

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA
TESE DEFENDIDA POR Antonio Fernando
Abreu de Andrade E APROVADA PELA
COMISSÃO JULGADORA EM 24.05.2001.

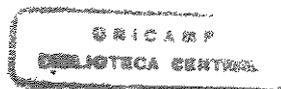
Douglas F. Zampieri
ORIENTADOR

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA**

**Controle de Suspensões Ativas Utilizando
Redes Neurais**

Autor : **Antonio Fernando Abreu de Andrade**
Orientador: **Prof. Dr. Douglas Eduardo Zampieri**

35/01



200201281

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE MECÂNICA COMPUTACIONAL**

**Controle de Suspensões Ativas Utilizando
Redes Neurais**

Autor : Antonio Fernando Abreu de Andrade

Orientador: Prof. Dr. Douglas Eduardo Zampieri

Curso: Engenharia Mecânica.

Área de Concentração: Mecânica dos Sólidos e Projeto Mecânico

Dissertação de mestrado apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para obtenção do título de Mestre em Engenharia Mecânica.

Campinas, 2001

S.P. - Brasil

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

An24c

Andrade, Antonio Fernando Abreu de
Controle de suspensões ativas utilizando redes neurais /
Antonio Fernando Abreu de Andrade. --Campinas, SP: [s.n.],
2001.

Orientador: Douglas Eduardo Zampieri.
Dissertação (mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Mecânica.

1. Redes neurais (Computação). 2. Automóveis – Molas e
suspensão. 3. Teoria do controle não-linear. 4. Automóveis -
Dinâmica. 5. Identificação de sistemas. I. Zampieri,
Douglas Eduardo. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Mecânica. III. Título.

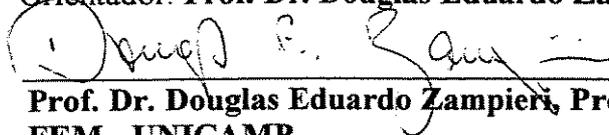
**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE MECÂNICA COMPUTACIONAL**

DISSERTAÇÃO DE MESTRADO

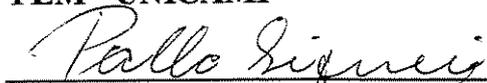
**Controle de Suspensões Ativas Utilizando
Redes Neurais**

Autor : Antonio Fernando Abreu de Andrade

Orientador: Prof. Dr. Douglas Eduardo Zampieri



**Prof. Dr. Douglas Eduardo Zampieri, Presidente
FEM - UNICAMP**



**Prof. Dr. Pablo Siqueira Meirelles
FEM - UNICAMP**



**Prof. Dr. Márcio Luiz de Andrade Netto
FEEC - UNICAMP**

Campinas, 24 de Maio de 2001

Dedicatória:

Este trabalho é dedicado aos meus pais que sempre estiveram presentes e me deram todas as condições e os ensinamentos para que eu possa realizar meus sonhos ...

Agradecimentos

Para a realização deste trabalho houve a fundamental colaboração de diversas pessoas que de forma direta ou indireta, desde o meu nascimento até os dias de hoje, me ajudaram a realizar mais um sonho, a cumprir mais uma etapa na minha vida. A todas elas presto o meu agradecimento:

Aos meus pais, Seu José e Dona Eunice, pelo amor, carinho, educação, cuidado, incentivo, dedicação e apoio em todas as etapas da minha vida, sempre confiando, sempre acreditando e principalmente, sempre sonhando comigo.

Aos meus avós, Seu José Frade e Dona Raimunda, pelo carinho, cuidado e ensinamentos.

Aos meus irmãos, Livia e Fábio, pela amizade, apoio e paciência, sobretudo nos primeiros dias aqui em Campinas, onde devo um agradecimento especial ao meu irmão Fábio, ou simplesmente, “Zé”, pela ajuda, pelos conselhos, pelos cuidados, que contribuíram de forma decisiva para a evolução e conclusão deste trabalho.

Ao meu orientador, Prof. Dr. Douglas Eduardo Zampieri, inicialmente por ter me escolhido como seu orientado, e por sua paciência, amizade, compreensão e incentivo, que contribuíram não só para a minha formação acadêmica, como também na minha formação pessoal.

Aos meus amigos e colegas, em especial à Adilson e Mildô, que sempre me apoiaram mesmo distantes, à Egnilson, Kelly, André e Sílvia, pela amizade, apoio e principalmente por me ajudar a matar um pouco a saudade da nossa terra e do nosso povo, que têm na esperança, na

perseverança, na honestidade e na amizade, a base para a realização de nossas conquistas. Aos meus amigos peruanos, em especial à Justo e a John. Devo agradecer também em especial à Alex, Daniel, Marco Antônio (o popular Marcão) e a Ubatan (todos baianos legítimos), pela amizade, apoio e toda a ajuda que me deram para a realização deste trabalho.

A todos os professores, em especial ao Prof. Dr. Pablo Siqueira e ao Prof. Dr. Allan Kardec, aos funcionários, aos demais colegas do DMC e DPM pelo apoio, paciência e pela ajuda durante todos estes meses de convivência.

Gostaria de fazer agora um agradecimento muito especial, à uma mulher LINDA, especial e essencial para mim, onde graças a sua dedicação, carinho, amizade, lealdade, paciência, cuidado e amor, conduziram-me durante o desenvolvimento e a conclusão deste trabalho. Por isto e por muito mais, gostaria de dedicar este trabalho também para você Fabiana, minha Fá, sem dúvidas, a realização de meu maior sonho e a certeza da realização de todos os demais.

Por fim gostaria de agradecer a Deus, pela inteligência, pela saúde, pela paz, e principalmente por jamais ter deixado que eu perdesse a fé em buscar e realizar meus sonhos.

*Nunca se afaste de seus sonhos.
Porque, se eles se forem,
você continuará vivendo,
mas terá deixado de existir ...
(Mark Twain)*

Sumário

1. INTRODUÇÃO.....	1
2. REVISÃO DE LITERATURA.....	5
2.1 - Introdução.....	5
2.2 – Desenvolvimento.....	6
3. SUSPENSÕES AUTOMOTIVAS: PASSIVAS E ATIVAS.....	21
3.1 - Introdução.....	21
3.2 – Sistema de Suspensão.....	22
3.3 – Suspensões Passivas.....	24
3.3.1 – Modelagem Matemática.....	30
3.4 – Suspensões Ativas.....	36
3.4.1 – Modelagem Matemática.....	42
3.4.2 – Controle ótimo para suspensões ativas:	44
4. REDES NEURAIS.....	47
4.1 - Introdução.....	47
4.2 – Redes Neurais Artificiais (RNA's)	48
4.3 – Características Principais.....	50
4.4 – Modelo de um Neurônio.....	51
4.4.1 – Tipos de Funções de Ativação.....	53

4.5 – Redes Neurais Tipo Perceptron Multicamadas (MLP's).....	55
4.6 – Aprendizado Supervisionado.....	57
4.7 – Algoritmos de Treinamento.....	58
4.7.1 – O <i>Backpropagation</i> - Algoritmo de Retropropagação.....	59
4.7.2 – Critérios de Parada.....	70
4.7.3 – Métodos de Segunda Ordem.....	71
5. IDENTIFICAÇÃO E CONTROLE DE SISTEMAS USANDO REDES NEURAIS.....	75
5.1 - Introdução.....	75
5.2 – Redes Neurais na Identificação de Sistemas Dinâmicos.....	76
5.3 – Neuro-Veículo.....	80
5.4 – Redes Neurais em Sistemas de Controle.....	83
5.5 – Neuro-Controlador.....	86
6. SIMULAÇÕES E RESULTADOS.....	93
6.1 - Introdução.....	93
6.2 – Considerações Iniciais.....	94
6.2.1 –Excitação em degrau.....	95
6.2.2 –Excitação tipo pulso.....	96
6.3 – Identificação Off-Line da Dinâmica Direta de Uma Suspensão Ativa Veicular (Treinamento do Neuro-Veículo).....	98
6.4 – Neuro-Controlador Para Uma Suspensão Ativa Veicular.....	102
6.4.1 – Análise dos resultados.....	106
6.4.2 – Construção e análise de um modelo de suspensão ativa não linear.....	111
6.4.3 – Alterando o modelo de referência e os parâmetros do sistema original.....	124
7. CONCLUSÕES E PERSPECTIVAS FUTURAS.....	132
8. REFERÊNCIAS BIBLIOGRÁFICAS.....	135

ANEXOS.....	141
A.1 – Obtenção das Matrizes de Ponderação.....	141
A.2 – Algoritmo de Implementação do <i>Backpropagation</i>	148
A.2.1 - Processo Iterativo I – Método Padrão-a-Padrão.....	148
A.2.2 - Processo Iterativo II – Método em Lote ou Batelada.....	149
A.3 – Programas.....	150
A.4 – Redes Neurais.....	155

Resumo

ANDRADE, Antonio Fernando Abreu de, *Controle de Suspensões Ativas Utilizando Redes Neurais*, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2001. 155 p. Dissertação (Mestrado)

Neste trabalho, é feito um estudo sobre a aplicação de redes neurais artificiais na identificação e no controle de suspensões ativas não lineares. Considerando um modelo de um quarto de veículo, a modelagem matemática de sistemas passivos e ativos é detalhada. Noções básicas da aplicação das redes neurais, arquitetura, tipos de aprendizado, algoritmos de treinamento, etc., é apresentado, assim como, sua aplicação atual na identificação e controle de sistemas dinâmicos não lineares. O neuro-controlador proposto é do tipo “baseado em um modelo de referência”, e seus pesos são ajustados utilizando um novo conceito de mapeamento inverso através do emprego de um neuro-modelo (neuro-veículo) para a retropropagação do sinal de erro (diferença entre a saída de referência desejada e a saída da rede neural). Simulações computacionais são realizadas com a finalidade de testar o modelo proposto, através da análise do espaço de trabalho da suspensão, da aceleração da massa suspensa e da força de contato do pneu com o solo, principais parâmetros utilizados no projeto de suspensões automotivas. Os resultados demonstram o poder das redes neurais na identificação e no controle de sistemas dinâmicos com características não lineares.

Palavras Chave

Redes neurais artificiais, Suspensões ativas, Sistemas não lineares, Controle e Identificação.

Abstract

ANDRADE, Antonio Fernando Abreu de, *Controle de Suspensões Ativas Utilizando Redes Neurais*, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2001. 155 p. Dissertação (Mestrado)

This work presents a study about applications of artificial neural networks in the identification and control of nonlinear active suspensions. Considering an one-quarter vehicle model, the mathematical modeling of passive and active systems is detailed. Basic concepts of neural networks application, architecture, type of learning, training algorithms, etc., are presented as well as the application in identification and control of nonlinear dynamic systems which is used in this work. The proposed neuro-controller is of type “*model reference neuro-controller*”, and their weights are set using a new concept of inverse mapping through the use of a neuro-model (neuro-vehicle) to a *backpropagation* of the error signal (difference between the output of the desired reference and the output of the neural network). Computational simulations are performed in order to test the proposed model, through the analysis of the suspensions work space, acceleration of the sprung mass and contact force of the tire with the ground, which are the main parameters used in the design of automotive suspensions. The results show the suitability of neural network in the identification and control of dynamic systems with nonlinear characteristics.

Key Words

Neural networks, Active suspensions, Nonlinear systems, Control and Identification.

Lista de Figuras

Figura 3.1 – Sistema de suspensão (Modelo de $\frac{1}{4}$ de veículo).....	22
Figura 3.2 – Tipos de molas.....	25
Figura 3.3 – Sistema de suspensão: (a) apenas com mola e (b) com a mola e o amortecedor.....	26
Figura 3.4 – Amortecedor.....	27
Figura 3.5 – Diagrama do sistema passivo.....	30
Figura 3.6 – Modelo de uma suspensão ativa.....	37
Figura 3.7 – Resposta em frequência do sistema de suspensão de $\frac{1}{2}$ veículo (Karnopp, 1974)....	38
Figura 3.8 – Diagrama do sistema ativo.....	42
Figura 4.1 – Modelo de um neurônio tipo <i>perceptron</i>	51
Figura 4.2 – Funções de Ativação: (a) função sinal, (b) função semi-linear, (c) função sigmoideal e (d) função tangente hiperbólica.....	55
Figura 4.3 – Arquitetura de uma rede MLP com duas camadas intermediárias.....	56
Figura 4.4 – Diagrama de blocos de um sistema com aprendizado supervisionado.....	57
Figura 4.5 – Neurônio de saída <i>j</i> de uma rede MLP	59
Figura 4.6 – Neurônio <i>j</i> da camada intermediária conectado a um neurônio da camada de saída <i>k</i>	66
Figura 5.1 – Diagrama de blocos da identificação de um sistema.....	78
Figura 5.2 – Diagrama de blocos do sistema de um veículo.....	81
Figura 5.3 – Diagrama de blocos do treinamento do neuro-veículo.....	82
Figura 5.4 – Sistema de controle genérico.....	83
Figura 5.5 – Processo dinâmico.....	84
Figura 5.6 – Diagrama de blocos do treinamento do neuro-controlador.....	87
Figura 5.7 – Rede neural: Neuro-Controlador + Neuro-Veículo.....	88

Figura 6.1 – Janela de entrada de dados para as simulações.....	95
Figure 6.2 – Excitação tipo pulso.....	97
Figure 6.3 – Excitação em lombada.....	97
Figura 6.4 – Treinamento do neuro-veículo (EQM = 9.95×10^{-5}).....	100
Figura 6.5 – Resposta à excitação tipo pulso. CI = [0 0 0 0].....	100
Figura 6.6 – Resposta à excitação tipo pulso. CI = [0.05 -0.12 -0.74 0.52].....	101
Figura 6.7 – Resposta à excitação em lombada. CI = [0 0 0 0].....	101
Figura 6.8 – Resposta à excitação em lombada. CI = [0.05 -0.12 -0.74 0.52].....	102
Figura 6.9 - Treinamento do neuro-controlador via <i>Backpropagation</i>	104
Figura 6.10 – Lei de controle à excitação tipo pulso. CI = [0.05 -0.20 0 0].....	105
Figura 6.11 – Lei de controle à excitação em lombada. CI = [0 0 0 0].....	105
Figura 6.12 – Resposta à excitação tipo pulso (espaço de trabalho da suspensão). CI = [0.05 0 -0.20 0].....	107
Figura 6.13 – Resposta à excitação tipo pulso (aceleração da massa suspensa). CI = [0.05 0 -0.20 0].....	108
Figura 6.14 – Resposta à excitação tipo pulso (força de contato do pneu com a via). CI = [0.05 0 -0.20 0].....	109
Figura 6.15 – Resposta à excitação em lombada (espaço de trabalho da suspensão). CI = [0 0 0 0].....	110
Figura 6.16 – Resposta à excitação em lombada (aceleração da massa suspensa). CI = [0 0 0 0].....	110
Figura 6.17 – Resposta à excitação em lombada (força de contato do pneu com a via). CI = [0 0 0 0].....	111
Figura 6.18 – Mola não linear.....	113
Figura 6.19 – Treinamento do neuro-veículo (EQM = 8.154×10^{-4}).....	114
Figura 6.20 – Resposta à excitação em lombada. CI = [0 0 0 0].....	115
Figura 6.21 – Treinamento do neuro-veículo (EQM = 1.6×10^{-5}).....	116
Figura 6.22 – Resposta à excitação tipo pulso. CI = [0.05 0 -0.2 0].....	116
Figura 6.23 – Resposta à excitação em lombada. CI = [0 0 0 0].....	117
Figura 6.24 - Treinamento do neuro-controlador via <i>Backpropagation</i>	118
Figura 6.25 – Lei de controle à excitação tipo pulso. CI = [0.05 0 -0.2 0].....	118
Figura 6.26 – Lei de controle à excitação em lombada. CI = [0 0 0 0].....	119

Figura 6.27 – Resposta à excitação tipo pulso (espaço de trabalho da suspensão). CI = [0.05 0 -0.20 0].....	120
Figura 6.28 – Resposta à excitação tipo pulso (aceleração da massa suspensa). CI = [0.05 0 -0.20 0].....	121
Figura 6.29 – Resposta à excitação tipo pulso (força de contato do pneu com a via). CI = [0.05 0 -0.20 0].....	121
Figura 6.30 – Resposta à excitação em lombada (espaço de trabalho da suspensão). CI = [0 0 0 0].....	122
Figura 6.31 – Resposta à excitação em lombada (aceleração da massa suspensa). CI = [0 0 0 0].....	123
Figura 6.32 – Resposta à excitação em lombada (força de contato do pneu com a via). CI = [0 0 0 0].....	123
Figura 6.33 - Treinamento do neuro-controlador para o novo modelo de referência.....	125
Figura 6.34 – Lei de controle à excitação tipo pulso. CI = [0.01 0 -0.2 0].....	125
Figura 6.35 – Resposta à excitação tipo pulso (espaço de trabalho da suspensão). CI = [0.05 0 -0.2 0].....	126
Figura 6.36 – Resposta à excitação tipo pulso (aceleração da massa suspensa). CI = [0.05 0 -0.2 0].....	127
Figura 6.37 – Resposta à excitação tipo pulso (força de contato do pneu com a via). CI = [0.05 0 -0.2 0].....	127
Figura 6.38 – Lei de controle à excitação em lombada. CI = [0 0 0 0].....	128
Figura 6.39 – Resposta à excitação em lombada (espaço de trabalho da suspensão). CI = [0 0 0 0].....	129
Figura 6.40 – Resposta à excitação em lombada (aceleração da massa suspensa). CI = [0 0 0 0].....	129
Figura 6.41 – Resposta à excitação em lombada (força de contato do pneu com a via). CI = [0 0 0 0].....	130

Lista de Tabelas

Tabela 3.1 – Variáveis utilizadas no modelo passivo.....	31
Tabela 6.1 – Valores dos parâmetros usadas na simulação.....	94

Nomenclatura

Letras Latinas

- A** - matriz do sistema ou matriz de estado
- A_d** - matriz de estado discretizada
- B** - matriz de controle
- b** - número de neurônios da segunda camada intermediária do neuro-controlador
- B_d** - matriz de controle discretizada
- C _{\ddot{X}_1}** - vetor usado para representar a ponderação de \ddot{X}_1
- C_{ES}** - vetor usado para representar a ponderação de ES
- C_{F_p}** - vetor usado para representar a ponderação de F_p
- C_u** - vetor usado para representar a ponderação de **u**
- C** - conjunto que contém todos os neurônios da camada de saída da RNA
- C** - matriz de saída
- ca** - fator de amortecimento do amortecedor da suspensão
- e(n)** - sinal de erro para um neurônio qualquer
- f** - força exercida pelo atuador
- f_s** - força dinâmica da mola não linear
- f(.)** - função de ativação
- J** - função de custo
- K** - matriz de ganho (LQR)
- k, j** - índices que representam neurônios da RNA

k_1, k_2, k_3 e k_4 - coeficientes da matriz de ganho (LQR)
 k_p - constante de rigidez do pneu
 k_s - constante de rigidez da mola da suspensão
 k_{ms} - constante de rigidez equivalente da mola não linear da suspensão
 L - matriz que multiplica o vetor de excitação
 L_d - matriz que multiplica o vetor de excitação discretizada
 M - matriz de controlabilidade
 m - número de neurônios da camada intermediária do neuro-veículo
 M_s - massa suspensa ($1/4$ de veículo)
 μ - massa do sistema eixo-roda ($1/4$ de veículo)
 N - matriz de ponderação dos elementos cruzados (variáveis de estado e força de controle)
 n - número de interações (épocas)
 N - número total de padrões de treinamento
 n - ordem do sistema
 n_e - número de neurônios da camada de entrada do neuro-controlador
 nl - expoente que representa a não linearidade da mola
 o - número de entradas do neuro-veículo
 p - número de neurônios da camada de saída do neuro-veículo
 P - número de pesos ajustáveis da RNA
 p_e - número total de entradas (excluindo a polarização)
 $P_{\ddot{X}_1}$ - representação da ponderação de \ddot{X}_1
 P_{ES} - representação da ponderação de ES
 P_{F_p} - representação da ponderação de F_p
 P_u - representação da ponderação de u
 Q - matriz de ponderação dos estados
 q - número de neurônios da primeira camada intermediária do neuro-controlador
 Q_c - matriz de observabilidade
 r - excitações impostas a suspensão
 R - matriz de ponderação da força do controlador
 $\text{sgn}(\cdot)$ - sinal da força da mola não linear
 $u(t)$ - vetor de controle

$w(t)$ - vetor de excitação
 $w_{k0}, w_{k1}, \dots, w_{kp}$ - pesos sinápticos de um neurônio qualquer
 W_{NC1} - matriz de pesos da segunda camada intermediária do neuro-controlador
 W_{NC2} - matriz de pesos da primeira camada intermediária do neuro-controlador
 W_{NC3} - matriz de pesos da camada de saída do neuro-controlador
 W_{NV1} - matriz de pesos da camada de saída do neuro-veículo
 W_{NV2} - matriz de pesos da camada intermediária do neuro-veículo
 x_0, x_1, \dots, x_p - sinais de entrada de um neurônio qualquer
 X_1 - deslocamento vertical da massa suspensa
 X_2 - deslocamento vertical da massa do sistema eixo-roda
 \dot{X}_2 - velocidade da massa do sistema eixo-roda
 \dot{X}_1 - velocidade da massa suspensa
 \ddot{X}_2 - aceleração da massa do sistema eixo-roda
 \ddot{X}_1 - aceleração da massa suspensa
 \hat{y} - saída do neuro-sistema
 $y(n)$ - saída de um neurônio qualquer
 $y(t)$ - vetor de saída
 $Yd(n)$ - saída de um neurônio qualquer
 \hat{z} - vetor de estado estimado pela rede neural
 $\dot{z}(t)$ - derivada de primeira ordem do vetor de estado
 $z(t)$ - vetor de estado
 z_r - saída do modelo de referência

Letras Gregas

Δt - intervalo de amostragem

α - fator de correção

Δx - movimento relativo de ambos os extremos da mola não linear

$\varepsilon(\mathbf{n})$ - soma dos erros quadráticos instantâneos da RNA

ε_{av} - erro quadrático médio

$\eta(\mathbf{n})$ - passo ou taxa de aprendizagem

$\beta(\mathbf{n})$ - coeficiente de momento

$\delta(\mathbf{n})$ - gradiente local de um neurônio qualquer

$\nabla \varepsilon$ - vetor gradiente

ξ, τ - limiares

Φ, Ψ - funções não lineares

Δ - atraso da saída do sistema

Superescritos

T - transposto

Subscritos

ES - espaço de trabalho da suspensão

F_p - força de contato do pneu com o solo

NV - neuro-veículo

NC - neuro-controlador

Capítulo 1

Introdução

Desde a criação do automóvel por volta do final do século XIX concentrou-se esforços para tornar o mesmo mais confortável e seguro. Após mais de um século do surgimento dos veículos automotivos, ocorreram grandes avanços nas tecnologias empregadas na indústria automobilística. Sistemas de suspensão automotivas têm sido alvo de estudos analíticos e experimentais, sempre no sentido de melhorar os critérios de projeto de uma suspensão, ou seja, a medida que os carros estão se tornando cada vez mais velozes, o conforto e a segurança dos passageiros devem ser garantidos sob todas as condições de operação. Estes estudos demonstraram que a utilização de suspensões contendo somente molas e amortecedores passivos não permite um aumento do desempenho do sistema devido ao fato destes elementos possuírem características fixas.

