*, v.39, p. 20-23, April.
- TAKAHASHI, I. (1999). Identification for critical flutter load and boundary conditions of a beam using neural networks. *Journal of Sound and Vibration*, v.228, n.4, p.857-870.
- TISCHLER, M. B. (1999). System Identification Methods for Aircraft Flight Control Development and Validation. *NASA Technical Memorandum 110369*.
- VERDULT, V. & VERHAEGEN, M. (2000). Bilinear State Space Systems for Nonlinear Dynamical Modelling. *Theory in Biosciences*, v.119, p.1-9.
- ZHANG, G.; PATUWO, B. E. & HU, M. Y (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, v.14, p.35-62, July.