Assim sendo, surge como uma alternativa para o aumento desse desempenho, a introdução de atuadores ativos e semi-ativos nos sistemas de suspensão dos automóveis, através do emprego da teoria de controle moderno. Além disso, com o grande desenvolvimento nos últimos anos da micro-eletrônica e da informática, a realização física de tais sistemas tem-se tornado possível, permitindo assim o emprego destas novas tecnologias nos veículos convencionais.

Outra área cujas pesquisas recentes tem provocado relativo entusiasmo é a dos sistemas neurais artificiais ou simplesmente redes neurais artificiais (RNA's). Inspirada no ser humano, esta nova área de pesquisa objetiva o desenvolvimento de sistemas artificiais equivalentes aos

sistemas cerebrais que controlam e gerenciam o corpo e suas atividades. Tais sistemas neurais possuem a capacidade de “aprender”, onde através de leis de aprendizado são capazes de realizar tarefas complexas. As RNA's tem recentemente atraído uma grande atenção da comunidade científica devido a sua versatilidade e capacidade de mapeamento entrada-saída, junto com a habilidade de processamento paralelo e coletivo, aumentando as expectativas da aplicação das RNA's para resolver muitos tipos de problemas relacionados à sistemas dinâmicos. Estas aplicações incluem a identificação e o controle de sistemas dinâmico não lineares, onde estão incluídas as suspensões ativas veiculares.

Dentro deste contexto, surge a possibilidade de aplicação das RNA's na identificação e no controle do sistema ativo de suspensão veicular. Neste trabalho, um novo conceito de mapeamento inverso foi desenvolvido para o projeto de um controlador neural, utilizando para isto, um modelo do sistema de suspensão, identificado através de uma rede neural, sem a necessidade de técnicas de linearização, e que será empregado para retropropagar o erro durante a atualização dos pesos sinápticos da rede de controle. O neuro-controlador proposto baseia-se em um modelo de referência para conduzir a saída do sistema a um comportamento satisfatório que atenda aos critérios de segurança e conforto do projeto de suspensões veiculares. Foram realizadas simulações num modelo de suspensão passiva e ativa de $\frac{1}{4}$ de veículo sob determinadas condições de operação, a fim de testar a capacidade das redes neurais em realizar as tarefas propostas.

Na elaboração deste trabalho, foi feita inicialmente, no capítulo 2, uma revisão dos trabalhos na área de controle ativo usando redes neurais artificiais (RNA's), aplicadas em suspensões de veículos automotivos, ou seja, suspensões ativas veiculares. Os três principais tipos de sistemas de suspensão são apresentados e alguns trabalhos realizam uma análise comparativa destes, indicando as principais vantagens e desvantagens envolvendo cada um deles, assim como o uso das RNA's na identificação e no controle de tais sistemas.

No capítulo 3, é feito um estudo comparativo sobre os principais tipos de suspensões automotivas, destacando os sistemas passivos e ativos. Foi feita uma introdução do funcionamento, aplicação, vantagens e desvantagens de cada sistema. Foram deduzidas também, as equações de movimento, de ambos os modelos de suspensão, de um sistema de dois graus de

liberdade, representado por um modelo $\frac{1}{4}$ de veículo, através de uma análise dinâmica no espaço de estados, para se obter as respostas dos mesmos à diferentes tipos de excitações que caracterizam as condições de operação.

No capítulo 4, é apresentado um estudo da aplicação das redes neurais artificiais (**RNA's**) e de suas características principais. É desenvolvido um modelo matemático com tratamento vetorial para as **RNA's**, destacando as redes neurais do tipo *perceptron* multicamadas (**MLP's**), juntamente com os respectivos algoritmos de aprendizado baseado em um método de minimização de uma dada função de custo, com a inclusão da retropropagação (*backpropagation*) de sinais através da rede. Destacam-se dentre as técnicas de aprendizado utilizadas, aquelas baseados em métodos de otimização de primeira ordem, gradiente com *backpropagation* (**BPA**), ou de segunda ordem, gradiente conjugado escalonado modificado (**SCGM**), que são empregados para ajustar os pesos sinápticos das **RNA's** durante a fase de treinamento das mesmas.

No capítulo 5, será mostrado como o paradigma das redes neurais pode ser aplicado na identificação e controle de sistemas dinâmicos lineares e não lineares, especificamente no caso de sistemas ativos não lineares de suspensões veiculares. Inicialmente é dada uma visão geral do processo de identificação de sistemas, utilizando as **RNA's** para predizer as características dinâmicas do veículo. No caso das suspensões é utilizado um modelo chamado “neuro-veículo” para esta representação, sendo posteriormente utilizado no treinamento do neuro-controlador proposto. Para o controle do sistema de suspensão ativa de um veículo é proposto um neuro-controlador, cujo objetivo final é conduzir a saída do sistema a uma resposta satisfatória segundo um dado modelo de referência, utilizando as **RNA's** para que seja feito um processo de aproximação da dinâmica inversa do mesmo, obtendo assim a lei de controle que atuará sobre a dinâmica do sistema.

No capítulo 6, são feitas as simulações computacionais dos modelos propostas anteriormente no capítulo 3, empregando a metodologia apresentada e desenvolvida nos capítulos 4 e 5. Foi analisado também um novo modelo proposto, com a substituição da mola linear do modelo original, por outra, com características não lineares, visando demonstrar a capacidade das redes neurais quando empregadas na identificação e no controle de sistemas não lineares. Primeiramente foi feita a identificação *off-line* da dinâmica de uma suspensão ativa, utilizando

uma rede **MLP**, para em seguida ser discutida a performance do neuro-controlador proposto, seu treinamento e os testes necessários à verificação de seu desempenho segundo os critérios de projeto de um sistema de suspensão automotiva. Destacou-se a capacidade do neuro-controlador em realizar um processo de aproximação da dinâmica inversa do sistema, sua adaptatividade em relação à variações dos parâmetros e a identificação de sistemas não lineares, sem a necessidade do emprego de técnicas de linearização.

Por fim, nos Anexos, encontram-se os programas e os algoritmos utilizados neste trabalho, como também a metodologia empregada na determinação das matrizes de ponderação que serão utilizadas no cálculo da matriz de ganho do controlador **LQR**.

Capítulo 2

Revisão de Literatura

2.1 - Introdução

Este capítulo apresenta uma revisão dos trabalhos na área de controle ativo utilizando redes neurais artificiais (RNA's), aplicadas em suspensões de veículos automotivos, ou seja, suspensões ativas veiculares.

Os modelos matemáticos envolvidos vão desde um quarto de veículo, composto de duas massas, até o modelo tridimensional, compreendendo a massa total do corpo do veículo e as quatro rodas. Os três principais tipos de sistemas de suspensão são apresentados e alguns trabalhos realizam uma análise comparativa destes, indicando as principais vantagens e desvantagens envolvendo cada um dos modelos.

Noções básicas da aplicação das redes neurais são descritas, assim como o grande desenvolvimento das pesquisas nos últimos anos na área de identificação e controle de sistemas dinâmicos não lineares. Os principais tipos de neuro-controladores são discutidos através de uma ampla revisão bibliográfica, com destaque para o controle “baseado num modelo de referência”, uma técnica de controle ativo na qual a lei de controle é obtida, segundo um dado modelo escolhido através de critérios de projeto, a partir da retropropagação do erro na saída da rede neural, até a atualização de sus parâmetros internos.

2.2 – Desenvolvimento

Os primeiros trabalhos em suspensão ativa surgiram no início dos anos 70, mostrando um novo modelo de suspensão que pudesse aumentar o desempenho do sistema com relação ao conforto e a segurança dos passageiros. Estes dois conceitos empregados para uma mesma suspensão são conflitantes, uma vez que a otimização de um conceito gera a diminuição do desempenho do outro. Para projetar uma suspensão automotiva, um estudo de caso deve ser feito a “priori”, com a finalidade de estabelecer uma relação entre os conceitos de segurança e conforto.

Nos últimos anos, a melhoria dos sistemas de suspensão de veículos foi um assunto de intensa pesquisa e desenvolvimento. Este fato tem basicamente duas motivações: comercial e científica. A principal causa do aumento da atividade comercial foi o crescente interesse dos fabricantes de veículo e fornecedores de componentes, que desejam melhorar o desempenho e a qualidade dos seus produtos. Da comunidade científica, os pesquisadores de tais sistemas perceberam que o controle automático de suspensões veiculares seria uma meta óbvia e desejável que poderia ser perfeitamente realizável devido sobretudo ao desenvolvimento de novos atuadores, sensores e uma eletrônica de baixo custo. O desenvolvimento do controle de uma suspensão é uma tarefa interessante e desafiadora, sobretudo pela gama de características de desempenho que um bom sistema de suspensão tem que alcançar. As características desejáveis são:

- Regular o movimento do corpo do veículo: a suspensão ideal deve isolar o corpo do veículo das perturbações das estradas;
- Controle do espaço de trabalho da suspensão: O deslocamento excessivo da roda na direção vertical resultará em um comportamento não desejado do pneu em relação a estrada, ou seja, poderá haver perda de aderência e conseqüentemente problemas de dirigibilidade e estabilidade;

- Distribuição da força de contato do pneu com o solo: o contato ótimo do pneu-para-estrada deve ser mantido em todas as quatro rodas.

Como foi dito anteriormente estas características são às vezes contraditórias e não são conhecidas em um sistema de suspensão convencional sob todas as condições de operação. Isto pode ser estudado através de modelos matemáticos e simulações, principalmente em relação à identificação das características dinâmicas e o controle de tais sistemas, utilizando para isto modelos lineares ou não lineares de complexidade variável de um sistema de suspensão veicular. A natureza de qualquer solução para o controle de suspensões dependerá de alguns fatores como custo, desempenho, componentes, implementação, mercado, etc. Estas podem ser classificadas de acordo com a presença ou não de fontes adicionais de energia em passivas, semi-ativas e ativas. Especificamente, uma suspensão passiva é um sistema convencional de fontes não controladas de energia, tais como molas e amortecedores. Num sistema semi-ativo é observado um controle sobre o fator de amortecimento que pode ser variável de acordo com as necessidades e cargas que atuam sobre o sistema. Os amortecedores apresentam coeficientes de amortecimento eletronicamente selecionáveis que podem ser modificados, amolecendo ou endurecendo a suspensão, visando um desempenho ótimo. Uma suspensão ativa é caracterizada pela substituição dos componentes passivos por atuadores que empregam forças adicionais ao sistema. Os sensores disponíveis para um sistema de suspensão ativo variam muito, mas podemos citar os acelerômetros de centro de roda, sensores de desvio da suspensão, células de carga etc. Tais sistemas são capazes de trabalhar numa ampla faixa de frequências, sendo porém de difícil implementação e custo elevado quando comparadas com os demais tipos. Como em todos os aspectos de projeto de automóveis, o custo dos componentes é um assunto primário, como também é o aspecto da satisfação do cliente, onde inclui custo baixo de manutenção, baixo nível de barulho, conforto e segurança, o que tem dificultado a utilização das suspensões ativas em grande escala. Pode-se encontrar na literatura inúmeros trabalhos que realizam um estudo detalhado dos diferentes sistemas de suspensão, suas aplicações, desempenho e viabilidade econômica de implementação.

Karnopp, et al., 1974 desenvolveu um trabalho sobre o controle de vibrações em suspensões automotivas, onde foi feita a modelagem matemática e a comparação do desempenho de suspensões passivas, semi-ativas e ativas, em relação aos critérios de projeto (conforto e

segurança). Simulações computacionais permitem um estudo de conceitos físicos que são utilizados na análise das suspensões citadas anteriormente. Levando-se em conta as vantagens e desvantagens dos três tipos de suspensões, as do tipo semi-ativa sobressaíram em relação aos demais, devido principalmente ao baixo custo de instalação e de manutenção.

Nagai, 1993, revisa os problemas básicos das suspensões automotivas, os principais tipos de controle e toda a abordagem teórica para a implementação das suspensões ativas em sistemas práticos. Modelos de $\frac{1}{4}$ e $\frac{1}{2}$ veículo são usados para as simulações computacionais, permitindo uma análise comparativa entre as suspensões passivas, semi-ativas e ativas. Uma análise dos principais componentes das suspensões ativas é feita, assim como, uma classificação das mesmas segundo tais componentes. O controle empregado é baseado na teoria de controle ótimo e os resultados encontrados são analisados levando-se em conta a relação custo/benefício da implementação prática de modelos ativos nos automóveis convencionais.

Tan, 1997, analisou as vantagens do crescente interesse na utilização dos sistemas de suspensão ativa em automóveis convencionais nos últimos anos. Uma suspensão ativa completa baseada em atuadores hidráulicos é vista como uma excelente alternativa para atender todas estas expectativas. Contudo, as características de tais sistemas quando submetidas a elevadas frequências, são consideradas como um problemas para a implementação prática de tais sistemas. Uma modelagem matemática de alta fidelidade, capturando as características dinâmicas de uma suspensão ativa hidráulica, é requerida para o desenvolvimento e análise de tal abordagem. Neste trabalho, o modelo de $\frac{1}{4}$ de veículo de um sistema ativo hidráulico de suspensão é analisado. Este quantifica a contribuição dos componentes mecânicos necessários para atender as exigências de projeto de uma suspensão ativa e seu comportamento quando em altas frequências. Os parâmetros da identificação do modelo em resposta a uma excitação padrão são apresentados. A eficiência do modelo proposto como ferramenta para a identificação e controle de uma suspensão ativa é comprovada.

Williams, 1997 publicou outro trabalho comparativo entre os três tipos de suspensões automotivas, subdividido em duas partes. A primeira revisa os compromissos requeridos no projeto de uma suspensão passiva convencional, com todo o equacionamento e simulações envolvendo modelos de $\frac{1}{4}$ de veículo. Foi analisado também como aqueles compromissos podem

ser mudados pela inclusão de componentes ativos. No caso das suspensões semi-ativas o estudo foi baseado no amortecedor tipo “*skyhook*”, comparando seu desempenho com o dos amortecedores convencionais. Para as suspensões ativas, todos os componentes passivos foram substituídos por um atuador, capaz de impor ao sistema uma força de mesma intensidade, porém de sentidos opostos, na massa suspensa e na massa do sistema eixo/roda. O controlador é do tipo “regulador linear quadrático” (**LQR**), que segue um esquema de controle em malha fechada com realimentação de estados. Para modelos altamente não lineares o **LQR** torna-se de difícil ou impossível implementação. A segunda parte do trabalho discute o “hardware” empregado (principais componentes e equipamentos), incluindo todo o equacionamento necessário às simulações e testes dos diferentes tipos de suspensões. Os benefícios de cada uma das tecnologias empregadas é discutido juntamente com uma análise de suas principais vantagens e desvantagens.

Hrovat, 1997, pesquisou as aplicações das técnicas de controle ótimo para o projeto de suspensões ativas, partindo de modelos simples de $\frac{1}{4}$ de veículo, até modelos mais complexos como o de $\frac{1}{2}$ veículo e o modelo completo. Enquanto o enfoque principal é o controle ótimo linear quadrático (**LQ**) e suas aplicações em sistemas de suspensão ativos, este trabalho também analisa o comportamento dos sistemas passivos e semi-ativos, e as técnicas de controle não linear adaptativo e robusto, realizando algumas considerações práticas importantes na implementação e desenvolvimento destes modelos nos últimos anos.

Hwang et al., 1998, destaca que os sistemas semi-ativos, devido a capacidade de alcançar uma qualidade elevada no conforto dos passageiros dos automóveis, com baixo custo de implementação e manutenção deve ser adotada num futuro próximo nos automóveis convencionais. A chave para o sucesso é desenvolver controladores que alcancem um conforto elevado em relação aos sistemas passivos sem afetar com isso os critérios de segurança. Com esta finalidade um controlador prático para sistemas de suspensão semi-ativas com amortecimento variável está sendo desenvolvido. O controlador foi projetado para rapidamente alcançar um desempenho satisfatório a partir de um algoritmo modificado do controle tipo “*skyhook*”. Este algoritmo foi testado em um simulador “*hardware-in-the-loop*”, onde o veículo em tempo real é submetido a condições de operação e sua resposta dinâmica é analisada. Foi adotado o modelo matemático de $\frac{1}{4}$ de veículo para a realização de tais simulações.

Através destes trabalhos pode-se observar o avanço nas pesquisas e no desenvolvimento do controle eletrônico do desempenho de suspensões, sobretudo nos últimos anos. Isto ocorreu basicamente devido a redução do custo dos sensores e microcomputadores disponíveis atualmente. Alguns exemplos de suspensões eletronicamente controladas, ditas “adaptáveis” tiveram grande desenvolvimento e passaram a ser produzidas sobretudo na última década.

As redes neurais artificiais (RNA's) também conhecidas como processadores de aprendizado conexionista e distribuidores em paralelo, encontram atualmente aplicações em várias áreas: muitos ramos da engenharia, medicina, arqueologia, finanças, etc. Nestas áreas, existem duas categorias de problemas que têm sido resolvidos com sucesso através da metodologia das RNA's. São a identificação de sistemas e o controle da dinâmica dos mesmos.

O tipo de rede neural mais comumente utilizado é a rede neural multicamada do tipo MLP. Métodos de aprendizado supervisionado, onde a rede neural é treinada para aprender com os padrões entrada/saída apresentados a ela, são tipicamente usados. Mais freqüentemente, versões do algoritmo *backpropagation* são usadas para ajustar os pesos da rede neural durante o treinamento; isto é feito usualmente com um grande gasto de tempo de convergência. As funções de ativação mais freqüentemente usadas são as sigmoidais, mas elas podem ser também funções sinal ou Guassianas.

Uma propriedade das redes multicamadas é fundamental para muitas aplicações de controle. Tais redes podem gerar mapeamentos entrada/saída os quais podem aproximar qualquer função com uma precisão desejada. Para modelar as características de um sistema dinâmico, a rede neural é treinada usando dados de entradas/saída e os pesos da rede neural são ajustados usando um algoritmo de treinamento. Devido a aplicações típicas envolvendo sistemas não lineares, a rede neural é treinada para classes de entradas e condições iniciais. O mapeamento estático gerado por uma rede neural pode adequadamente representar as características do sistema em um intervalo de interesse para uma aplicação particular. Fica evidente então, a questão de como uma rede neural, a qual realiza um mapeamento estático, pode representar as características entrada/saída de um sistema dinâmico. Para que isto seja possível, deve ser fornecida à rede neural informações sobre a história do sistema – tipicamente atrasos das entradas e saídas. O número de atrasos empregados, implica em uma maior precisão e está associado com a ordem do

sistema. Existe uma relação entre precisão e complexidade computacional do treinamento, conseqüentemente o número de entradas usadas afeta o número de pesos na rede neural e subseqüentemente o tempo de treinamento. A arquitetura empregada, assim como o número de neurônios nas camadas intermediárias, baseia-se em critérios empíricos e acompanham o grau de aproximação desejado.

A tarefa de um projetista de sistemas de controle é encontrar leis de controle, as quais direcionem um dado sistema a um estado final desejado de uma maneira específica. Esta tarefa requer o conhecimento do sistema que o controlador esta tentando influenciar e a forma deste controlador. Encontramos na literatura várias abordagens para o projeto de controladores baseados em redes neurais. Alguns destes, assumem um explícito conhecimento do sistema a ser controlado. Dentre estes citaremos alguns aplicados a sistemas dinâmicos, sobretudo suspensões automotivas (Balakrishnan, 1996):

- Controle Supervisionado – Neste tipo, o controlador neural aprende a reproduzir uma tarefa conhecida. Esta abordagem requer um vetor de entrada e um correspondente vetor de saída desejado. Em um exemplo prático, este pode ser entendido como um sistema controlado capaz de aprender a produzir um controle próprio em resposta a sinais de entrada lidos por sensores. O controle supervisionado pode ser implementado através de muitos métodos de aprendizado supervisionado.
- Controle Direto Inverso – Neste modo o neuro-controlador é treinado para aprender o mapeamento do espaço de estados através dos sinais de controle. As trajetórias dos sistemas são entradas do controlador e as saídas são o controle desejado. O controlador pode ser treinado *off-line* com trajetórias nominais, seja através de simulações, ou através da performance do sistema real. Se não existem um mapeamento pontual entre as coordenadas espaciais e os sinais dos atuadores, como alternativa de projeto, o neuro-controlador é combinado com uma rede neural para representar a dinâmica do sistema.
- Controle Ótimo – No projeto deste tipo de neuro-controlador, a RNA utiliza uma função custo que deve ser minimizada ao longo do tempo. Esta abordagem combina um

modelo das características do sistema com o controlador. Técnicas de retropropagação são usadas para calcular as derivadas da função custo no tempo final com respeito a atual ação de controle. Estas derivadas são usadas no treinamento dos pesos sinápticos da rede.

- Controle Neural Adaptativo – No controle neural adaptativo, a saída do controlador neural origina-se de uma trajetória criada por um modelo de referência desejado pelo projetista. Com um controlador adaptativo baseado em um dado modelo de referência, o neuro-controlador deve aprender a conduzir o sistema, para que o mesmo siga um padrão de comportamento especificado pelo modelo de referência adotado.

Como os problemas de engenharia tornam-se mais complexos com o passar do tempo, observa-se o surgimento de novas abordagens e técnicas para resolvê-los. Devido à sua natureza massivamente paralela, as redes neurais podem ser usadas para encontrar soluções para problemas de elevado custo computacional. Elas podem ser treinadas para se adaptar às mudanças no ambiente e podem ter representações não lineares. Quando uma rede neural multicamada é treinada como um controlador, este pode ser a malha aberta ou fechada, uma entrada apropriada de controle para o sistema é gerada, sendo deduzida de uma conhecida saída desejada do sistema. Para isto, pode-se utilizar uma aproximação baseada em um modelo matemático do sistema, caso disponível, ou um modelo neural da dinâmica do sistema ou então da dinâmica inversa do mesmo. No último caso assume-se que a dinâmica inversa pode ser representada por uma rede neural.

Redes neurais podem ser combinadas para identificar e controlar o sistema, implementando assim um controlador adaptativo. Também é possível adaptativamente mudar o controlador baseado em um sinal adicional de treinamento, o qual é uma indicação de como comporta-se o sistema. Redes neurais podem também ser usadas para detectar e identificar falhas no sistema, e ajuda a armazenar informações para tomadas de decisões em diferentes tipos de sistemas. Em geral, observa-se o grande potencial de aplicação das redes neurais em todos os níveis hierárquicos de controladores inteligentes o que garante um alto grau de autonomia ao sistema.

Tem-se uma grande quantidade de trabalhos que podem ser citados envolvendo a identificação e o controle de sistemas dinâmicos, sobretudo as suspensões ativas automotivas, interesse maior deste estudo. Dentre os principais destacam-se:

Narendra e Parthasarathy, 1990, desenvolveu um dos poucos trabalhos teóricos que utilizam redes neurais para a controlabilidade e estabilidade de sistemas dinâmicos. Este trabalho é restrito à redes multicamadas que utilizam o *backpropagation* dinâmico e sistemas não lineares com acesso completo às informações oriundas do vetor de estado. Os resultados são estendidos à sistemas não linearizáveis. A avaliação do desempenho de ambas abordagens é feita através de funções custo, tais como o erro quadrático médio (EQM). Em ambos os métodos, convencional e conexionista, pode-se ter uma abordagem *on-line* ou *off-line*. Nos métodos *off-line*, todos os dados de treinamento estão disponíveis em um tempo determinado. Nos métodos *on-line* (por exemplo, usado no sistema guia de mísseis) contudo, a característica necessária é o aprendizado em tempo real, sendo assim, o método demonstra sua eficiência na memorização das alterações dos eventos ao longo do tempo. De acordo com Narendra, controladores baseados em redes neurais vão ser efetivos em quatro casos. O primeiro é o projeto de um controlador para plantas não lineares onde formulações matemáticas (analíticas) são bastante complexas. O segundo uso é na identificação de plantas, tais como estruturas espaciais flexíveis, que podem ser melhor acompanhadas através de uma rede neural. O terceiro caso é na área de controle de processos, onde a planta opera com vários *set-points*. O quarto caso é na situação onde o espaço de estados é particionado dentro de regiões disjuntas e a rede neural pode ser utilizada para gerar um controle ótimo correspondente a cada região. Foram apresentados quatro modelos de redes de diferentes complexidade para a identificação e controle de sistemas dinâmicos não lineares e demonstraram através de exemplos simples, as características de cada modelo.

Nguyen e Widrow, 1991, demonstram o poder do *backpropagation através do tempo* para o controle não linear com o celebrado “Truck Backer-Uper”. Sua abordagem inicia com o treinamento de uma rede neural para modelar o *trailer* de um caminhão. O modelo foi treinado através de um *backpropagation* dinâmico para o desenvolvimento de um controlador capaz de conduzir o caminhão até o local de carga e descarga em marcha ré. Simulações computacionais demonstram o poder das redes neurais na identificação e controle de veículos autônomos.

Rivals et al., 1994, aplica duas redes neurais separadamente para controlar as variáveis de estado de um veículo autônomo. Foram utilizadas duas arquiteturas de redes diferentes para a implementação desses controladores chamados, controlador simples de retropropagação de estados (SFC) e controlador de modelo interno (IMC) respectivamente. Este tipo de neuro-controlador requer um neuro-modelo do sistema (uma rede já treinada para representar a dinâmica do sistema) para retropropagar um sinal de erro enquanto os pesos sinápticos do controlador são atualizados. Este modelo de controlador segue um dado modelo de referência gerando uma lei de controle que leva o sistema a desempenhar um comportamento dinâmico próximo a este modelo.

Moran e Nagai, 1994 analisaram o desempenho e o comportamento dinâmico das redes neurais utilizadas para a identificação direta e inversa, e o controle da suspensão ativa de um veículo com características não lineares. Eles compararam o desempenho de um veículo que tem um controlador não linear para suspensão ativa (neuro-controlador) com outro que possui um controlador ótimo linear (LQ) projetado depois da linearização dos componentes da suspensão ao redor do ponto de equilíbrio. O controlador neural proposto consiste em neuro-modelo do veículo (neuro-veículo), um neuro-observador, e dois neuro-controladores para as suspensões dianteiras e traseiras. O neuro-veículo proposto possui uma camada de entrada, duas camadas intermediárias e uma de saída. A entrada possui dois vetores, um vetor de perturbação de estrada e outro vetor representando a força de controle do atuador. A saída do neuro-veículo são as variáveis de estado do mesmo, sendo os vetores de estado das suspensões dianteiras e traseiras. Cada vetor contém quatro variáveis; a deflexão da suspensão e o desvio do pneu e as velocidades das massas suspensa e do sistema eixo/roda. A saída também é atrasada e torna-se uma entrada do neuro-controlador. Os dados usados no treinamento foram obtidos de um modelo de $\frac{1}{2}$ veículo com quatro graus de liberdade com movimento vertical e rotacional. As molas e os amortecedores foram considerados como elementos de características não lineares. O neuro-observador retorna a velocidade de perturbação da estrada entre as suspensões dianteira e traseira depois de processar o vetor de variáveis de estado dianteiro e o sinal de controle dianteiro. O sinal de perturbação da estrada e vetor das variáveis de estado das suspensões dianteira e traseira são processados então pelo neuro-controlador traseiro. Este controlador calcula a lei de controle exigida para a suspensão traseira. O neuro-controlador dianteiro usa apenas o vetor de estado dianteiro e traseiro e calcula o sinal de controle dianteiro.

O objetivo do controlador da suspensão é impor ao veículo um desempenho dinâmico ótimo expresso em termos de conforto do passageiro, estabilidade e dirigibilidade na execução de manobras a fim de garantir a segurança de operação dos mesmos, através da minimização de uma determinada função custo. O erro é retropropagado através do tempo, cuja execução seguiu duas estratégias de treinamento, uma direta e uma recorrente. Os autores informaram que os resultados são melhores quando foi empregado o método de treinamento recorrente, sobretudo na presença de ruídos. Para o neuro-observador, a dinâmica inversa do sistema (suspensão dianteira) foi modelada usando um neuro-veículo treinado pelo método direto. Para o neuro-controlador, a técnica de minimização da função custo foi executada usando a retropropagação do erro através do neuro-veículo para ao treinamento dos pesos do neuro-controlador. Uma comparação entre o desempenho do neuro-controlador proposto, com um sistema de controle ótimo linear, onde os sistemas são excitados com uma entrada randômica, foi realizada. Nesta comparação, o neuro-controlador apresentou um desempenho superior ao do modelo com controle ótimo linear.

Nagai et al., 1995, em outro trabalho, propõem um novo método de projeto do sistema de suspensão veicular considerando as características não lineares dos pneus e dos componentes da suspensão. A integração de redes neurais artificiais com a teoria de controle linear é introduzida para a identificação e controle de um modelo não linear estruturado de um veículo usando um software para análises dinâmicas (ADAMS). Os resultados das simulações computacionais mostraram que a proposta da abordagem não linear consegue atingir uma boa eficácia em relação ao conforto e a segurança dos passageiros.

Choromanski, 1996, desenvolveu um estudo da utilização de redes neurais artificiais aplicadas no controle da dinâmica de veículos ferroviários. As redes foram usadas para o controle adaptativo baseado em um modelo de referência, visando garantir a estabilidade, segurança e uma melhoria de performance dos conjuntos de rodas dos vagões, quando sob altas velocidades em curvas. O controle baseado em um modelo de referência impõe no sistema um comportamento próximo àquele usado no treinamento das redes. Exemplos dos resultados das simulações computacionais são dadas para um modelo não linear de um veículo ferroviário.

Park et al., 1996, apresenta um estudo do emprego de redes neurais multicamadas no projeto de um controlador neural chamado “Optimal Tracking Neuro-Controller” (OTNC), para sistemas não lineares com função de custo quadráticas. O OTNC é composto por dois controladores: um neuro-controlador *feedforward* (FFNC) e outro neuro-controlador *feedback* (FBNC). O FFNC é projetado usando um novo conceito de mapeamento inverso através da utilização de um neuro-modelo (neuro-veículo) para a retropropagação do sinal de erro. Este tipo de controle é dito “baseado em um modelo de referência” pois um modelo cujo comportamento é ótimo, será utilizado no treinamento do neuro-controlador através da retropropagação do erro na saída do neuro-veículo, ou seja, a diferença entre a saída de referência desejada e a saída da rede neural será minimizada segundo uma função de custo, implicando na atualização dos pesos sinápticos do controlador. Os padrões são obtidos a partir de um histórico do sistema e o treinamento ocorre segundo a abordagem *off-line*, usando o *backpropagation* estático como algoritmo de treinamento das redes. Para o treinamento do FBNC foi criado um algoritmo de treinamento baseado na generalização *backpropagation através do tempo*, onde através da minimização de uma dada função custo, os pesos do controlador são treinados. A metodologia proposta destaca-se como um método de controle *off-line*, onde o sistema é primeiro identificado, para em seguida o controlador ser projetado. Um exemplo de um sistema dinâmico típico com características não lineares foi analisado para comprovar a boa performance do controlador proposto.

El-Gindy et al., 1996, afirmam que os recentes desenvolvimentos na área de redes neurais artificiais fornecem uma abordagem alternativa na modelagem da dinâmica veicular, particularmente próxima aos seus limites operacionais onde o sistema é altamente não linear. O objetivo deste trabalho é investigar a habilidade de uma rede neural chamada de “neuro-veículo”, em simular as características dinâmicas de um modelo completo de uma sistema de suspensão veicular. A entrada do sistema é a velocidade e o ângulo de direção do veículo e as saídas são a aceleração lateral, a taxa de inclinação e as cargas laterais do mesmo. Para aumentar a performance da rede neural, os estados atuais e passados (atrasados) do neuro-veículo são retropropagados na rede, para o treinamento de seus pesos sinápticos. Utiliza-se o *backpropagation*, com aprendizado adaptativo e coeficiente de momento, para o treinamento da rede. Os padrões usados no treinamento supervisionado foram obtidos utilizando um modelo não

linear de um veículo. Toda a precisão dos resultados na identificação das características dinâmicas não lineares de um veículo foi demonstrada e analisada.

No trabalho de Moran e Nagai, 1997, o desempenho de redes neurais para a identificação e controle ótimo de suspensões pneumáticas ativas de veículos automotivos, foi analisado. Para isto é demonstrado que redes neurais podem ser que eficientemente treinadas para identificar a dinâmica de suspensões pneumáticas não lineares, e também após o devido treinamento, podem trabalhar como controladores ótimos. O desempenho da suspensão não linear com o neuro-controlador é comparado com o desempenho do controlador tipo LQ, projetado depois da linearização dos componentes da suspensão ao redor do ponto de equilíbrio. Este trabalho propõe para sistemas não lineares, um método de controle ótimo para melhorar o conforto dos passageiros, aplicado a suspensões ativas pneumáticas. Um modelo de um-grau-de-liberdade do sistema de suspensão experimental foi construído para examinar a efetividade dos métodos propostos. Os dados experimentais foram analisados e a resposta do sistema quando submetido a determinados tipos de excitação, permitiu verificar a precisão das redes neurais quando empregadas na identificação e controle de sistemas dinâmicos não lineares.

Kasparian e Batur, 1998, destaca que nas últimas três décadas a teoria de sistemas lineares teve contribuições significantes nos desenvolvimentos na área de controles clássicos. A motivação deste trabalho surge da necessidade de se desenvolver estratégias de controle mais modernas, que podem ser aplicadas a sistemas dinâmicos não lineares. Além disso, a necessidade de um método adaptativo de controle, surge para tratar problemas que envolvam sistemas em tempo variável. Neste trabalho uma rede neural treinada para representar a dinâmica de um modelo de referência pode ser usada para controle adaptativo de processos lineares e não lineares. O controlador neural proposto é testado em várias simulações de sistemas não lineares.

Stroeve, 1998, tem como objetivo principal do seu trabalho indicar como o processo de aprendizagem de um sistema de controle neural com o *backpropagation* através do tempo (BPTT) é influenciado pela estrutura do sistema de controle, pela dinâmica da planta e pelos parâmetros do processo de aprendizagem. Como a análise de um sistema linear é mais tratável que a análise de um sistema não linear, utilizou-se os primeiros para aprender técnicas de controle, procurando em seguida aplicá-los no controle de sistemas não lineares. Um modelo de

aprendizado direto foi usado e demonstrou uma boa precisão na identificação e controle de tais sistemas. Uma rede neural feedforward é usada como controlador e seus pesos são atualizados pela minimização de uma dada função de custo.

Poznyak, et al., 1998, resume uma pesquisa relacionada com o controle neural de sistemas dinâmicos. Em particular, é realizada a identificação de sistemas não lineares, como também um estudo da estabilidade de redes neurais dinâmicas é apresentado. A principal ferramenta de análise utilizada é a aproximação de **Lyapunov**. Foi ilustrada a aplicabilidade dos resultados por meio de exemplos. Ficou constatado que nem o sistema precisa ser estável, nem o tempo deve ser limitado a um intervalo fechado. Isto é um grande avanço em relação aos resultados existentes, que exigem a satisfação destas condições.

Em Mitsui et al., 1998, pode-se verificar a aplicação de redes neurais na identificação de aeronaves. Uma aproximação para a identificação das cargas que agem em aeronaves é feita através do uso de uma rede neural artificial para modelar a relação de carga-tensão em uma análise estrutural. Como o primeiro passo do estudo, este trabalho descreve a aplicação de uma rede neural artificial para identificar as cargas distribuídas por uma viga engastada. As cargas distribuídas são aproximadas por um conjunto de cargas concentradas. Neste trabalho foi demonstrado que usando uma rede neural artificial para identificar cargas é possível alcançar um grau de precisão elevado no processo de identificação de tais cargas.

Em Zhang et al., 1999, o controle adaptativo de sistemas não lineares baseado na utilização de redes neurais multicamadas (**MNN's**) foi apresentado para uma ampla classe de sistemas dinâmicos. O método proposto evita a possibilidade de problemas de singularidade do controlador usualmente projetados através da metodologia de controle adaptativo. O erro de ajuste do sistema é induzido a convergir para uma pequena vizinhança de zero, enquanto a estabilidade do sistema a malha fechada é garantida. A performance em regime transiente do sistema adaptativo resultante é analiticamente quantificada, e um exemplo é testado para demonstrar a eficiência do método.

Watanabe e Sharp, 1999, apresentam um trabalho que contém os tratamentos teóricos da aplicação de controladores neurais em várias suspensões ativas simplificadas. Em cada caso

considerado, o sistema controlado tem um objetivo definido a ser alcançado, a minimização de uma função de custo, sendo que as redes neurais são aplicadas em uma estrutura de aprendizado, tal que sistematicamente a performance do sistema é otimizada, via repetidas tentativas e modificações de parâmetros (os pesos sinápticos), até que o comportamento dinâmico da suspensão alcance os resultados esperados. O processo de aprendizado mostra-se eficiente e em alguns casos, conduzindo o sistema a resultados satisfatórios, tanto para sistemas lineares e não lineares. O objetivo final deste trabalho é o controle efetivo de uma variável geométrica do sistema de uma suspensão ativa veicular, ação esta que é essencialmente não linear. O modelo matemático de $\frac{1}{4}$ de veículo foi usado para uma análise comparativa do desempenho das suspensões passivas e ativas, através de simulações computacionais.

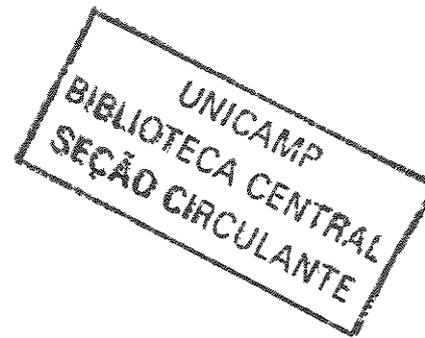
Pham et al., 1999, investiga uma aproximação da dinâmica inversa de plantas desconhecidas que usam um novo tipo de rede neural de retropropagação periódica. A rede tem dois elementos de contribuição durante a modelagem de uma planta de saída única, um representa a saída da planta e o outro, contribuição do erro para compensar as incertezas da modelagem. O ponto principal da aproximação proposta é fazer uso do método de aprendizagem direto inverso, para realizar a identificação do sistema, mesmo na presença de ruídos. Esta aproximação pode ser estendida facilmente à área de controle adaptativo on-line, cujos conceitos são discutidos brevemente. São utilizados resultados de simulações para ilustrar a precisão do método quando aplicado no controle de sistemas invariantes no tempo com elevados graus de não linearidades, demonstrando uma boa precisão nas respostas obtidas. Devido à natureza simples da rede neural e a arquitetura de treinamento on-line adotada, o método proposto possui as vantagens da aprendizagem direta inversa, sem sofrer as desvantagens de outras técnicas de aprendizagem inversa existentes.

Martins e Coelho, 2000, apresentam uma nova metodologia de controle baseada no controle proporcional, integral, e derivado (**PID**), conjugado com redes neurais artificiais *feedforward* (**FANN's**). As **FANN's** são usadas como modelos que predizem as variáveis controladas num tempo futuro. Estas informações são transferidas aos controladores **PID**, através do reajuste do “*set-point*” pré-estabelecido. A metodologia proposta foi testada em sistemas de primeira e segunda ordem usando um controlador **PI**, e as **FANN's**, cujos resultados foram considerados

excelentes para sistemas não lineares. Os resultados obtidos sugerem esta metodologia como sendo uma ferramenta complementar para os algoritmos de controle **PID**.

Krishnapura e Jutan, 2000, neste trabalho uma nova abordagem através da utilização de redes neurais artificiais para o controle de sistemas dinâmicos não lineares é empregada utilizando um modelo explícito do sistema. Redes neurais tradicionais normalmente não são consideradas para aplicações práticas em ambientes adaptativos, por causa do grande número de pesos associado a elas. Na estrutura proposta, influenciada pelo controlador **PID** clássico, o controlador neural possui muito poucos pesos de conexão e conseqüentemente é adequado para implementação em tempo real. Seu bom desempenho foi demonstrado através de simulações e testes práticos.

No trabalho de Safak e Turkay, 2000, uma rede neural multicamadas é treinada através do algoritmo *Levenberg Marquardt (LM)* é implementada para identificar a dinâmica de um sistema não linear. Um sinal de multi-freqüência “*flat-spectro*” é usado como sinal de excitação. São avaliados quantitativamente os efeitos da presença de ruído branco no desempenho da identificação de tais sistemas. A simulação e resultados experimentais confirmam que aquela identificação através da rede neural é afetada pela presença do ruído, mas é capaz de aprender, razoavelmente bem, o comportamento dinâmico do sistema. O método de treinamento para a identificação do sistema segue a abordagem de treinamento *off-line*.



Capítulo 3

Suspensões Automotivas: Passivas e Ativas

3.1 - Introdução

Neste capítulo é feito um estudo comparativo sobre os principais tipos de suspensões automotivas, destacando os sistemas passivos e ativos. Foi feita uma introdução ao funcionamento, aplicação, vantagens e desvantagens de cada sistema, através da análise de seus principais componentes, montagem e características, que são apresentadas em relação aos critérios de projeto dos sistemas de suspensão, o conforto e a segurança dos passageiros.

Foram deduzidas as equações de movimento, tanto para o modelo passivo de suspensão como para o modelo ativo, de um sistema de dois graus de liberdade, representado por um modelo $\frac{1}{4}$ de veículo, através de uma análise dinâmica, representando tais sistemas através da abordagem do espaço de estados, para se obter as respostas a diferentes tipos de excitações que caracterizam as suas condições de operação.

3.2 – Sistema de Suspensão

O sistema de suspensão de um automóvel tem como principais objetivos isolar as vibrações oriundas da via e garantir a estabilidade e a dirigibilidade do mesmo. Os seus ocupantes devem ficar isolados das acelerações geradas pelas irregularidades das vias por onde trafegam. Estas irregularidades surgem como resultado do contato pneu-via e adquirem valores de pico significativos no caso de estradas mal projetadas, construídas com materiais inadequados, de qualidade duvidosa e falta de manutenção. Em relação à segurança do veículo, o sistema de suspensão deve impedir que os pneus percam o contato com o solo, evitando assim derrapagens e perda de direção. Estes dois conceitos, segurança e conforto, devem sempre acompanhar o projeto de suspensões automotivas.

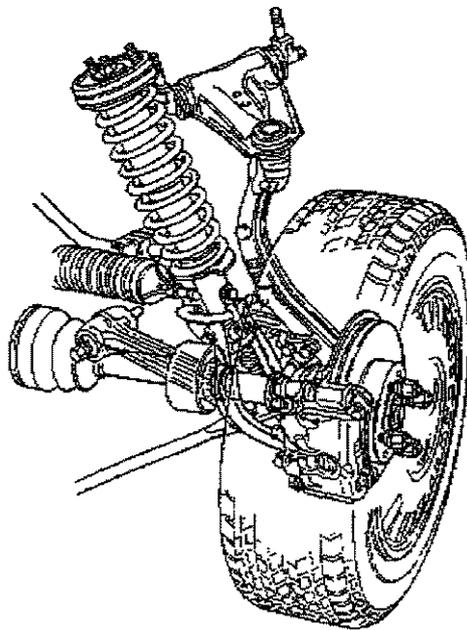


Figura 3.1 – Sistema de suspensão (Modelo de $\frac{1}{4}$ de veículo).

Os primeiros fabricantes de automóveis começaram a procurar soluções para problemas de direção e conforto dos passageiros. Os pneus dianteiros foram ligados ao eixo utilizando a ponta de eixo dianteira e pino mestre, o que permitiu que as rodas girassem, sendo que o eixo permanecia imóvel. As oscilações para cima e para baixo das molas foram diminuídas com uma estrutura chamada amortecedor. Estes primeiros amortecedores eram simplesmente dois braços

conectados por um parafuso, com um disco de fricção entre eles. A resistência era ajustada apertando ou soltando-se o parafuso. Com o passar dos anos, as suspensões passaram a ter um “*design*” mais sofisticado. Mudou o conceito dos componentes, novos materiais passaram a ser empregados, componentes eletrônicos foram incorporados, porém os objetivos básicos permaneceram os mesmos (Monroe, 2000).

O problema é que essas duas funções requerem características diferentes: rodar sobre pisos irregulares exige uma suspensão mais mole para absorver vibrações e impactos quando em baixas velocidades e mais rígidas em altas velocidades afim de garantir a estabilidade. Seus componentes são projetados, construídos e montados para atender estas exigências, levando em consideração também fatores como, custo e manutenção. A suspensão pode ser vista como um filtro, onde ao mesmo tempo que deve eliminar os efeitos negativos das vibrações causadas pela rugosidade do terreno, deve também deixar passar os componentes de frequência baixa que são necessários para seguir as ondulações das vias. Existem muitas maneiras analíticas de se descrever as entradas impostas à suspensão pelas estradas, as quais podem ser classificadas como “*shocks*” e vibrações. “*Shocks*” são eventos discretos de duração relativamente curta e alta intensidade, como causada por um buraco de grandes proporções, por exemplo. Já as vibrações, são caracterizadas por prolongadas e consistentes excitações (Hrovat, 1997).

Diferentes sistemas de suspensões satisfazem os critérios de segurança e conforto em diferentes graus. Podemos classificar as suspensões numa ordem ascendente de performance desejada em: suspensões passivas, semi-ativas e totalmente ativas. Suspensões passivas são aquelas encontradas nos veículos convencionais. Caracterizam-se pela ausência de fontes externas de energia, ao contrário das suspensões ativas que requerem fontes adicionais de energia, tais como compressores, bombas e atuadores. Descreveremos com maior detalhe os principais componentes e as características destes dois tipos de suspensão nos próximos itens deste capítulo.

As suspensões semi-ativas (ou semi-passivas), possuem características dos dois outros tipos de suspensões, o que oferece uma melhoria significativa na performance enquanto requer um consumo de energia relativamente pequeno. Apresentam também baixo custo de instalação e de manutenção quando comparadas com a suspensões ativas. Nas suspensões semi-ativas, a ação de

controle é ajustar o fator de amortecimento. Esta ação é realizada sobre um componente (amortecedor), que por sua vez, age sobre o sistema que está sendo controlado (Picado, 1998). Na suspensão passiva o fator de amortecimento é constante, enquanto nas suspensões semi-ativas podemos variá-lo para ter-se um conforto maior em toda faixa de frequência.

Os sistemas ativos tem demonstrado vantagens no controle de vibrações em relação aos sistemas passivos, mas o aumento nos custos e sua complexidade só justificariam o seu uso em casos nos quais a performance é crítica. Sistemas semi-ativos usando elementos de dissipação de energia modulados como geradores de força podem fornecer uma grande parte dos ganhos de performance dos sistemas ativos. Os sistemas semi-ativos são sempre não lineares mas simulações tem mostrado que freqüentemente é possível o projeto de tais sistemas usando leis de controle lineares. Por exemplo, para o algoritmo de controle da força de amortecimento, a lei de controle *skyhook* e suas variações são muito utilizadas (Karnopp, 1974) (Hwang, 1998).

3.3 – Suspensões Passivas

Nas suspensões passivas nenhuma força de controle externa é requerida, a vibração é controlada apenas, armazenando ou dissipando a energia associada ao movimento através de seus principais componentes como molas, amortecedores e massas. As propriedades destes componentes mecânicos não podem ser modificadas em tempo real. São as suspensões mais utilizadas pela indústria automobilística atualmente. Apesar do grande avanço tecnológico alcançado no projeto e desenvolvimento de suspensões ativas e semi-ativas, as suspensões passivas demonstram ser ainda bastante competitivas, devido ao baixo custo e simplicidade de montagem e de manutenção.

Existem dois tipos de suspensões passivas disponíveis no mercado atualmente, as otimizadas e as reguláveis. Nestas últimas, é possível ao motorista ajustar a rigidez das molas ou o fator de amortecimento dos amortecedores através de um dispositivo de regulagem estática. Nas suspensões passivas otimizadas não é possível ajustar as características dos componentes,

pois elas são projetadas e montadas para atender uma condição pré determinada de operação (Picado, 1998).

Os principais componentes da suspensão passiva são as molas e os amortecedores. As molas suportam o peso do veículo e mantêm a distância do solo durante o percurso absorvendo os impactos da rua. As molas são o elo flexível que permitem ao corpo do veículo, rodar relativamente tranqüilo, enquanto os pneus e a suspensão acompanham os buracos e impactos do solo. São utilizadas atualmente quatro concepções principais: molas helicoidais, molas laminadas, barra de torção e molas a ar (Figura 3.2). A mola mais utilizada é a mola helicoidal. Nesta mola não ocorre atrito interno e, devido a isso, proporcionam um percurso mais suave. Já nas molas laminadas, há presença de atrito interno devido ao fato que este tipo de mola são feixes de lâminas de aço de diferentes comprimentos que deslizam umas sobre as outras, mantidas unidas por grampos. A barra de torção pode ser reta ou em forma de L. São montadas firmemente no chassis, com a outra extremidade fixa na suspensão. Durante o movimento, a barra de torção irá se torcer, exercendo assim a ação da mola. A mola a ar é outro tipo que está ficando mais popular em carros de passeio, caminhões leves e pesados. É um cilindro de borracha que contém ar comprimido. Um pistão, conectado ao braço de controle inferior, se move para cima e para baixo juntamente com o braço. Isto faz o ar comprimido agir como uma mola. Variações da carga podem ser suportadas mediante a regulagens na quantidade de ar através de válvulas localizadas no topo do cilindro de borracha (Monroe, 2000).

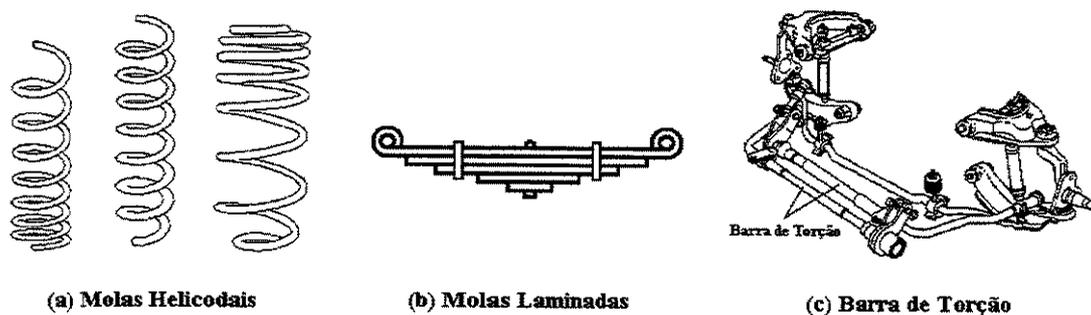


Figura 3.2 – Tipos de molas.

Os amortecedores são componentes importantes que controlam ou amortecem os movimentos excessivos da mola e da suspensão. Quando as rodas do veículo passam sobre um

obstáculo, elas podem oscilar para cima ou para baixo graças à ligação elástica que mantém com o chassi ou com a carroceria. Essa oscilação situa-se entre dois valores máximos fixados pelo projeto e delimitados pelos chamados “limitadores de curso”. O processo é este: molas excitadas, começam a oscilar à passagem sobre o obstáculo e repetem a oscilação por um certo tempo, mesmo após a superação do obstáculo pelo veículo. Sem a presença de amortecedores adequados, a absorção dessas oscilações ficaria confiada unicamente aos atritos da suspensão e à geração de calor na mola; como resultado, a roda perderia o contato com o terreno, contra o qual se chocaria com certa violência, ver figura 3.3. O amortecedor ideal deve transmitir o menos possível os desníveis do terreno à carroceria e manter a roda sempre em contato com o chão. Quando as irregularidades do solo apresentam-se com pouca frequência, isto é, são distantes entre si, é bom que o amortecimento seja elevado, de modo a evitar que a carroceria continue a oscilar depois de superar o obstáculo; se, ao contrário, as irregularidades são frequentes (devido à pouca distância entre si ou às altas velocidades), o amortecedor deve ser bastante flexível, para impedir que contínuas oscilações sejam transmitidas à carroceria. Com o aumento da velocidade do veículo, e portanto da frequência com que as irregularidades do piso se apresentam às rodas, crescem as forças de inércia aplicadas às massas não suspensas do veículo. Disso resulta o maior retardo com que as rodas conseguem adaptar-se às irregularidades da estrada; assim, para garantir eficiência, exige-se do amortecedor um progressivo enrijecimento com o aumento da velocidade.

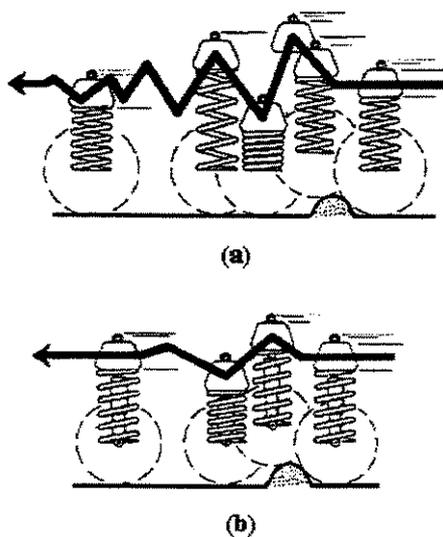


Figura 3.3 – Sistema de suspensão: (a) apenas com mola e (b) com a mola e o amortecedor.

Os amortecedores são basicamente bombas de óleo (Figura 3.4). Possuem um pistão que é ligado no final de uma haste e pressiona um fluido hidráulico dentro da câmara. A medida que a suspensão passa por buracos e lombadas, o líquido hidráulico é forçado a passar por pequenos orifícios dentro do pistão. Esta passagem é controlada, onde apenas uma pequena quantidade de fluido consegue fluir pelo pistão, desacelerando o mesmo, que por sua vez, irá desacelerar a mola e a suspensão. A força de resistência desenvolvida por um amortecedor é função de parâmetros como velocidade da suspensão, número dos orifícios do pistão e o tamanho dos mesmos. Amortecedores são mecanismos hidráulicos sensíveis à variações de velocidade, isto é, quanto mais rápido a suspensão se mover, maior será a resistência oferecida pelo amortecedor. Podemos citar alguns tipos, tais como: amortecedores de fricção, amortecedores hidráulicos de êmbolo rotativo, amortecedores hidráulicos de alavanca, amortecedores hidráulicos incorporados, amortecedores hidráulicos telescópicos, etc.

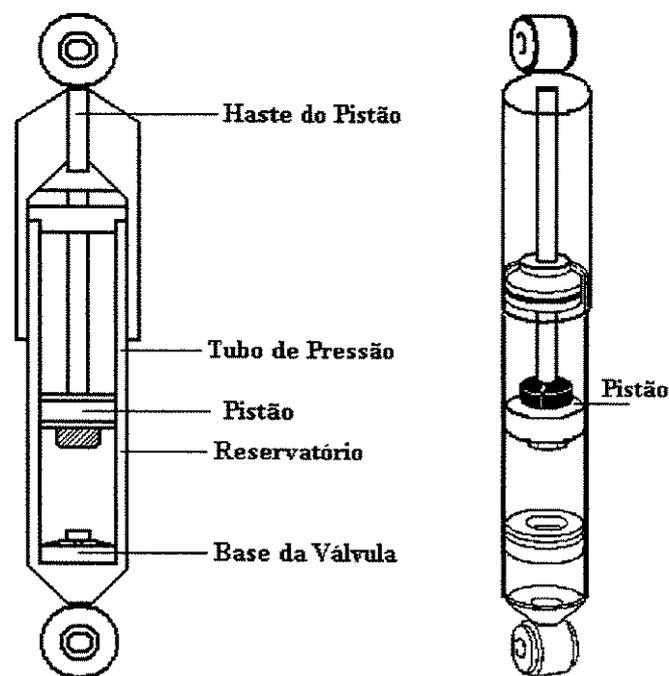


Figura 3.4 – Amortecedor.

A geometria dos componentes, a escolha adequada dos materiais utilizados, a regulagem dos amortecedores e as dimensões das molas são fatores importantes para o perfeito

funcionamento do sistema de suspensão passiva. Estes fatores que, junto com outros (rigidez, relação altura/largura ou diâmetro do pneu, ou espaço de trabalho da suspensão) determinam a resposta da suspensão às excitações. O que caracteriza a qualidade de uma suspensão passiva é o processo de otimização dos seus componentes e das dimensões da suspensão. Esta otimização atende a condições pré-determinadas pelo funcionamento do sistema de suspensão, o que pode explicar as diferenças entre características dinâmicas e estáticas para suspensões passivas de diferentes tipos de carro (carros esportivos, veículos *off-road*, carros de passeio e carros de alto luxo).

Os componentes das suspensões passivas apresentam características altamente não lineares que são aproveitadas pelas fábricas para atender às exigências de operação através de um processo empírico de otimização. Muitas vezes, as características não lineares são desconhecidas (analiticamente), pois estes estão interessados nas respostas do sistema de suspensão e não na lei matemática que representa os fenômenos de resposta dos componentes às excitações externas. Em um amortecedor comum, a resposta dinâmica é caracterizada por diversos fatores como o diâmetro de um furo no cabeçote, o comprimento do duto de óleo, a constante de rigidez da mola do cabeçote do amortecedor, o tipo de óleo, o diâmetro do pistão, dentre muitos outros. Já em uma mola helicoidal se o material de que é feita a mola, ou o diâmetro do helicóide, ou o número de voltas do helicóide ou ainda se o passo do helicóide forem modificados, as características da mola também se modificam. É bastante difícil de se obter um modelo teórico que quantifique a influência dos fatores citados acima, ainda que para as indústrias fabricantes de componentes automotivos estas informações sejam de pouco valor (Picado, 1998).

Se comparadas com os outros tipos de suspensões automotivas, as suspensões passivas otimizadas são os sistemas de suspensão mais simples. Porém, o processo de otimização de uma suspensão passiva é extremamente complexo e passa por diversas etapas de simplificações e linearizações para que se torne um problema razoavelmente tratável. As linearizações são justificáveis, uma vez que o procedimento de projeto envolve erros de ordem superior aos das linearizações adotadas, pois os critérios de otimização são bastante subjetivos. Entretanto, é necessária cautela para assumir as hipóteses simplificadoras, pois estas podem levar a resultados inválidos se o projetista não souber exatamente as implicações de cada hipótese assumida na resposta do modelo simulado.

Para otimizar os componentes da suspensão os fabricantes cruzam as respostas dos sistemas de suspensão que são ajustados empiricamente, com a experiência de uma equipe de pilotos de teste e de técnicos, que modificam as especificações do amortecedor, da mola ou de outros componentes da suspensão, até que a resposta dinâmica do sistema de suspensão seja satisfatória. Esta otimização atende a uma condição inicial arbitrária, que deve ser definida pelo projetista "a priori". Dessa forma, cada fabricante apresenta seus componentes com propriedades distintas, ou seja, preparados para condições diferentes de uso. Os amortecedores e as molas existentes no mercado atual são otimizadas e as suas características de não linearidade são exaustivamente utilizadas para atender a uma gama cada vez maior de condições ideais de funcionamento. Este fato faz com que as suspensões passivas mantenham a preferência dos fabricantes de automóveis em série, pois apresentam bom desempenho a baixos custos de instalação e de manutenção (Picado, 1998).

3.3.1 – Modelagem Matemática:

Usaremos para a análise matemática da suspensão passiva de $\frac{1}{4}$ de veículo com dois graus de liberdade o modelo proposto na figura 3.5 (Karnopp, 1974) (Nagai, 1993).

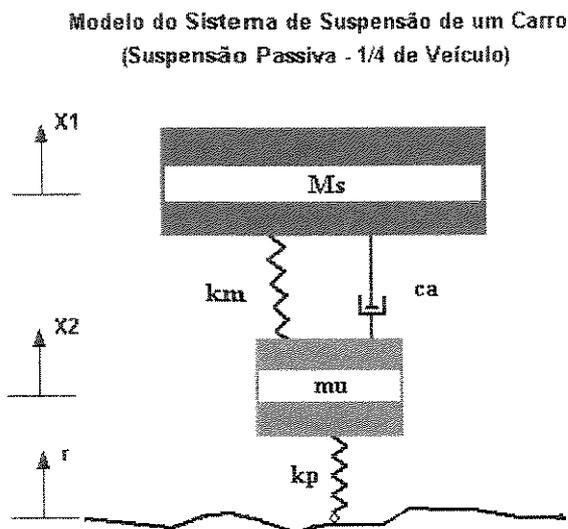


Figura 3.5 – Diagrama do sistema passivo

Neste modelo apenas o movimento vertical da suspensão é permitido e suas variáveis estão listadas na tabela 1. A variável r representa as irregularidades do terreno ou seja, as excitações que serão impostas à suspensão a fim de obtermos as respostas para posterior análise. A “massa suspensa” pode ser definida como o peso da carroceria, motor, chassi e ocupantes. A “massa do sistema eixo-roda”, inclui o peso do sistema de freios, rodas, bandejas, direção e eixo. O pneu foi representado por uma mola de rigidez k_p . Salienta-se que estes pesos se referem à $\frac{1}{4}$ de um veículo de passeio e que seus componentes são lineares.

Variável	Unidades	Descrição
M_s	kg	Massa suspensa (¼ de veículo)
k_m	N/cm	Constante de rigidez da mola da suspensão
c_a	Ns/cm	Fator de amortecimento do amortecedor da suspensão
μ	kg	Massa do sistema eixo-roda (¼ de veículo)
k_p	N/cm	Constante de rigidez do pneu
X_1	cm	Deslocamento vertical da massa suspensa
X_2	cm	Deslocamento vertical da massa do sistema eixo-roda

Tabela 3.1 – Variáveis utilizadas no modelo passivo.

O modelo da figura 3.5 pode ser descrito a partir das equações diferenciais 3.1 e 3.2:

$$M_s \ddot{X}_1 + c_a (\dot{X}_1 - \dot{X}_2) + k_m (X_1 - X_2) = 0 \quad (3.1)$$

$$\mu \ddot{X}_2 - c_a (\dot{X}_1 - \dot{X}_2) - k_m (X_1 - X_2) + k_p X_2 = k_p r \quad (3.2)$$

Para análise e simulação deste sistema passivo, as suas equações de movimento são representadas na forma de estado, ou seja o sistema de segunda ordem é reduzido a um sistema de primeira. Para isso, são definidas como novas variáveis, chamadas de variáveis de estado, as coordenadas do modelo e suas derivadas de primeira ordem, ou seja, as variáveis de posição e velocidade do sistema. A equação de estado em tempo contínuo é então dada por:

$$\dot{z}(t) = A * z(t) + L * w(t) \quad (3.3)$$

onde as matrizes A (matriz do sistema ou matriz de estado) e L (matriz que multiplica o vetor de excitação), o vetor de excitação $w(t)$ e o vetor de estado $z(t)$ são dados por:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_m/M_s & k_m/M_s & -c_a/M_s & c_a/M_s \\ k_m/\mu & -(k_m+k_p)/\mu & c_a/\mu & -c_a/\mu \end{bmatrix} \quad (3.4)$$

$$\mathbf{L} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ kp/mu \end{bmatrix} \quad (3.5)$$

$$\mathbf{z}(t) = \begin{bmatrix} X_1 \\ X_2 \\ \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} \quad (3.6)$$

$$\mathbf{w}(t) = r(t) \quad (3.7)$$

A equação de saída é dada por:

$$\mathbf{y}(t) = \mathbf{C} * \mathbf{z}(t) \quad (3.8)$$

onde a matriz de saída \mathbf{C} é:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

Devido ao grande avanço na tecnologia dos computadores digitais, atualmente, estão disponíveis computadores de alta capacidade de processamento em pequenas pastilhas de silício. Este fato faz com que os sistemas de controle atuais sejam desenvolvidos para trabalhar em acordo com os princípios de tais processadores. A característica básica dos controladores digitais é a de que os valores numéricos sejam tratados como pontos flutuantes, ou seja, são valores discretos. A variável tempo nestes sistemas deve ser discreta e este fato causa algumas mudanças no equacionamento do sistema de controle.

A descrição do sistema digital baseia-se na representação na forma de equações de diferenças, que quando agrupadas na forma de uma equação de matrizes e vetores de primeira ordem são chamadas de método de espaço de estados para sistemas digitais. Este método é semelhante ao método de espaço de estados para sistemas de domínio contínuo do tempo descrito anteriormente, porém algumas modificações devem ser feitas nas equações, para que o sistema represente a resposta discreta correta. A escolha da taxa de amostragem define a estabilidade do sistema para determinada condição de trabalho. As definições dos termos para o sistema no domínio contínuo do tempo também valem para a abordagem discreta (Picado, 1998).

Para sistemas lineares discretos e invariantes no tempo as equações de estado e de saída podem ser escritas como na equação 3.10.

$$\begin{aligned} \mathbf{z}(k+1) &= \mathbf{A}_d * \mathbf{z}(k) + \mathbf{L}_d * \mathbf{w}(k) \\ \mathbf{y}(k+1) &= \mathbf{C} * \mathbf{z}(k) \end{aligned} \quad (3.10)$$

A equação 3.10 tem forma semelhante à da equação 3.3, sendo que as matrizes \mathbf{A}_d e \mathbf{L}_d são as formas discretizadas das matrizes \mathbf{A} e \mathbf{L} . A forma discreta pode ser obtida através da resolução da equação de estado contínua, ou seja, a equação de estado em tempo discreto é, essencialmente um integrador discreto explícito (Picado, 1998). Para discretizar um sistema contínuo, devemos resolver a equação de estados em tempo contínuo (equação 3.3) que pode ser reescrita na forma,

$$\dot{\mathbf{z}}(t) - \mathbf{A} * \mathbf{z}(t) = \mathbf{L} * \mathbf{w}(t) \quad (3.11)$$

Multiplicando ambos os lados por $e^{-\mathbf{A}t}$, temos

$$e^{-\mathbf{A}t} [\dot{\mathbf{z}}(t) - \mathbf{A} \mathbf{z}(t)] = \frac{d}{dt} [e^{-\mathbf{A}t} \mathbf{z}(t)] = e^{-\mathbf{A}t} \mathbf{L} \mathbf{w}(t) \quad (3.12)$$

Integrando de 0 a t, temos

$$e^{-\mathbf{A}t} \mathbf{z}(t) = \mathbf{z}(0) + \int_0^t e^{-\mathbf{A}\tau} \mathbf{L} \mathbf{w}(\tau) d\tau \quad (3.13)$$

ou,

$$\mathbf{z}(t) = e^{\mathbf{A}t} \mathbf{z}(0) + \int_0^t e^{\mathbf{A}(t-\tau)} \mathbf{L} \mathbf{w}(\tau) d\tau \quad (3.14)$$

Esta equação descreve a variação com o tempo da variável de estado \mathbf{z} com respeito à condição inicial $\mathbf{z}(0)$ e a entrada $\mathbf{w}(t)$. O cálculo de $\mathbf{z}(t)$ em intervalos igualmente espaçados de tempo t pode ser obtido por uma representação de tempo discreto dado pela equação 3.10. Temos intervalos de tempo igualmente espaçados dados por $0, \Delta t, 2\Delta t, \dots, (k+1)\Delta t, \dots$ onde Δt é um intervalo constante chamado “intervalo de amostragem” (Juang, 1994).

Assumindo que a entrada $\mathbf{w}(t)$ é amostrada e submetida a um integrador de ordem zero, de forma que o seu valor permanece constante entre dois momentos de amostragem, ou seja, ao longo de um intervalo de amostragem, temos:

$$\mathbf{w}(t) = \mathbf{w}(k\Delta t), \text{ para } k\Delta t < t \leq (k+1)\Delta t \quad (3.15)$$

Substituindo t por $k\Delta t$ na equação 3.14,

$$\mathbf{z}(k\Delta t) = e^{\mathbf{A}k\Delta t} \mathbf{z}(0) + e^{\mathbf{A}k\Delta t} \int_0^{k\Delta t} e^{-\mathbf{A}\tau} \mathbf{L} \mathbf{w}(\tau) d\tau \quad (3.16)$$

Substituindo t por $(k+1)\Delta t$,

$$\mathbf{z}((k+1)\Delta t) = e^{\mathbf{A}(k+1)\Delta t} \mathbf{z}(0) + e^{\mathbf{A}(k+1)\Delta t} \int_0^{(k+1)\Delta t} e^{-\mathbf{A}\tau} \mathbf{L} \mathbf{w}(\tau) d\tau \quad (3.17)$$

Multiplicando a equação 3.16 por $e^{\mathbf{A}\Delta t}$ e subtraindo-a da equação 3.17,

$$\mathbf{z}((k+1)\Delta t) = e^{\mathbf{A}\Delta t}\mathbf{z}(k\Delta t) + e^{\mathbf{A}(k+1)\Delta t} \int_{k\Delta t}^{(k+1)\Delta t} e^{-\mathbf{A}\tau} \mathbf{L}\mathbf{w}(\tau) d\tau \quad (3.18)$$

Da equação 3.15, o valor de $\mathbf{w}(\tau)$ na integral do segundo membro pode ser substituído por $\mathbf{w}(k\Delta t)$ para que a equação 3.18 resulte em,

$$\mathbf{z}((k+1)\Delta t) = e^{\mathbf{A}\Delta t}\mathbf{z}(k\Delta t) + \int_0^{\Delta t} e^{\mathbf{A}(\Delta t-t)} \mathbf{L}\mathbf{w}(k\Delta t) dt \quad (3.19)$$

Podemos definir agora as seguintes expressões para ser usadas na equação 3.10,

$$\begin{aligned} \mathbf{A}_d &= e^{\mathbf{A}\Delta t} \\ \mathbf{L}_d &= \left(\int_0^{\Delta t} e^{\mathbf{A}(\Delta t-t)} dt \right) * \mathbf{L} \end{aligned} \quad (3.20)$$

As matrizes em tempo discreto \mathbf{A}_d e \mathbf{L}_d na equação 3.20 podem ser calculadas através de expansões em series:

$$\begin{aligned} \mathbf{A}_d &= e^{\mathbf{A}\Delta t} = \mathbf{I} + \mathbf{A}\Delta t + \frac{1}{2!}(\mathbf{A}\Delta t)^2 + \frac{1}{3!}(\mathbf{A}\Delta t)^3 + \dots \\ \mathbf{L}_d &= \left(\int_0^{\Delta t} e^{\mathbf{A}(\Delta t-t)} dt \right) * \mathbf{L} = \mathbf{B} \left[\mathbf{I}\Delta t + \frac{1}{2!}\mathbf{A}(\Delta t)^2 + \frac{1}{3!}\mathbf{A}^2(\Delta t)^3 + \dots \right] \end{aligned} \quad (3.21)$$

Estas series convergem se a matriz de estado \mathbf{A} é assintoticamente estável no sentido que a parte real de todos os autovalores são negativos. Se nenhum dos autovalores de \mathbf{A} são nulos, a matriz \mathbf{L}_d da equação 3.20 pode ser calculada por:

$$\mathbf{L}_d = \mathbf{A}^{-1}(\mathbf{A}_d - \mathbf{I}) * \mathbf{L} \quad (3.22)$$

3.4 – Suspensões Ativas

As pesquisas recentes revelam que métodos passivos são incapazes de reduzir a vibração da suspensão através de todo intervalo de frequências, e que métodos de controle ativos são, em contrapartida, muito eficientes para impor um isolamento adequado. Como foi descrito anteriormente as suspensões dos automóveis convencionais, em suas configurações mais básicas, consistem de um elemento que armazena energia - normalmente uma mola - e um elemento que dissipa energia - normalmente um amortecedor ou um absorvedor de “*shocks*”. Estes dois elementos são passivos por natureza, em outras palavras, eles não podem introduzir energia no sistema - as forças que eles geram são uma função do deslocamento relativo entre o chassi do veículo e cada um dos eixos das rodas. As taxas segundo às quais é armazenada energia temporariamente pela mola, ou dissipada pelo amortecedor, são geralmente fixas. Eles são escolhidos tal que a suspensão garanta uma boa performance em relação ao conforto e a segurança dos passageiros.

Resolver o conflito inerente entre apoiar o chassi do veículo efetivamente, enquanto isola-o das indesejadas vibrações produzidas pelas vias, garantido assim o conforto dos passageiros, enquanto mantém ao mesmo tempo, o contato suficiente entre os pneus e a estrada, para fornecer a tração necessária para as manobras e a força para acelerar e frear, e manter as dimensões físicas do sistema de suspensão dentro de limites razoáveis, representa o principal desafio no projeto da suspensão de um automóvel.

As suspensões ativas oferecem uma solução tecnologicamente avançada a este desafio. Estes sistemas são apresentados como uma resposta à necessidade de se desenvolver veículos de passeio mais seguros, que são capazes de combinar níveis altos de conforto, dirigibilidade e estabilidade. Ao contrário da suspensão passiva que só pode armazenar ou dissipar energia segundo taxas fixas, as suspensões ativas podem modular o fluxo e prover de energia o sistema. Além disso, as características dinâmicas de uma suspensão ativa podem ser ajustadas em tempo real, continuamente, em resposta para condições motrizes medidas por sensores montados no veículo.

Este tipo de suspensão é caracterizado por um atuador embutido que pode gerar forças de controle (calculadas por um computador), suprimindo assim os movimentos desagradáveis impostos pelas estradas. Tais atuadores podem ser hidráulicos ou hidropneumáticos, controlados por servo válvulas eletro-hidráulicas de ação rápida. A energia para o atuador normalmente deve ser fornecida por uma bomba hidráulica.

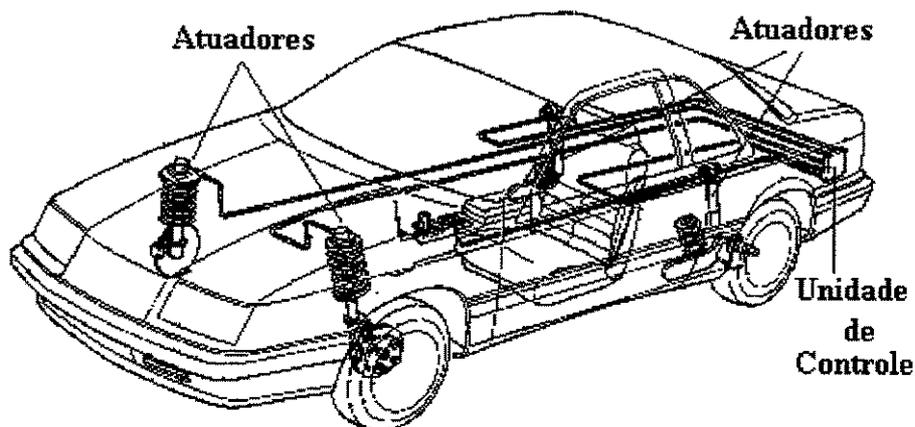


Figura 3.6 – Modelo de uma suspensão ativa.

Para “sistemas completamente ativos”, também conhecidos como sistemas de largura de banda alta ou simplesmente banda-larga, o atuador hidráulico substitui o amortecedor padrão da suspensão passiva. O atuador é posicionado entre o corpo do veículo e o eixo, tendo em geral uma mola em paralelo para equilibrar a carga estática. O objetivo destes sistemas é controlar a suspensão sobre todos os níveis de frequências, aos quais os passageiros são muito sensíveis, entre 1 Hz e 15 Hz. Em particular, um sistema deste tipo serve para melhorar a resposta da suspensão ao redor seus dois modos principais ressonantes: a “vibração da carroceria”, geralmente entre 1 Hz e 2 Hz e o “salto do pneu”, geralmente entre 10 Hz e 12 Hz. O critério de segurança foi melhorado porque o comportamento dinâmico da força de contato entre os pneus e estrada foi melhor controlado, devido à ausência da frequência de ressonância baixa do corpo de veículo (ver figura 3.7). Estes sistemas podem melhorar consideravelmente o conforto dos passageiros, pela habilidade de controlar as vibrações da massa suspensa (corpo de veículo) de um modo ativo. Estes sistemas ativos precisam de uma largura de banda alta porque as vibrações

da massa suspensa (originadas pelas irregularidades das estradas) tem que ser absorvidas ativamente. Alcançar uma largura de banda razoável não é fácil devido as limitações dos custos e quanto mais alta a largura de banda, haverá proporcionalmente um aumento do consumo de energia do sistema.

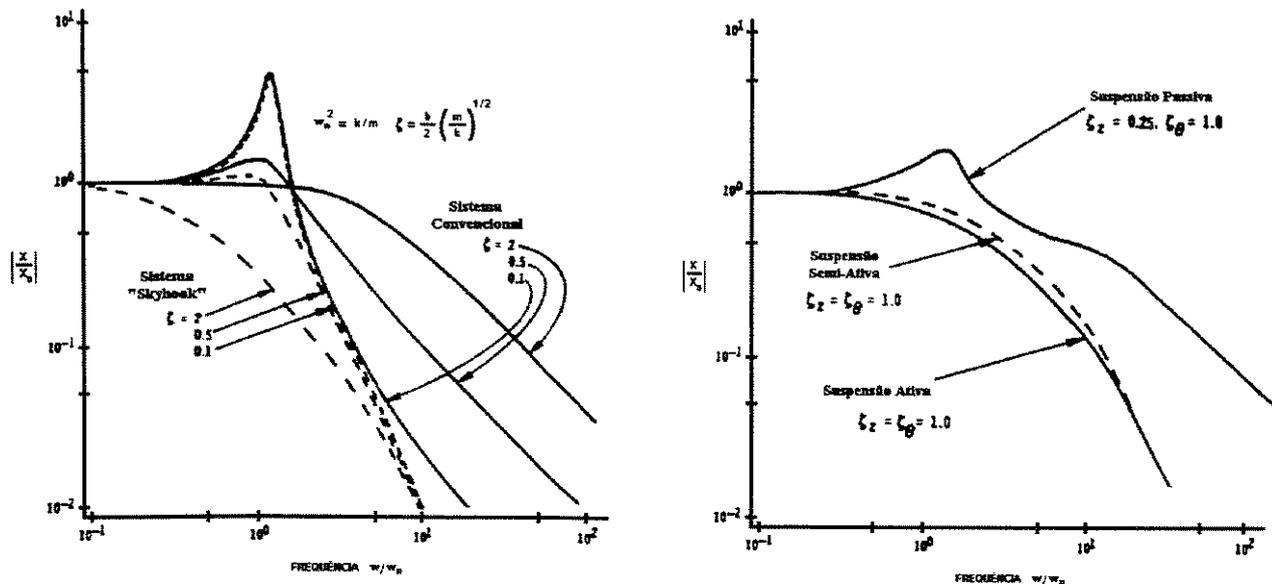


Figura 3.7 – Resposta em frequência do sistema de suspensão de $\frac{1}{2}$ veículo (Karnopp, 1974).

Para “sistemas ativos lentos”, também conhecido como sistemas de banda limitada ou de banda-estreita, o atuador normalmente é instalado em série com a mola ou com o amortecedor da suspensão passiva. O objetivo destes sistemas é controlar a suspensão até um valor de frequência limitado, aproximadamente 5 Hz, e em particular, melhorar o desempenho da suspensão ao redor da frequência de vibração da carroceria. O atuador efetivamente bloqueia as frequências mais altas, e conseqüentemente, o sistema atua da mesma maneira que uma suspensão passiva ao redor da frequência de “salto do pneu”.

Podemos também classificá-las de acordo com seus componentes em: suspensões hidráulicas, suspensões de ar e suspensões magnéticas.

As suspensão hidráulica, são aquelas onde o sistema ativo envolve a substituição dos elementos convencionais da suspensão por um atuador hidráulico, que é controlado através de servo válvulas de resposta de alta frequência via compensação por realimentação. Para realizar um bom isolamento de vibração, semelhante ao do amortecedor *skyhook*, a aceleração vertical do corpo do veículo é medida e passada ao microprocessador que calcula a força necessária do atuador. As suspensões ativas com atuadores hidráulicos também podem controlar a atitude do corpo do veículo, medindo-se a aceleração lateral. Com este controle total da atitude do corpo do veículo, podem ser suprimidos movimentos indesejados durante as manobras, garantindo assim uma boa dirigibilidade e estabilidade sobretudo em altas velocidades. Na aplicação prática, com o objetivo de diminuir o consumo de energia, uma câmara de ar auxiliar é presa ao atuador hidráulico, através da qual as vibração de alta-freqüência são absorvidas pelo efeito da “almofada de ar”.

Suspensões ativas à ar com atuadores pneumáticos tem sido aplicadas em automóveis e trens, para controlar a atitude de corpo de veículo e diminuir os movimentos laterais e de torção em curvas, conduzindo à uma boa estabilidade em altas velocidades. É esperado que as vibrações de alta frequência sejam absorvidas pela almofada de ar devido as propriedades de compressão do ar, garantindo assim uma boa qualidade de passeio. Porém, os atuadores pneumáticos atuam em uma faixa de frequência baixa se comparados aos atuadores hidráulicos. É necessário então, compensar a força pneumática considerando-se a demora do tempo de resposta do atuador e utilizando uma compensação direta, como também um *loop* de realimentação, medindo-se a aceleração vertical, a velocidade, o deslocamento e outras variáveis se necessário.

Sistemas de suspensão magnéticos são divididos em dois tipos: suspensão eletromagnética (**EMS**) que é o tipo de força atraente controlada, e suspensão electrodinâmica (**EDS**) que é o tipo de força repulsiva. Este tipo de suspensão é utilizado em trens-bala de alta velocidade e trens urbanos com baixo ruído, sendo encontrados atualmente no Japão e na Alemanha, por exemplo. Como é bem conhecido, a suspensão eletromagnética que usa eletroímãs é bastante instável, assim uma compensação por realimentação é indispensável. Além disso, como a suspensão tipo

EMS só permite aproximadamente 10 mm de espaço entre os imãs e os trilhos, é muito importante controlar o intervalo livre entre as superfícies, para evitar o contato entre os imãs e os trilhos. Este tipo de suspensão pode ser classificada como uma suspensão ativa. Por outro lado, suspensão eletrodinâmica sem controle é classificada como uma suspensão passiva. Embora este tipo gere uma força repulsiva forte para obter um intervalo de ar grande, tem alguns problemas do ponto de vista de isolamento de vibração, como, uma força de amortecimento baixa e uma relação pequena de massa entre o corpo de veículo e a massa da suspensão. Assim, são necessárias suspensões primárias ou secundárias ativamente controladas para se garantir uma alta estabilidade em regime transitório, “*handling*” e “*ride*” satisfatório, como mencionado anteriormente (Nagai, 1993).

Embora suspensões ativas provassem ser capazes de produzir notáveis melhorias de desempenho no conforto, especialmente nas frequências de movimento vertical do veículo, e reduções significantes no deslocamento do corpo do mesmo durante arrancadas, freadas e derrapagens; o peso total, custo alto dos componentes e a quantidade de energia consumido pelo sistema, este último fator é especialmente crítico para sistemas completamente ativos, tem limitado sua utilização nos carros fabricados em série.

O consumo elevado de energia deste sistemas ativos hidraulicamente controlados se origina no modo com que as forças de controle são geradas na suspensão. A maioria destas suspensões ativas tem um atuador construído em série com uma fonte de compressão (normalmente uma mola hidropneumática à gás). Para produzir a força de controle desejada o atuador tem que comprimir a mola causando assim o consumo de energia. Para obter um bom desempenho é necessário que o controle da largura de banda do atuador se estenda substancialmente além da frequência natural da vibração da roda (tipicamente 10 - 15 Hz). O problema de transmissão de ruído é resolvido quando se adiciona algum componente de maior flexibilidade em série com o atuador. Diferentes níveis de força requeridos pelo atuador, podem ser reduzidos aliviando o braço da carga estática da massa suspensa. Isto pode ser conseguido incorporando uma mola em paralelo com o atuador.

Suspensões ativas não só são capazes de fornecer uma melhor dirigibilidade, melhor estabilidade durante as manobras, como também garantem um maior conforto dos passageiros e

um uso mais efetivo do espaço de trabalho da suspensão. Em contrapartida ocorre um aumento significativo do consumo de energia no veículo, um aumento do peso e um problema complexo de alocação de componentes.

O uso das suspensões ativas foi restringido principalmente a aplicações de alta tecnologia, como nos carros de Fórmula 1 e certos protótipos de veículos onde o custo não é a preocupação principal. Nestas aplicações, vários sistemas de banda-larga foram implementados com sucesso e posteriormente testados. A maioria dos fabricantes de automóveis tem pesquisado seus próprios sistemas ativos, mas muito poucos dos esforços deles tem sido posto em prática na produção de veículos, devido principalmente a falta de um mercado que absorva esta tecnologia.

Nos últimos anos, o surgimento de materiais compostos de baixo peso, tecnologias alternativas de atuação, e avanços nos dispositivos eletrônicos e sensores de medidas; como também uma redução do custo computacional na implementação em tempo real do processamento de sinais e do controle, tem ampliado o interesse dos pesquisadores na área de suspensões ativas. O projeto de chassis integrados que consideram a suspensão, direção e o sistema de freios, como parte de um único componente interativo multifuncional, em vez de olhar cada um destes sistemas como partes independentes que estão conectadas juntas, está se mostrando como grande promessa, e resultará no surgimento de sistemas completamente ativos viáveis comercialmente em um futuro próximo.

3.4.1 – Modelagem Matemática:

Usaremos para a análise matemática da suspensão ativa de ¼ de veículo com dois graus de liberdade o modelo proposto na figura 3.8 (Moran, 1994) (Watanabe, 1999)

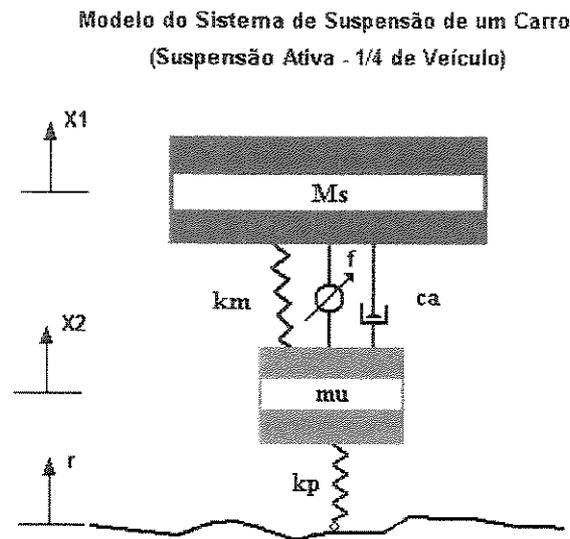


Figura 3.8 – Diagrama do sistema ativo

Neste modelo apenas o movimento vertical da suspensão é permitido e suas variáveis estão listadas na tabela 1, com a inclusão da variável f , representando a força em Newtons que o atuador exerce sobre a suspensão. Todos os seus componentes são lineares.

O modelo da figura 3.8 pode ser descrito a partir das equações diferenciais 3.23 e 3.24:

$$M_s \ddot{X}_1 + c_a (\dot{X}_1 - \dot{X}_2) + k_m (X_1 - X_2) + f = 0 \quad (3.23)$$

$$m_u \ddot{X}_2 - c_a (\dot{X}_1 - \dot{X}_2) - k_m (X_1 - X_2) - f + k_p X_2 = k_p r \quad (3.24)$$

Para análise e simulação deste sistema ativo, as suas equações de movimento são representadas na forma de estado. A análise é similar ao modelo da suspensão passiva. A equação de estado em tempo contínuo é então dada por:

$$\dot{\mathbf{z}}(t) = \mathbf{A} * \mathbf{z}(t) + \mathbf{B} * \mathbf{u}(t) + \mathbf{L} * \mathbf{w}(t) \quad (3.25)$$

onde as matrizes \mathbf{A} (equação 3.4) e \mathbf{L} (equação 3.5), os vetores $\mathbf{w}(t)$ (equação 3.7) e $\mathbf{z}(t)$ (equação 3.6) são os mesmos do modelo passivo. Teremos nesta análise a inclusão da matriz \mathbf{B} (matriz de controle) e do vetor $\mathbf{u}(t)$ (vetor de controle ou sinal de controle) que são dados por:

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ -1/Ms \\ 1/mu \end{bmatrix} \quad (3.26)$$

$$\mathbf{u}(t) = \mathbf{f}(t) \quad (3.27)$$

A equação de saída é dada pelo equação 3.8 e a matriz \mathbf{C} pela equação 3.9.

Para sistemas lineares discretos e invariantes no tempo as equações de estado e de saída podem ser escritas como na equação 3.28, seguindo a metodologia descrita anteriormente para as suspensões passivas.

$$\begin{aligned} \mathbf{z}(k+1) &= \mathbf{A}_d * \mathbf{z}(k) + \mathbf{B}_d * \mathbf{u}(t) + \mathbf{L}_d * \mathbf{w}(k) \\ \mathbf{y}(k+1) &= \mathbf{C} * \mathbf{z}(k) \end{aligned} \quad (3.28)$$

Na equação 3.28 as matrizes \mathbf{A}_d , \mathbf{B}_d e \mathbf{L}_d são as formas discretizadas das matrizes \mathbf{A} , \mathbf{B} e \mathbf{L} . Para discretizar um sistema ativo contínuo, vamos usar o mesmo procedimento desenvolvido anteriormente para as suspensões passivas. Conseguimos então a forma de estado em tempo discreto dada pela equação 3.28. Onde teremos (Truscott, 1995),

$$\begin{aligned}
\mathbf{A}_d &= \mathbf{e}^{\mathbf{A}\Delta t} \\
\mathbf{B}_d &= \mathbf{A}^{-1}(\mathbf{A}_d - \mathbf{I}) * \mathbf{B} \\
\mathbf{L}_d &= \mathbf{A}^{-1}(\mathbf{A}_d - \mathbf{I}) * \mathbf{L}
\end{aligned}
\tag{3.29}$$

3.4.2 – Controle ótimo para suspensões ativas:

Antes de propormos a aplicação de um controlador neural para a suspensão ativa, vamos analisar o modelo da suspensão ativa com um controlador ótimo linear quadrático ou **LQR**, que será usado posteriormente como modelo de referência para o controlador neural.

O sistema a ser controlado será definido pela equação 3.23. Ao se projetarem sistemas de controle, há freqüentemente o interesse de se escolher o vetor de controle $\mathbf{u}(\mathbf{t})$ tal que um dado índice de desempenho seja minimizado. Os controladores ativos com o **LQR** atuarão no sistema mediante uma lei de controle baseada na realimentação de estado, através de uma matriz de ganho.

Considerando a lei de controle para um atuador ativo e sendo \mathbf{K} sua matriz de ganho, tem-se:

$$\mathbf{u} = -\mathbf{Kz} \tag{3.33}$$

onde,

$$\mathbf{K} = [k_1 \quad k_2 \quad k_3 \quad k_4] \tag{3.34}$$

O processo de otimização consiste na obtenção dessa lei de controle de modo que uma função custo do tipo integral quadrática seja minimizada:

$$J(\mathbf{u}) = \int_{t_0}^{t_f} (\mathbf{z}^T * \mathbf{Q} * \mathbf{z} + 2 * \mathbf{z}^T * \mathbf{N} * \mathbf{u} + \mathbf{u}^T * \mathbf{R} * \mathbf{u}) d\tau \quad (3.35)$$

onde a matriz \mathbf{Q} representa a ponderação dos estados, a matriz \mathbf{R} a ponderação da força do controlador e a matriz \mathbf{N} a ponderação dos elementos cruzados, ou seja, das variáveis de estado e da força de controle (Alves, 1997).

Considerando a teoria de controle ótimo, obtêm-se através da seguinte expressão a matriz de ganho \mathbf{K} que minimizará a função de custo:

$$\mathbf{K} = \mathbf{R}^{-1} * (\mathbf{N}^T + \mathbf{B}^T * \mathbf{P}) \quad (3.36)$$

onde \mathbf{P} é uma matriz simétrica positiva definida, dada pela solução da equação de Ricatti:

$$\mathbf{P}(\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{N}^T) - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} - \mathbf{N}\mathbf{R}^{-1}\mathbf{N}^T + (\mathbf{A}^T - \mathbf{N}\mathbf{R}^{-1}\mathbf{B}^T) = \mathbf{0} \quad (3.37)$$

Substituindo a expressão da matriz \mathbf{K} na equação 3.33, tem-se a lei de controle para o atuador ativo:

$$\mathbf{u} = -\mathbf{R}^{-1} * (\mathbf{N}^T + \mathbf{B}^T * \mathbf{P}) * \mathbf{X} \quad (3.36)$$

A equação de movimento do sistema escrita na forma de estado é obtida substituindo a equação 3.33 em 3.23, logo teremos:

$$\dot{\mathbf{z}} = (\mathbf{A} - \mathbf{B} * \mathbf{K}) * \mathbf{z} + \mathbf{L} * \mathbf{w} \quad (3.37)$$

Fazendo

$$\mathbf{A}_k = \mathbf{A} - \mathbf{B} * \mathbf{K} \quad (3.38)$$

tem-se

$$\dot{z} = A_k * z + L * w \quad (3.39)$$

Para a obtenção das matriz de ganho do controlador **LQR** é necessário a determinação das matrizes de ponderação citadas anteriormente.

As matrizes de ponderação adotadas são obtidas a partir das restrições impostas ao sistema que devem ser minimizadas de modo a se ter o comportamento desejado. Os parâmetros que servem de medida dessas restrições são então representados através de um vetor que multiplica as variáveis de estado e/ou a força do controlador ativo. As matrizes de ponderação são então obtidas multiplicando-se a transposta deste vetor por ele próprio, pois a ponderação na função custo é na forma quadrática (Alves, 1997). Os parâmetros que devem ser ponderados na obtenção destas matrizes são: a aceleração vertical da massa suspensa (como uma medida indireta do desconforto), o espaço de trabalho da suspensão, a força de contato do pneu com o solo (relacionado com a segurança) e a energia do controlador. Todo o detalhamento de como se obtém estas matrizes encontra-se no anexo 1.

Capítulo 4

Redes Neurais

4.1 - Introdução

Este capítulo apresenta um estudo da aplicação das redes neurais artificiais (**RNA's**), sistemas compostos por diversos elementos simples, inspirados por elementos biológicos humanos, que se conectam entre si e operam em paralelo. É desenvolvido um modelo matemático com tratamento vetorial para as **RNA's** com base nas estruturas *perceptrons*, juntamente com os respectivos algoritmos de aprendizado baseado em um método de minimização de uma função custo, com a inclusão da retropropagação (*backpropagation*) de sinais através da rede.

Inicialmente foi feita uma revisão histórica desde o surgimento até o desenvolvimento das **RNA's**, sobretudo nos últimos 20 anos. Em seguida apresenta-se os seus principais componentes, características e a formulação matemática para a aplicação das **RNA's**, destacando as redes do tipo *perceptron* multicamadas (**MLP's**), que constituem um dos modelos mais conhecidos e utilizados. Destaca-se também as técnicas de aprendizado utilizadas, juntamente com os algoritmos empregados, baseados em métodos de minimização de primeira ordem, *backpropagation* (**BPA**), ou de segunda ordem, gradiente conjugado escalonado modificado (**SCGM**), que são usados para ajustar os pesos sinápticos das **RNA's** durante a fase de treinamento das mesmas.

4.2 – Redes Neurais Artificiais (RNA's)

O cérebro humano realiza várias tarefas como por exemplo reconhecimento de padrões, percepção e controle motor, muitíssimo mais rápido e melhor do que qualquer computador já produzido. Foi através do esforço para entender detalhadamente o funcionamento do cérebro humano que surgiu o estímulo inicial que conduziu ao desenvolvimento de modelos matemáticos de redes neurais. O objetivo era construir mecanismos que pudessem operar de modo similar ao cérebro humano, ou seja, que tomassem decisões, processassem informações, aprendessem, lembrassem e otimizassem da mesma forma que o nosso cérebro.

Hoje tem-se conseguido evidências concretas e convincentes acerca do enorme potencial destas estruturas quando aplicadas na análise e síntese de sistemas não lineares e na generalização de resultados expressivos já obtidos em outras áreas de atuação científica. Podemos citar dentre estas áreas, estatística, teoria de informação, teoria de aproximação de funções, teoria de processamento de sinais, teoria de controle de processos e otimização de sistemas. A recente maturidade atingida pelas RNA's como área de atuação científica tem contribuído para o desenvolvimento de ferramentas de engenharia mais eficazes e à utilização mais eficiente dos recursos computacionais hoje disponíveis, implicando assim numa ampliação sem precedentes na capacidade de manipular informações em suas mais diversas formas (De Castro, 1998a).

Em 1943 quando McCulloch e Pitts propuseram um modelo matemático simples para um neurônio deu-se início ao processo de evolução do estudo das RNA's. Por volta dos anos 60, as atividades de pesquisas na área foram dirigidas a se encontrar métodos para a obtenção dos pesos corretos na execução de uma dada tarefa. Em 1957, Rosenblat introduziu uma nova abordagem para o problema de reconhecimento de padrões com o desenvolvimento do *perceptron* (figura 4.1), e propôs por volta de 1961 um algoritmo de aprendizado convergente para o ajuste dos pesos do *perceptron*, causando um grande entusiasmo na comunidade científica. Por volta do mesmo período em 1962, Widrow e seus colaboradores desenvolveram o *adaline*, um outro modelo de neurônio.

Todavia, com a publicação do livro “*Perceptrons*” de Minsk e Papert em 1969, as pesquisas na área de redes neurais sofreram uma retração significativa. Neste livro, conceitos de matemática moderna como topologia e teoria de grupo são aplicados com o objetivo de analisar as capacidades adaptativas e computacionais dos *perceptrons*. Tais autores demonstraram que o perceptron não era capaz de aprender algumas funções elementares, como por exemplo a função lógica **XOR (OU-exclusivo)**. Além disso, esses autores não acreditavam que uma arquitetura adequada, juntamente com um algoritmo de ajustes de pesos, pudesse ser desenvolvido com o objetivo de superar esta limitação. Após a publicação destes resultados, houve um êxodo generalizado de pesquisadores da área de redes neurais, que passaram a buscar alternativas dentro do campo da engenharia e, principalmente, da lógica matemática, em franca expansão na época, devido às grandes conquistas realizadas na área de computação.

Alguns grupos continuaram as pesquisas durante os anos 70. A maioria dos trabalhos foram na área de memória associativa, em que diferentes padrões de entrada são associados com outros de suficiente similaridade. Os nomes de Kohonen, Grossberg, Widrow, Anderson, Caianiello, Fukushima e Aleksander estão associados a este período.

Em 1982 o ganhador do Prêmio Nobel J. Hopfield, juntamente com o pesquisador D. Tank, desenvolveram grande quantidade de modelos de redes neurais baseadas em pesos fixos e ativações adaptativas. Estas redes podem servir como memórias autoassociativas e serem usadas para resolver problemas de otimização restrita.

O fato que efetivamente colocou a área de rede neurais novamente em destaque na comunidade científica foi o desenvolvimento de um método para ajuste de parâmetros de redes não-recorrentes de múltiplas camadas. Este método, baseado em um algoritmo denominado retropropagação (*backpropagation*), tornou-se largamente conhecido em 1986, após a publicação dos trabalhos de McClelland e Rumelhart, fazendo com que pesquisadores das mais diferentes áreas passassem a visualizar interessantes aplicações para as **RNA's**.

O interesse pelas **RNA's** pode ser atribuído a vários fatores, tais como: técnicas de treinamento foram desenvolvidas para as arquiteturas mais sofisticadas, capazes de resolver os problemas que causaram o descrédito dos primeiros modelos; contribuições provenientes de

outras áreas têm guiado o processo de formalização dos resultados, substituindo procedimentos empíricos de validação; a tecnologia atualmente disponível pode ser utilizada para construir *hardware* específicos para RNA's; ao mesmo tempo que o progresso da computação clássica tornou o estudo das RNA's mais promissor, limitações encontradas no processamento inerentemente seqüencial destas máquinas, motivaram novos rumos na pesquisa em redes neurais; e por fim, as mais recentes abordagens sobre computação paralela podem se beneficiar do estudo dos sistemas neurais biológicos, que são inerentemente paralelos (De Castro, 1998a).

4.3 – Características Principais

As RNA's têm sido desenvolvidas como generalizações de modelos matemáticos da biologia neural humana, considerando que o processo da informação ocorre em vários elementos chamados neurônios; que os sinais são propagados de um elemento a outro através de conexões; que cada conexão possui um *peso* associado, que em uma rede neural típica, pondera o sinal transmitido; e cada neurônio aplica uma *função de ativação* (geralmente não linear) à sua entrada de rede (soma ponderada dos sinais de entrada) para determinar sua saída.

Uma rede neural pode ser caracterizada por três aspectos principais: o padrão de conexões entre as unidades – *arquitetura*; o método de determinação dos pesos destas conexões – *algoritmo de treinamento ou aprendizado* e sua *função de ativação*.

“*Aprender*” para uma rede neural, trata-se da habilidade de adaptar-se, de acordo com regras pré-existentes, ao seu ambiente, alterando seu desempenho ao longo do tempo. Sendo assim, considera-se *aprendizado* o processo que adapta o comportamento e conduz a melhoria de desempenho, ou seja, as redes neurais tem a capacidade de “*aprender*” a resolver problemas através de exemplos (Von Zuben, 1999).

O aprendizado ou treinamento das RNA's pode ser dividido em: aprendizado supervisionado - é baseado em um conjunto de exemplos de estímulo-resposta (entrada-saída), ou

em algum tipo de informação que represente o comportamento que deve ser apresentado pela rede neural, dizemos então que este tipo de treinamento necessita de um “*professor*” durante a fase de aprendizagem, que antecede a utilização da rede; aprendizado por reforço – o comportamento da rede é avaliado apenas com base em algum critério numérico, fornecido em instantes espaçados de tempo; e aprendizado não-supervisionado – é baseado apenas nos estímulos recebidos pela rede neural. Basicamente, a rede deve aprender a “categorizar” os estímulos e portanto não necessita de um “*professor*” (Von Zuben, 1999).

4.4 – Modelo de um Neurônio

Um neurônio é a unidade fundamental de processamento de informações de uma rede neural (Haykin, 1999). O modelo de um neurônio artificial tipo perceptron é apresentado na figura 4.1.

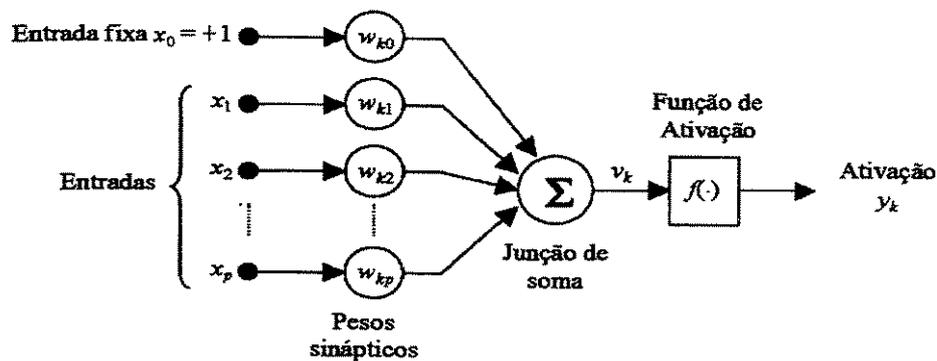


Figura 4.1 – Modelo de um neurônio tipo *perceptron*.

Identificamos três elementos básicos em um modelo neuronal:

1 – Um conjunto de *sinapses* ou *links de conexão*, cada um dos quais caracterizado por um *peso*. O *peso sináptico* de um neurônio artificial pode variar em um intervalo que inclui valores positivos e negativos. Sendo assim, um sinal x_j na entrada da sinapse j conectada ao neurônio k é

multiplicado pelo peso sináptico w_{kj} , onde o primeiro índice se refere ao neurônio em questão e o segundo índice ao terminal de entrada da sinapse ao qual o peso se refere. Quando uma entrada fixa ou “bias” está presente (entrada x_0 na figura 4.1), então o peso sináptico correspondente é denominado *peso de polarização*. O “bias” tem o efeito de aumentar ou diminuir (transladar) a rede de entrada da função de ativação, dependendo se esta é positiva ou negativa, respectivamente.

2 – Uma *junção de soma*, responsável pela combinação aditiva dos sinais de entrada ponderados pelos respectivos pesos das sinapses do neurônio.

3 – Uma *função de ativação* geralmente não linear e de formato sigmoidal, representando um efeito de saturação na ativação de saída y_k do neurônio, ou seja, limita a amplitude da saída de um neurônio dentro de uma faixa permitida. Tipicamente o intervalo de amplitude normalizado da saída de um neurônio é: $[0, 1]$ ou $[-1, 1]$.

O modelo matemático do neurônio da figura 4.1 pode ser descrito pelas seguintes equações:

$$v_k = \sum_{j=0}^p w_{kj} x_j \quad (4.1)$$

$$y_k = f(v_k) \quad (4.2)$$

onde x_0, x_1, \dots, x_p , são os sinais de entrada, $w_{k0}, w_{k1}, \dots, w_{kp}$ são os pesos sinápticos do neurônio k , v_k é o potencial de ativação do neurônio k , $f(\cdot)$ é a função de ativação e y_k é a ativação de saída do neurônio k .

Observe na figura 4.1 a presença de uma entrada de polarização fixa ou “bias” $x_0 = +1$. Esta entrada, juntamente com o peso a ela associada, tem o efeito de transladar a função de ativação em torno da origem (transformação afim), fazendo com que a ativação interna v_k do neurônio não seja nula quando todas as demais entradas $\{x_1, x_2, \dots, x_p\}$ forem nulas (Iyoda, 2000). O valor do

“*bias*” pode ser também -1 ou 0 , dependendo do problema em questão. Podemos então rescrever a equação 4.2 sob o efeito da polarização como:

$$y_k = f\left(\sum_{j=1}^p w_{kj}x_j - w_{k0}\right) \quad (4.3)$$

Escrevendo em notação vetorial a ativação do neurônio, temos:

$$\mathbf{x} = [x_0 \quad x_1 \quad \dots \quad x_p]^T \quad \text{vetor de entradas do neurônio } k. \quad (4.4)$$

$$\mathbf{w} = [w_{k0} \quad w_{k1} \quad \dots \quad w_{kp}]^T \quad \text{vetor de pesos do neurônio } k. \quad (4.5)$$

então a ativação y_k é dada por

$$y_k(\mathbf{w}, \mathbf{x}) = f(\mathbf{w}^T * \mathbf{x}) \quad (4.6)$$

4.4.1 – Tipos de Funções de Ativação

A função de ativação representa o efeito que a entrada interna e o estado atual de ativação exercem na definição do próximo estado de ativação da unidade. As características fundamentais da função de ativação são: comportamento monotônico sobre uma faixa do argumento v , conhecida como “faixa dinâmica” e a saturação fora desta faixa, ou seja, ela é monotonicamente não-decrescente e apresenta um tipo de não linearidade associada ao efeito da saturação.

Podemos identificar três classes principais de função de ativação (Iyoda, 2000):

1 – *Função Sinal (Heaviside)*. Para este tipo de função, ilustrada na figura 4.2(a), temos

$$f(v) = \begin{cases} 1 & \text{se } v > 0 \\ 0 & \text{se } v \leq 0 \end{cases} \quad (4.7)$$

Assim, a ativação de um neurônio k que usa este tipo de função de ativação é dada por:

$$y_k = \begin{cases} 1 & \text{se } v_k > 0 \\ 0 & \text{se } v_k \leq 0 \end{cases} \quad (4.8)$$

onde v_k é o nível de ativação interno do neurônio, definido na equação 4.1.

2 – *Função Linear por Partes ou Semi-linear*. Para este tipo de função, ilustrada na figura 4.2(b), temos

$$f(v) = \begin{cases} 0 & \text{se } v \leq -a \\ v + a & \text{se } -a < v < a \\ 2a & \text{se } v \geq a \end{cases} \quad (4.8)$$

No limite quando a tende a zero, esta função se aproxima assintoticamente da função sinal..

3 – *Função Sigmoidal*. É a função mais utilizada em **RNA's**. Ela é definida como uma função monotônica crescente que apresenta propriedades assintóticas e de suavidade. Um exemplo de função sigmoidal é a chamada função logística ilustrada na figura 4.2(c), onde temos

$$f(v) = \frac{1}{1 + e^{-av}} \quad (4.9)$$

onde a é o parâmetro de inclinação da função sigmoideal. Variando o parâmetro a podemos obter funções sigmoideais com diferentes inclinações, como mostrado na figura 4.2(c). Para esta função, no limite quando a tende a zero, esta função se aproxima assintoticamente da função sinal.

Em muitas situações é desejável termos uma função sigmoideal que varie entre -1 e $+1$. Nestas situações, uma função bastante empregada é a *tangente hiperbólica*, ilustrada na figura 4.2(d), e definida por:

$$f(v) = \operatorname{tgh}\left(\frac{v}{2}\right) = \frac{1 - e^{-av}}{1 + e^{-av}} \quad (4.10)$$

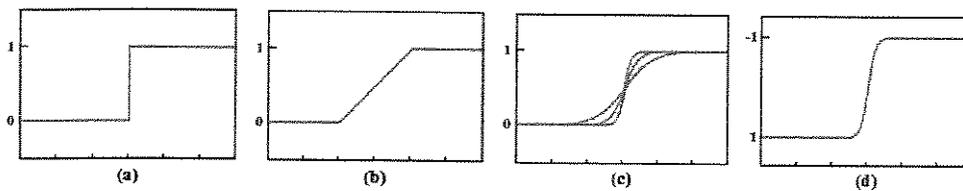


Figura 4.2 – Funções de Ativação: (a) função sinal, (b) função semi-linear, (c) função sigmoideal e (d) função tangente hiperbólica.

4.5 – Redes Neurais Tipo Perceptron Multicamadas (MLP's)

A rede neural do tipo perceptron multicamadas (MLP's) constituem um dos modelos de redes mais utilizados e conhecidos. São arquiteturas onde os neurônios são organizados em camadas de processamento. As razões de sua grande popularidade estão na capacidade de aproximação universal e flexibilidade para formar soluções de qualidade para uma vasta classe de problemas, a partir de um mesmo algoritmo de aprendizado.

Esta arquitetura consiste de um conjunto de unidades sensoriais que formam uma camada de entrada, uma ou mais camadas intermediárias (ou escondidas) e uma camada de saída (figura

4.3). Os neurônios da camada de entrada são neurônios especiais, cujo papel é exclusivamente distribuir cada uma das entradas da rede (sem modificá-las) a todos os neurônios da camada posterior. Os sinais de entrada são então propagados camada a camada pela rede em uma direção positiva, ou seja, da entrada para saída. Esta arquitetura representa uma generalização do *perceptron* apresentado na figura 4.1.

As redes **MLP's** tem sido utilizadas com sucesso para a solução de vários problemas envolvendo altos graus de não linearidade. Na figura 4.3 apresentamos uma rede **MLP** com duas camadas intermediárias. A rede apresentada aqui possui todas as conexões, o que significa que um neurônio de qualquer camada da rede está conectado a todas as outras unidades (neurônios) na camada anterior. O fluxo de sinais através da rede é feito positivamente, da esquerda para a direita, camada a camada. Podemos identificar dois tipos de sinais em uma rede: *senal funcional* – é um sinal de entrada (estímulo) que chega na entrada e é propagado neurônio a neurônio através da rede, e aparece na saída como um sinal de saída, e o *senal de erro* – que origina-se nas saídas e é retropropagado através da rede. Em uma MLP cada unidade intermediária ou de saída é responsável por duas tarefas: calcular o sinal na saída da unidade, que geralmente é expresso como uma função não linear do sinal de entrada e dos pesos sinápticos associados e calcular uma estimativa instantânea do vetor gradiente, que é necessário para a retropropagação do erro através da rede (De Castro, 1998a).

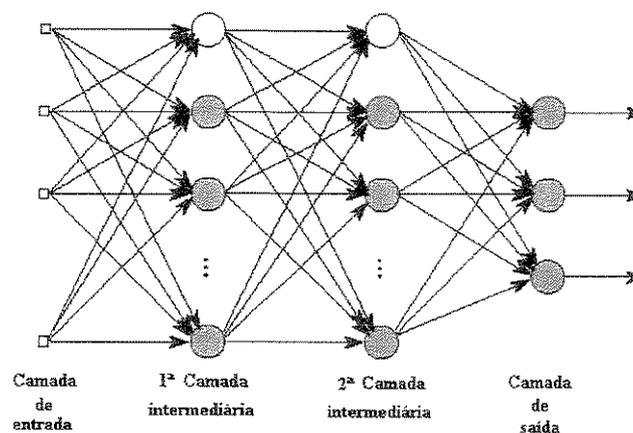


Figura 4.3 – Arquitetura de uma rede **MLP** com duas camadas intermediárias.

4.6 – Aprendizado Supervisionado

A capacidade de aprendizado de uma RNA é sem dúvida uma de suas mais importantes propriedades. Uma rede neural aprende através de um processo iterativo de ajuste dos seus pesos sinápticos e limiares, o qual pode ser expresso na forma de um algoritmo computacional. Este processo de aprendizagem segue uma seqüência lógica de eventos: a rede neural é estimulada pelo ambiente de informação; a estrutura interna da rede é alterada como resultado do estímulo; e devido às alterações que ocorreram em sua estrutura interna, a rede tem modificada sua resposta aos estímulos do ambiente (Iyoda, 2000).

O *aprendizado supervisionado* como foi dito anteriormente, é caracterizado pela presença de um “*professor*” externo. A função do “professor” é fornecer a rede neural durante o processo de aprendizado uma resposta desejada a um determinado estímulo apresentado pelo ambiente. Um sinal de erro pode ser definido como a diferença entre a resposta desejada e a resposta observada na saída da rede, sendo posteriormente usado para ajustar os parâmetros da mesma. Temos na figura 4.4 o diagrama de blocos de um sistema com aprendizado supervisionado.

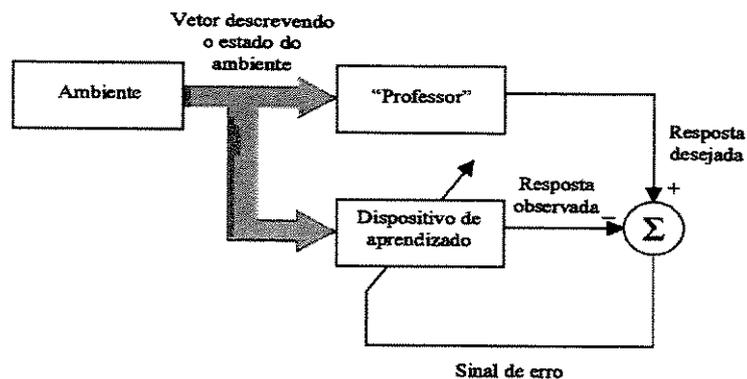


Figura 4.4 – Diagrama de blocos de um sistema com aprendizado supervisionado

Uma forma de se implementar em redes neurais um aprendizado supervisionado é através de procedimentos iterativos de correção de erro. Seja $y_{d_k}(\mathbf{n})$ a resposta desejada para um neurônio k no instante \mathbf{n} e seja $y_k(\mathbf{n})$ a resposta gerada por este neurônio. A resposta $y_k(\mathbf{n})$ é produzida por um estímulo (vetor $\mathbf{x}(\mathbf{n})$) aplicado à entrada da rede da qual o neurônio k faz parte.

O sinal de erro para o neurônio k é definido como a diferença entre a resposta desejada e a resposta observada:

$$\mathbf{e}_k(\mathbf{n}) = \mathbf{y}d_k(\mathbf{n}) - \mathbf{y}_k(\mathbf{n}) \quad (4.11)$$

O objetivo do aprendizado por correção de erro é minimizar alguma função de custo baseada no sinal de erro $\mathbf{e}_k(\mathbf{n})$, de modo que a resposta observada de cada neurônio da rede se aproxime da resposta desejada para aquele neurônio, em algum sentido estatístico. De fato, uma vez definida uma função de custo, o problema de aprendizado torna-se um problema de otimização (Iyoda, 2000). Uma função de custo bastante empregada é a soma dos erros quadráticos:

$$\mathbf{J} = \frac{1}{2} \sum_k \mathbf{e}_k^2(\mathbf{n}) = \frac{1}{2} \sum_k (\mathbf{y}d_k(\mathbf{n}) - \mathbf{y}_k(\mathbf{n}))^2 \quad (4.12)$$

onde o somatório é realizado sobre todos os neurônios da camada de saída da rede neural. O aprendizado ocorre então através da minimização da função custo \mathbf{J} em relação aos pesos sinápticos da rede. Observe que esta função custo define uma superfície de erro sobre o espaço dos pesos. Se \mathbf{P} é o número de pesos ajustáveis da rede neural, então $\mathbf{J} : \mathfrak{R}^{\mathbf{P}} \rightarrow \mathfrak{R}$. A superfície de erro é caracterizada pela presença de mínimos locais e um ou mais mínimos globais. Os métodos de otimização utilizados na minimização de \mathbf{J} usualmente recorrem à informação de gradiente do erro para ajustar os parâmetros da rede. Teoricamente, estes métodos sempre atingem um ponto de mínimo da superfície de erro, mas observe que nada se pode afirmar sobre a natureza (local ou global) do ponto de mínimo obtido a partir de uma condição inicial arbitrária (Iyoda, 2000).

4.7 – Algoritmos de Treinamento

Sabemos que o tipo de aprendizado ou de treinamento da rede é determinado pela técnica empregada no processo de ajustes dos pesos sinápticos (parâmetros da rede neural). Um conjunto

bem definido de regras para obtê-los é denominado algoritmo de aprendizado. Vamos citar a seguir dois algoritmos que serão usados neste trabalho.

4.7.1 – O *Backpropagation* - Algoritmo de Retropropagação

O algoritmo mais utilizado no treinamento de redes neurais multicamadas com uma ou mais camadas intermediárias é chamado retropropagação do erro (*error backpropagation*) ou simplesmente retropropagação. Este algoritmo é baseado numa regra de aprendizagem que “corrige” o erro durante o treinamento. Basicamente, o algoritmo de retropropagação consiste em dois passos de computação: o processamento direto (*feedforward*) e o processamento reverso (*backpropagation*). No processamento direto, uma entrada é aplicada à rede neural e seu efeito é propagado pela rede, camada a camada. Durante o processamento direto, os pesos da rede permanecem fixos. No processamento reverso, um sinal de erro calculado na saída da rede é propagado no sentido reverso, camada a camada, e ao final deste processo os pesos são ajustados de acordo com uma regra de correção de erro. Uma determinada entrada é apresentada e também é disponibilizada a resposta desejada para aquela entrada, caracterizando o aprendizado supervisionado.

Considere a figura 4.5, que mostra o neurônio de saída j de uma rede **MLP**, alimentado pelas ativações de todos os neurônios da camada imediatamente anterior.

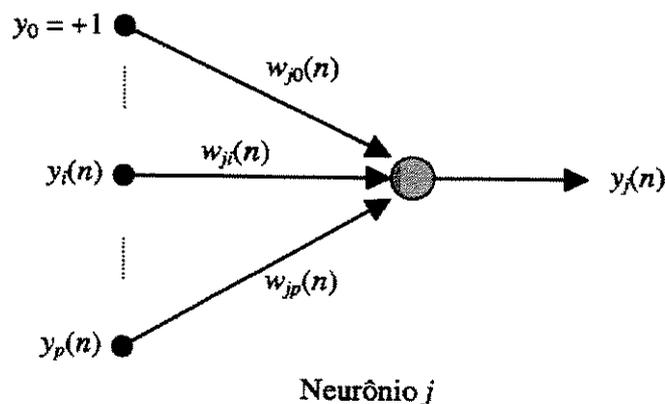


Figura 4.5 – Neurônio de saída j de uma rede **MLP**.

O sinal de erro na saída do neurônio j na iteração n (isto é, apresentação do n -ésimo padrão de treinamento) é definido pela equação 4.11, substituindo-se o índice k por j , onde $yd_j(n)$ é a resposta desejada para o neurônio j da camada de saída. A soma dos erros quadráticos instantâneos da rede é então definida por

$$\varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (4.13)$$

onde C é o conjunto que contém todos os neurônios da camada de saída da rede. Seja N o número total de padrões de treinamento contido no conjunto de treinamento. O erro quadrático médio é então definido por

$$\varepsilon_{av} = \frac{1}{N} \sum_{n=1}^N e_j^2(n) \quad (4.14)$$

Observe que a soma dos erros quadráticos instantâneos, e portanto o erro quadrático médio, é uma função de todos os pesos sinápticos da rede, já que são eles que definem a função de transferência ou mapeamento de entrada-saída da rede neural. Para um dado conjunto de treinamento, ε_{av} representa uma função de custo, isto é, uma medida de desempenho da rede neural. O objetivo do processo de treinamento da rede é justamente minimizar esta função de custo (Iyoda, 2000).

Em aplicações práticas do algoritmo de retropropagação, o aprendizado é resultado de apresentações repetidas de todos os padrões do conjunto de treinamento à rede. Cada apresentação de todo o conjunto de treinamento durante o processo de aprendizagem é chamada de *época*. O processo de aprendizagem é repetido época após época, até que o conjunto de pesos e limiares estabilize e o erro quadrático médio do conjunto de treinamento convirja para um valor mínimo. É uma boa prática fazer com que a ordem de apresentação dos padrões seja feita aleatoriamente de uma época para a outra. Esta aleatoriedade tende a fazer com que a busca no

espaço de pesos tenha um caráter estocástico ao longo dos ciclos de treinamento (De Castro, 1998a).

Para um dado conjunto de treinamento, o aprendizado por retropropagação pode ser feito por dois métodos de treinamento: o *padrão-a-padrão* (*pattern mode*) ou o por *lote ou batelada* (*batch mode*).

No método *padrão-a-padrão* a atualização dos pesos é feita imediatamente após a apresentação de cada padrão ou amostra (par entrada – resposta desejada) de treinamento à rede neural. Mais especificamente, considere uma época consistindo da apresentação de N padrões de treinamento organizados na ordem $[\mathbf{x}(1) \mathbf{y}_d(1)], \dots, [\mathbf{x}(N) \mathbf{y}_d(N)]$. O primeiro exemplo da época $[\mathbf{x}(1) \mathbf{y}_d(1)]$ é apresentado à rede, e a seqüência de processamentos direto e reverso é realizada, resultando no ajuste de pesos da rede, incluindo os pesos de polarização. Então o segundo exemplo da época $[\mathbf{x}(2) \mathbf{y}_d(2)]$ é apresentado, e a seqüência de processamentos se repete. Este processo continua até que o último padrão $[\mathbf{x}(N) \mathbf{y}_d(N)]$ seja apresentado à rede, finalizando a época. Neste modo de treinamento, é importante apresentar os padrões em ordem aleatória a cada época, já que a ordem de apresentação dos padrões de treinamento não representa nenhuma informação que deva condicionar o processo de ajuste de pesos.

No método por *lote ou batelada* os pesos são ajustados somente após a apresentação de todos os exemplos que constituem uma época. Para uma determinada época, definimos a função de custo como o erro quadrático médio, de forma similar a equação 4.14 só que em forma expandida

$$\mathcal{E}_{av} = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} e_j^2(n) \quad (4.15)$$

A correção a ser aplicada a cada peso w_{ji} será proporcional à derivada parcial de \mathcal{E}_{av} em relação a w_{ji} . Posteriormente mostraremos como ocorre este cálculo.

O desempenho de um modo ou outro de treinamento depende do da sua aplicação prática, ou seja, do problema em questão (Haykin, 1994). O método padrão-a-padrão parece apresentar melhor desempenho em problemas caracterizados por um conjunto de treinamento grande contendo um grande número de informação redundante e, por superfícies de erro não muito complexas. Já o método por lote, entretanto, é superior em problemas que não apresentem estas características (Iyoda, 2000).

Inicialmente, vamos considerar o método de treinamento padrão-a-padrão. O nível de ativação interno do neurônio j é dado por

$$v_j(\mathbf{n}) = \sum_{i=0}^p w_{ji}(\mathbf{n})y_i(\mathbf{n}) \quad (4.16)$$

onde p é o número total de entradas (excluindo a polarização) aplicadas ao neurônio j (figura 4.5), ou seja, o número de neurônios da camada imediatamente anterior. Portanto a ativação $y_j(\mathbf{n})$ do neurônio j é dada por

$$y_j(\mathbf{n}) = f(v_j(\mathbf{n})) \quad (4.17)$$

Para minimizar o erro quadrático médio, necessitamos primeiramente determinar o gradiente instantâneo $\partial \epsilon(\mathbf{n}) / \partial w_{ji}(\mathbf{n})$. Aplicando a regra da cadeia, podemos expressar este gradiente como

$$\frac{\partial \epsilon(\mathbf{n})}{\partial w_{ji}(\mathbf{n})} = \frac{\partial \epsilon(\mathbf{n})}{\partial e_j(\mathbf{n})} \frac{\partial e_j(\mathbf{n})}{\partial y_j(\mathbf{n})} \frac{\partial y_j(\mathbf{n})}{\partial v_j(\mathbf{n})} \frac{\partial v_j(\mathbf{n})}{\partial w_{ji}(\mathbf{n})} \quad (4.18)$$

Vamos obter as derivadas parciais, derivando a equação 4.13 em relação a $e_j(\mathbf{n})$, obtemos

$$\frac{\partial \epsilon(\mathbf{n})}{\partial e_j(\mathbf{n})} = e_j(\mathbf{n}) \quad (4.19)$$

Derivando a equação 4.11, com a mudança de índice para o neurônio j , em relação a $y_j(\mathbf{n})$, obtemos

$$\frac{\partial e_j(\mathbf{n})}{\partial y_j(\mathbf{n})} = -1 \quad (4.20)$$

Agora, derivando a equação 4.17 em relação a $v_j(\mathbf{n})$, obtemos

$$\frac{\partial y_j(\mathbf{n})}{\partial v_j(\mathbf{n})} = f_j'(v_j(\mathbf{n})) \quad (4.21)$$

onde o símbolo ' significa derivação em relação ao argumento. Por fim, derivando a equação 4.16 em relação a $w_{ji}(\mathbf{n})$ obtemos

$$\frac{\partial v_j(\mathbf{n})}{\partial w_{ji}(\mathbf{n})} = y_i(\mathbf{n}) \quad (4.22)$$

Substituindo as equações 4.19 a 4.22 na equação 4.18, obtemos

$$\frac{\partial \varepsilon(\mathbf{n})}{\partial w_{ji}(\mathbf{n})} = -e_j(\mathbf{n})f_j'(v_j(\mathbf{n}))y_i(\mathbf{n}) \quad (4.23)$$

que é a equação da derivada do erro instantâneo $\varepsilon(\mathbf{n})$ em relação ao peso sináptico $w_{ji}(\mathbf{n})$ do neurônio j da camada de saída. Observe que a equação 4.23 corresponde a um componente do vetor gradiente do erro, cujos elementos representam a derivada parcial de $\varepsilon(\mathbf{n})$ em relação a todos os pesos da rede neural, arranjados em uma ordem fixa, mas arbitrária (Iyoda, 2000). Definindo o gradiente local $\delta_j(\mathbf{n})$ como

$$\delta_j(\mathbf{n}) = -\frac{\partial \varepsilon(\mathbf{n})}{\partial e_j(\mathbf{n})} \frac{\partial e_j(\mathbf{n})}{\partial y_j(\mathbf{n})} \frac{\partial y_j(\mathbf{n})}{\partial v_j(\mathbf{n})} = e_j(\mathbf{n})f_j'(v_j(\mathbf{n})) \quad (4.24)$$

podemos então rescrever a equação 4.23 na seguinte forma

$$\frac{\partial \varepsilon(\mathbf{n})}{\partial w_{ji}(\mathbf{n})} = -\delta_j(\mathbf{n})y_i(\mathbf{n}) \quad (4.25)$$

Para minimizar o erro, vamos aplicar aos pesos uma correção proporcional ao oposto do gradiente do erro, pois vamos caminhar no espaço de pesos na direção oposta ao do gradiente (vamos retropropagar o erro). Matematicamente, podemos expressar esta regra de ajuste na forma

$$\begin{aligned} \Delta w_{ji}(\mathbf{n}) &= -\eta(\mathbf{n}) \frac{\partial \varepsilon(\mathbf{n})}{\partial w_{ji}(\mathbf{n})} \\ w_{ji}(\mathbf{n}+1) &= w_{ji}(\mathbf{n}) - \eta(\mathbf{n}) \frac{\partial \varepsilon(\mathbf{n})}{\partial w_{ji}(\mathbf{n})} \\ w_{ji}(\mathbf{n}+1) &= w_{ji}(\mathbf{n}) + \eta(\mathbf{n}) \delta_j(\mathbf{n})y_i(\mathbf{n}) \end{aligned} \quad (4.26)$$

onde $\eta(\mathbf{n})$ é o *passo* ou *taxa de aprendizagem*, constante em todas as interações do algoritmo para cada iteração \mathbf{n} . Da teoria de otimização não linear de funções sabe-se que há alternativas melhores a esta abordagem, como o uso de procedimentos de busca unidimensional para determinar $\eta(\mathbf{n})$ a cada iteração do algoritmo, chamados “*backpropagation* com taxa de aprendizagem variável” (Von Zuben, 1999).

A equação 4.26 é uma equação a diferenças de 1ª ordem que permite a solução do problema proposto (mínimo local). Alguns autores, no entanto incluem um termo adicional a esta equação, que então adquire a forma

$$w_{ji}(\mathbf{n}+1) = w_{ji}(\mathbf{n}) + \beta(\mathbf{n})\Delta w_{ji}(\mathbf{n}) - \eta(\mathbf{n}) \frac{\partial \varepsilon(\mathbf{n})}{\partial w_{ji}(\mathbf{n})} \quad (4.27)$$

o termo $\Delta w_{ji}(\mathbf{n}) = w_{ji}(\mathbf{n}) - w_{ji}(\mathbf{n}-1)$ representa a “inércia” do processo de ajuste de pesos, com $\beta(\mathbf{n})$ sendo denominado coeficiente de momento. A consequência imediata da inclusão deste termo é a transformação da equação 4.27 em uma equação a diferença de 2ª ordem. Com isso, dependendo

dos valores de $\eta(\mathbf{n})$ e $\beta(\mathbf{n})$, o ajuste de pesos pode adquirir um comportamento dinâmico cujos efeitos no processo de solução do problema de minimização da função custo podem ser positivos (Von Zuben, 1999). No trabalho de Vogl, 1988, podemos encontrar estas modificações do *backpropagation* e uma comparação do resultado do treinamento para cada modelo proposto.

Observe, nas equações 4.24 e 4.25, que um fator fundamental para o cálculo do gradiente do erro é o sinal do erro $\mathbf{e}_j(\mathbf{n})$. Devemos então considerar dois casos distintos: no caso I, o neurônio j é um neurônio da camada de saída da rede e no caso II, o neurônio j é um neurônio pertencente a uma camada intermediária. O caso I é bastante simples, pois as saídas desejadas para a rede neural são conhecidas. Já no caso II, não sabemos qual a saída desejada para um neurônio escondido, e portanto, não é possível calcular diretamente o sinal de erro. Este problema é resolvido de forma bastante elegante através da retropropagação do erro de saída através da rede (Iyoda, 2000).

Vamos considerar cada caso individualmente.

- **Caso I: Neurônio j é um Neurônio da Camada de Saída**

Quando o neurônio j é um neurônio de saída, sabemos qual é a saída desejada $\mathbf{y}_d(\mathbf{n})$ para o neurônio. Assim, podemos usar a equação 4.11 para calcular o sinal de erro $\mathbf{e}_j(\mathbf{n})$ para em seguida determinarmos diretamente $\delta_j(\mathbf{n})$ usando a equação 4.24.

- **Caso II: Neurônio j é um Neurônio da Camada Intermediária**

Quando o neurônio j é um neurônio da camada intermediária, não há nenhuma saída desejada pré-especificada para o neurônio. 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 o neurônio da camada intermediária está diretamente conectado. Considere a figura 4.6, que mostra o neurônio j como um neurônio intermediário da rede.

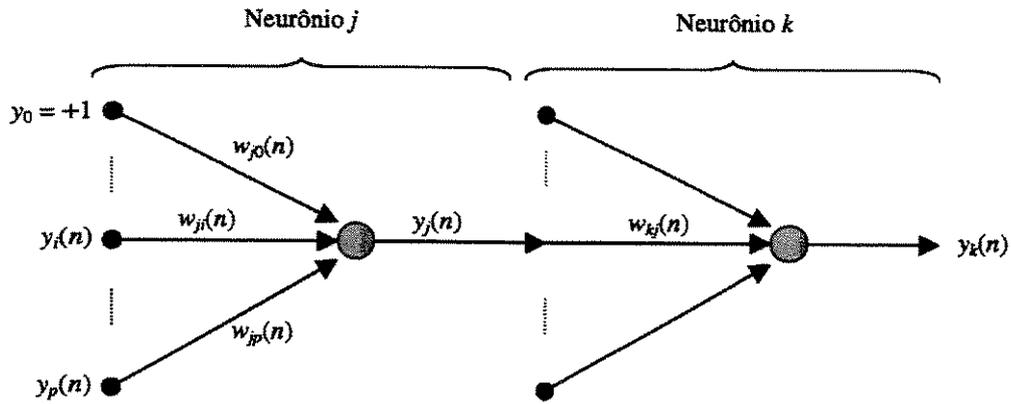


Figura 4.6 – Neurônio j da camada intermediária conectado a um neurônio da camada de saída k .

Da equação 4.24, podemos redefinir o gradiente local $\delta_j(\mathbf{n})$ para o neurônio da camada intermediária j como

$$\delta_j(\mathbf{n}) = -\frac{\partial \varepsilon(\mathbf{n})}{\partial y_j(\mathbf{n})} \frac{\partial y_j(\mathbf{n})}{\partial v_j(\mathbf{n})} = -\frac{\partial \varepsilon(\mathbf{n})}{\partial y_j(\mathbf{n})} f'_j(v_j(\mathbf{n})) \quad (4.28)$$

na qual usamos a equação 4.21. Para calcular a derivada parcial $\partial \varepsilon(\mathbf{n}) / \partial y_j(\mathbf{n})$, podemos proceder da seguinte forma:

Da figura 4.6, observamos que, para o neurônio k pertencente à camada de saída, temos

$$\varepsilon(\mathbf{n}) = \frac{1}{2} \sum_{k \in C} e_k^2(\mathbf{n}) \quad (4.29)$$

Derivando a equação 4.29 em relação a $y_j(\mathbf{n})$, obtemos

$$\frac{\partial \varepsilon(\mathbf{n})}{\partial y_j(\mathbf{n})} = \sum_k e_k(\mathbf{n}) \frac{\partial e_k(\mathbf{n})}{\partial y_j(\mathbf{n})} \quad (4.30)$$

Usando a regra da cadeia para calcular $\partial e_k(\mathbf{n})/\partial y_j(\mathbf{n})$, podemos rescrever a equação 4.30 na forma

$$\frac{\partial \varepsilon(\mathbf{n})}{\partial y_j(\mathbf{n})} = \sum_k e_k(\mathbf{n}) \frac{\partial e_k(\mathbf{n})}{\partial v_k(\mathbf{n})} \frac{\partial v_k(\mathbf{n})}{\partial y_j(\mathbf{n})} \quad (4.30)$$

O erro na saída do neurônio k é dado por

$$e_k(\mathbf{n}) = y_{d_k}(\mathbf{n}) - y_k(\mathbf{n}) = y_{d_k}(\mathbf{n}) - f_k(v_k(\mathbf{n})) \quad (4.31)$$

onde $v_k(\mathbf{n})$ é o nível de ativação interna do neurônio k . Portanto,

$$\frac{\partial e_k(\mathbf{n})}{\partial v_k(\mathbf{n})} = -f'_k(v_k(\mathbf{n})) \quad (4.32)$$

Da figura 4.6, podemos observar que o nível de ativação interna do neurônio k é dado por

$$v_k(\mathbf{n}) = \sum_{j=0}^q w_{kj}(\mathbf{n}) y_j(\mathbf{n}) \quad (4.33)$$

onde q é o número total de entradas (excluindo a polarização) aplicadas ao neurônio k . Derivando a equação 4.33 em relação a $y_j(\mathbf{n})$ teremos

$$\frac{\partial v_k(\mathbf{n})}{\partial y_j(\mathbf{n})} = w_{kj}(\mathbf{n}) \quad (4.34)$$

Substituindo as equações 4.32 e 4.34 na equação 4.30 obtemos a derivada parcial desejada:

$$\frac{\partial \varepsilon(\mathbf{n})}{\partial y_j(\mathbf{n})} = - \sum_k e_k(\mathbf{n}) f'_k(v_k(\mathbf{n})) w_{kj}(\mathbf{n}) = - \sum_k \delta_k(\mathbf{n}) w_{kj}(\mathbf{n}) \quad (4.35)$$

onde, na segunda igualdade, fizemos uso da definição de gradiente local $\delta_k(\mathbf{n})$ dada na equação 4.24, com a substituição dos índices.

Por fim, substituindo a equação 4.35 na equação 4.28, obtemos a expressão para o gradiente local $\delta_j(\mathbf{n})$ para o neurônio da camada intermediária j :

$$\delta_j(\mathbf{n}) = f_j'(v_j(\mathbf{n})) \sum_k \delta_k(\mathbf{n}) w_{kj}(\mathbf{n}) \quad (4.36)$$

O termo $f_j'(v_j(\mathbf{n}))$ na equação 4.36 depende apenas da função de ativação associada ao neurônio j . Já o termo restante, isto é, o somatório sobre k , depende de dois conjuntos de termos. O primeiro conjunto de termos, os $\delta_k(\mathbf{n})$, exigem o conhecimento dos sinais de erro $e_k(\mathbf{n})$, para todos os neurônios localizados na camada imediatamente posterior à camada onde se encontra o neurônio j , e que estão diretamente conectados ao neurônio j ; veja a figura 4.6. O segundo conjunto de termos, os $w_{kj}(\mathbf{n})$, consiste nos pesos sinápticos associados a estas conexões.

Para o método de treinamento por lote ou batelada a correção aplicada a cada peso w_{ji} será proporcional à derivada parcial de ε_{av} em relação a w_{ji} . Temos então

$$\frac{\partial \varepsilon_{av}}{\partial w_{ji}} = \frac{1}{N} \sum_{\mathbf{n}=1}^N \frac{\partial \varepsilon(\mathbf{n})}{\partial w_{ji}} \quad (4.37)$$

O procedimento de cálculo é o mesmo descrito anteriormente.

Vimos anteriormente que o algoritmo de retropropagação envolve a execução de dois passos distintos de computação: o processamento direto e o processamento reverso. O procedimento direto é executado no sentido entrada \rightarrow saída da rede. Neste caso, um exemplo de treinamento é apresentado à rede e as saídas de todos os neurônios são computadas, usando as equações 4.16 e 4.17. Comparamos então as saídas da rede com as saídas desejadas e calculamos o erro, usando as equações 4.11 e 4.13.

O processamento reverso é executado no sentido contrário e neste caso, os sinais de erro são propagados da saída para a entrada, camada a camada, através dos cálculos dos δ 's (gradientes locais) para cada neurônio da rede. Para os neurônios de saída, δ é calculado usando a equação 4.24 e os respectivos pesos são ajustados usando a equação 4.26. Com os δ 's da camada de saída calculados, usamos a equação 4.36 para calcular os δ 's dos neurônios localizados na camada imediatamente anterior à camada de saída; os pesos destes neurônios são então atualizados segundo a equação 4.26. Continuamos este procedimento camada a camada, até que a camada de entrada seja atingida (Iyoda, 2000).

Observando as equações 4.24 e 4.36, vemos que precisamos conhecer a derivada da função de ativação $f()$ para calcular os gradientes locais δ . Uma das funções mais utilizadas é a função sigmoidal, definida na equação 4.9, onde para o neurônio j , teremos

$$y_j(n) = f_j(v_j(n)) = \frac{1}{1 + e^{-v_j(n)}} \quad (4.38)$$

Uma característica interessante da função sigmoidal é que sua derivada pode ser expressa em termos da ativação $y_j(\mathbf{n})$, a saber:

$$f_j'(v_j(n)) = y_j(n)[1 - y_j(n)] \quad (4.39)$$

A função sigmoidal é limitada, assumindo valores no intervalo $[0, 1]$. Na grande maioria dos problemas, é desejável termos ativações dos neurônios no intervalo $[-1, 1]$. Uma função muito utilizada nestes casos é a tangente hiperbólica, cuja derivada é:

$$f_j'(v_j(n)) = \frac{b}{a} [a - y_j(n)] [a + y_j(n)] \quad (4.40)$$

onde a e b são constantes de escalamento (Haykin, 1999).

O algoritmo de implementação do *backpropagation* é dado no anexo 2. Lembramos que o primeiro passo do algoritmo de retropropagação é, obviamente, a inicialização dos pesos da rede, o que corresponde à definição de um ponto inicial da superfície de erro. Uma boa escolha inicial dos pesos é fundamental para um bom desempenho do algoritmo de retropropagação. Uma inicialização inadequada (ponto inicial mal localizado) pode fazer com que o algoritmo de treinamento fique preso em um mínimo local ou apresente problemas numéricos que de outra forma poderiam ser evitados. Como usualmente não temos nenhuma informação que possa ser diretamente empregada na inicialização dos pesos da rede, um dos métodos mais utilizado é inicializar os pesos aleatoriamente, com distribuição uniforme sobre um pequeno intervalo em torno do zero. Outros métodos foram propostos na literatura, usando informações dos dados de treinamento, que são utilizadas para inicializar os pesos da rede, de forma que, os neurônios estejam operando inicialmente nas proximidades de sua região mais linear. Tal procedimento tem demonstrado resultados superiores aos propostos anteriormente (De Castro, 1998b)(De Castro, 1998c).

4.7.2 – Critérios de Parada

Seja o vetor de pesos $\mathbf{w}^* \in \mathcal{R}^P$, onde P é o número de pesos ajustáveis da rede neural, um mínimo local ou global da superfície de erro definida no espaço de pesos, $\varepsilon : \mathcal{R}^P \rightarrow \mathcal{R}$. Uma condição necessária para \mathbf{w}^* ser um mínimo é que o vetor gradiente $\nabla \varepsilon$ da superfície de erro em relação a \mathbf{w} se anule em $\mathbf{w} = \mathbf{w}^*$ (Iyoda, 2000).

Haykin, 1999, sugere três possíveis critérios de parada para o algoritmo de retropropagação:

1. O algoritmo de retropropagação converge quando a norma euclidiana do gradiente do erro e $\|\nabla \varepsilon\|$ for inferior a um limiar pré-especificado e arbitrariamente pequeno.

2. O algoritmo de retropropagação converge quando a taxa de variação absoluta no erro quadrático médio por época for suficientemente pequena.
3. O algoritmo de retropropagação converge quando $||\nabla \epsilon|| \leq \xi$, onde ξ é um limiar suficientemente pequeno, ou quando $\epsilon_{av}(\mathbf{w}) \leq \tau$, onde τ é um limiar suficientemente pequeno.

4.7.3 – Métodos de Segunda Ordem

O treinamento de redes neurais multicamadas é um problema de otimização não linear de uma função de custo, que mede o erro quadrático médio produzido pela saída da rede neural comparada à uma saída desejada. Podemos encontrar na literatura vários métodos para a resolução de problemas de otimização não linear que podem ser aplicados ao problema de treinamento das RNA's, caso se disponha das informações necessárias.

O *backpropagation* provou sua utilidade no treinamento supervisionado de redes multicamadas para aplicação a muitos problemas de classificação e mapeamento estático de funções não lineares. Este algoritmo de treinamento é uma implementação de um método de otimização conhecido como método do gradiente. No método do gradiente, o vetor de pesos é ajustado na direção oposta ao do vetor gradiente. Tal método pode ser classificado como um método de primeira ordem, já que utiliza apenas a informação do gradiente da função de erro para ajustar os pesos da rede. Os métodos de primeira ordem são conhecidos por serem ineficientes no tratamento de problemas de larga escala, pois apresentam taxas de convergência muito pobres, especialmente em regiões próximas a mínimos locais (Iyoda, 2000). Observamos que para vários exemplos de treinamento se gastava bastante tempo na convergência do erro para um valor mínimo desejado, implicando num grande número de épocas. Era comum também, a saturação da rede, cuja convergência estabilizava-se em um patamar ou caía muito lentamente, inviabilizando assim o treinamento para aquele conjunto de padrões dado. Encontramos vários trabalhos na

literatura no sentido de tornar mais rápida a convergência do treinamento, onde podemos citar entre outros: Chen, 1996, propõe um algoritmo de auto-ajuste, através de funções de ativação modificadas; Verma, 1997, propõe um novo modo de treinamento baseado numa combinação de métodos de solução iterativos e diretos, com o uso de transformação inversa para linearização da saída das funções de ativação não lineares, uma solução direta para o método de treinamento da matriz de pesos da camada de saída; e “*gradient descent*”, regra do delta e outras técnicas para o treinamento dos pesos das camadas intermediárias; Gupta, 1999, propõe a utilização de algoritmos genéticos como uma nova alternativa de treinamento e compara-o com o *backpropagation*; e Elsken, 1999, propõe uma estrutura com redes pequenas e com conjuntos pequenos de padrões de treinamento.

Há casos em que a velocidade de aprendizagem é um fator limitante para possibilitar a implementação prática deste tipo de ferramenta computacional no processo de solução de problemas que requerem otimalidade, robustez e rapidez na convergência do processo de ajuste de parâmetros. Mesmo em aplicações onde uma resposta em tempo real não é necessária, a complexidade temporal do algoritmo pode resultar na intratabilidade computacional do problema. Como exemplo, o aumento da complexidade intrínseca dos problemas atuais da área de engenharia tem produzido uma explosão combinatória de vários métodos para a solução de tais problemas. Além disso, dentre estes métodos, é consenso a idéia de que não existe um método que seja o melhor para todos os casos. Sendo assim, muitas soluções encontradas por métodos específicos podem não atender aos requisitos mínimos do problema. Uma forma eficiente de lidar com esta situação é recorrer ao potencial de processamento computacional disponível hoje em dia e passar a operar com métodos que forneçam simultaneamente múltiplos candidatos a solução, dentre os quais se possa escolher o melhor segundo algum critério pré-estabelecido. No caso da solução ser produzida utilizando-se redes neurais artificiais, este procedimento é tão mais viável quanto maior for a velocidade de aprendizagem da rede neural. Por exemplo, um aumento de dez vezes na velocidade de busca de uma solução permite que se encontre dez vezes mais candidatos à solução com o mesmo custo computacional. Nesta classe encontram-se aplicações relacionadas com a modelagem e identificação de sistemas não lineares, previsão de séries temporais e controle de processos adaptativos (De Castro, 1998a).

Diante de tudo que foi analisado, e do fato de que os dados para o treinamento geralmente apresentam grande dimensionalidade, é justificável a utilização de um método de otimização não linear de segunda ordem. Atualmente estes métodos são considerados a maneira mais eficiente de se fazer o treinamento de redes neurais do tipo **MLP**. Estes algoritmos recorrem a um rigor matemático baseado em modelos de otimização não linear irrestrita bem definidos, não apresentando assim um vínculo natural com a inspiração biológica inicialmente proposta para as **RNA's**.

Nos métodos de segunda ordem, além do vetor gradiente da função objetivo, fazemos uso também da matriz hessiana (matriz de derivadas de segunda ordem) da função erro. Alguns destes métodos, ao passo que requerem poucas modificações no algoritmo de retropropagação padrão, resultam em elevados graus de aceleração, além de não mais requererem a escolha de parâmetros críticos da rede neural como a taxa de aprendizagem e o coeficiente de momento.

Encontramos atualmente na literatura referente à otimização não linear, uma classe de algoritmos de segunda ordem denominados algoritmos de gradiente conjugado. Ela é apontada como apropriada para resolução de problemas de larga escala. Apesar de notadamente superiores aos métodos de primeira ordem, os métodos de segunda ordem também apresentam desvantagens, sendo a principal delas o alto custo computacional associado ao cálculo e armazenamento da matriz hessiana.

Dentre estes métodos podemos destacar o *Gradiente Conjugado Escalonado Modificado (SCGM)*, pois este algoritmo apresenta as seguintes vantagens em relação a outros algoritmos de segunda ordem: não possui nenhum parâmetro crítico dependente de definição por parte do usuário (taxa de aprendizagem ou coeficiente de momento) e nenhum procedimento de busca unidimensional é necessário. Além disso, no cálculo exato da informação de segunda ordem, o custo computacional associado é da mesma ordem que o exigido para o cálculo da informação de primeira ordem.

No trabalho realizado por De Castro, 1998a, foram feitas várias comparações efetivas entre diversos métodos de primeira e segunda ordem aplicados no treinamento de redes neurais artificiais, onde concluiu-se que o SCGM apresenta uma velocidade de convergência bastante

alta, mesmo quando aplicado a conjuntos de treinamento de dimensões elevadas. Toda a formulação matemática para a implementação do SCGM encontra-se de forma clara e detalhada na referida tese. Em outro trabalho realizado por De Castro, 1998d, podemos encontrar o algoritmo em linguagem MATLAB, com todas informações necessárias à sua implementação.

Capítulo 5

Identificação e Controle de Sistemas Usando Redes Neurais

5.1 - Introdução

Neste capítulo será mostrado como o paradigma das redes neurais pode ser aplicado na identificação e controle de sistemas dinâmicos lineares e não lineares. Será apresentada inicialmente uma visão geral do processo de identificação de sistemas, utilizando as redes neurais artificiais (RNA's) para prever as características do sistema real, ou seja, para representar a dinâmica do mesmo. No caso das suspensões automotivas é utilizado um modelo chamado “neuro-veículo” para esta representação, sendo posteriormente aplicado no treinamento do neuro-controlador proposto.

Para o controle do sistema de suspensão ativa de um veículo, onde foram consideradas suas características não lineares, é proposto um neuro-controlador “baseado em um modelo de referência”, cujo objetivo final é conduzir a saída de um sistema a uma dada condição imposta por tal modelo de referência, utilizando as redes neurais para que seja feito um processo de aproximação da dinâmica inversa do sistema, obtendo assim a lei de controle que atuará sobre o mesmo.

5.2 – Redes Neurais na Identificação de Sistemas Dinâmicos

Durante os últimos anos, numerosos estudos têm sido feitos na área de identificação e controle de sistemas físicos dinâmicos, devido principalmente, aos grandes avanços alcançados na micro eletrônica e da popularização da informática atualmente. Basicamente, o procedimento envolve quatro passos chave incluindo a modelagem do sistema, a identificação, o projeto dos controladores e os testes de verificação.

Identificação é o processo de desenvolver ou aperfeiçoar uma representação matemática de um sistema físico usando dados experimentais. O objetivo é a construção de um modelo matemático que caracterize a dinâmica do sistema (Juang, 1994).

Podemos citar seis passos que devem ser seguidos na tarefa de identificação de um sistema dinâmico: desenvolver um modelo analítico do sistema; estabelecer os níveis de resposta dinâmica que provavelmente ocorrerão, usando o modelo analítico e as características das fontes de excitação; colocar a instrumentação requerida para o levantamento das medidas necessárias; realizar os experimentos e gravar os dados; aplicar as técnicas de identificação para o levantamento das características do sistema; e refinar o modelo analítico se necessário (Juang, 1994).

Nas três últimas décadas foram feitos grandes avanços nas áreas de identificação e controle de sistemas lineares invariantes no tempo, com parâmetros desconhecidos. A escolha do identificador e do controlador é baseada na bem fundamentada teoria de sistemas lineares. Nos últimos anos houve um grande desenvolvimento das pesquisas visando a identificação e o controle de sistemas não lineares, principalmente devido ao grande desenvolvimento dos sistemas computacionais, sobretudo com o surgimento de novos processadores e *softwares* capazes de resolver cálculos matemáticos complexos, num tempo relativamente pequeno, permitindo assim o desenvolvimento de tais teorias. As redes neurais artificiais (**RNA's**) tem recentemente atraído uma grande atenção da comunidade científica devido a suas habilidades de “reconhecer” muitas classes de funções contínuas não lineares através de pares entrada-saída dentro de uma precisão desejada. Esta versatilidade e capacidade de mapeamento entrada-saída, junto com a habilidade

de processamento paralelo e coletivo, aumentaram as expectativas da aplicação das RNA's para resolver muitos tipos de problemas relacionados à sistemas dinâmicos. Estas aplicações incluem identificação e controle de sistemas não lineares, detecção, armazenamento de informações para tomada de decisões em controle de sistemas, e o projeto de sistemas de controle inteligentes capazes de “sentir” o ambiente ao seu redor, processando informações para reduzir as incertezas, planejando, coordenando, tomando decisões, e gerando os sinais de controle com alto grau de autonomia.

A cada instante as saídas do sistema são funções de suas entradas e da sua natureza dinâmica. Podemos representar a dinâmica do sistema por equações diferenciais ou vetores. Sejam as seguintes equações diferenciais

$$\begin{aligned} \dot{\mathbf{z}}(t) &= \Phi[\mathbf{z}(t), \mathbf{u}(t)] & t \in \mathcal{R}^+ \\ \mathbf{y}(t) &= \Psi[\mathbf{z}(t)] \end{aligned} \quad (5.1)$$

onde $\mathbf{z}(t) \equiv [z_1(t), z_2(t), \dots, z_n(t)]^T$, $\mathbf{u}(t) \equiv [u_1(t), u_2(t), \dots, u_p(t)]^T$ e $\mathbf{y}(t) \equiv [y_1(t), y_2(t), \dots, y_m(t)]^T$ representa um sistema com p entradas e m , saídas, que possui ordem n , com $u_i(t)$ representando as entradas, $z_i(t)$ representando as variáveis de estado e $y_i(t)$ representando as saídas do sistema. Φ e Ψ são funções não lineares definidas como $\Phi: \mathcal{R}^n \times \mathcal{R}^p \rightarrow \mathcal{R}^n$ e $\Psi: \mathcal{R}^n \rightarrow \mathcal{R}^m$. O vetor $\mathbf{z}(t)$ define o estado do sistema no tempo t . A equação 5.1 refere-se à representação do sistema no espaço de estado, que já foi visto anteriormente no capítulo 3. Para o tempo discreto temos as seguintes equações

$$\begin{aligned} \mathbf{z}(k+1) &= \Phi[\mathbf{z}(k), \mathbf{u}(k)] \\ \mathbf{y}(k) &= \Psi[\mathbf{z}(k)] \end{aligned} \quad (5.2)$$

onde $\mathbf{u}(\cdot)$, $\mathbf{z}(\cdot)$, e $\mathbf{y}(\cdot)$ são seqüência discretizadas. Para a resolução dos problemas propostos pode-se seguir qualquer uma das abordagens descritas acima (Narendra, 1990).

Se o sistema for linear e invariante no tempo as equações que descrevem seu movimento podem ser expressas pelas equações 3.3 e 3.8, onde \mathbf{A} , \mathbf{B} , e \mathbf{C} são matrizes facilmente obtidas, permitindo assim a aplicação de certos conceitos como controlabilidade, estabilidade e

observabilidade. Caso contrário, se os problemas envolverem equações não lineares da forma da equação 5.1 ou 5.2, onde Φ e Ψ são funções conhecidas, resultam na solução de equações algébricas, para as quais não existem muito métodos de resolução. Quando tais funções, ou as matrizes A , B e C , são desconhecidas deve-se usar a teoria de identificação de sistemas para resolvê-las.

O objetivo da neuro-identificação de sistemas é a representação da dinâmica dos mesmos usando redes neurais. Um modelo chamado “neuro-sistema” pode ser treinado para prever as características do sistema real. Quando o neuro-sistema estiver sujeito a uma dada entrada $u(k)$ idêntica a do sistema original, este produz uma saída $\hat{y}(k)$ a qual aproxima $y(k)$ (saída do sistema original) em um sentido desejado (Figura 5.1). A neuro-identificação é portanto um problema de otimização para o qual os pesos sinápticos da rede neural (parâmetros da rede) são determinados assim que a seguinte função custo (erro quadrático médio, equação 4.14) for minimizada

$$J = \frac{1}{2N} \sum_{k=1}^N (\hat{y}(k) - y(k))^T (\hat{y}(k) - y(k)) \quad (5.3)$$

onde N é o número total de padrões entrada-saída, utilizados no treinamento da rede.

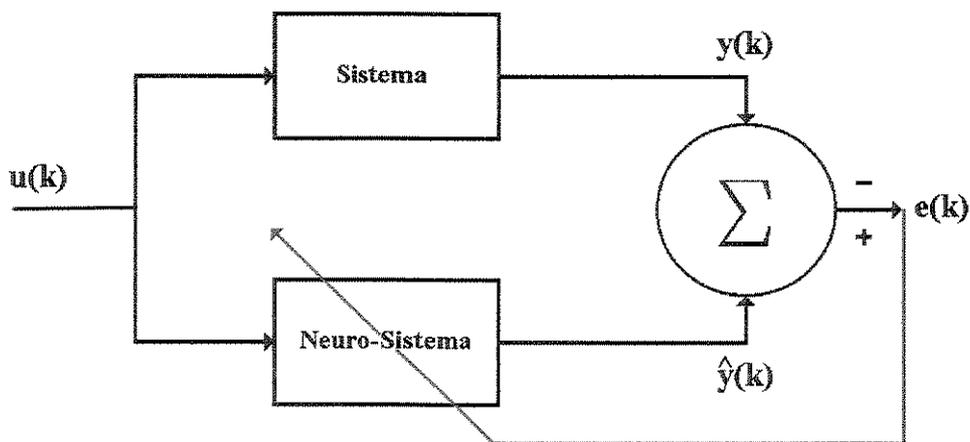


Figura 5.1 – Diagrama de blocos da identificação de um sistema

O método de treinamento mais comum usado na neuro-identificação é o de retropropagação do erro (*error backpropagation method*) que pode ser realizado de acordo com dois tipos de estratégias de aprendizagem: o aprendizado direto e o aprendizado recorrente (Moran, 1994). A diferença básica entre ambos os métodos está na relação dinâmica entre as entradas e as saídas da rede. Enquanto no método de aprendizagem direto as entradas e as saídas da rede não tem um relacionamento temporal, no método recorrente, a saída da rede torna-se sua entrada no próximo intervalo de tempo (no próximo passo). Enquanto o primeiro método considera basicamente uma rede neural multicamadas (**MLP**) padrão (*rede feedforward*), com as entradas (vetor de estado) vindo da saída do sistema real, o segundo método considera uma rede recorrente com conexões de retropropagação entre os neurônios da camada de entrada e de saída. Considerando que as redes *feedforward* citadas anteriormente, apenas realizam um mapeamento estático entre as entradas e as saídas, as redes recorrentes possuem memória, tendo dinâmica e podem representar sequencialmente as características dinâmicas do sistema real (Narendra, 1990). As redes *feedforward* podem ser diretamente treinadas pelo algoritmo convencional do *backpropagation* estático, descrito no capítulo 4, segundo estratégias de implementação (aquisição de dados e treinamento) *on-line* ou *off-line*. Já as redes recorrentes podem ser treinadas pelo *backpropagation* dinâmico (**DBP**) (Narendra, 1990) ou através do *backpropagation* através do tempo (**BPTT**) (Werbos, 1990).

Usadas como sinais de referência para o treinamento da rede, as saídas do sistema trabalham bem em ambas estratégias de aprendizagem (direta ou recorrente). Embora possam ser medidas sem ruído, o método de aprendizagem recorrente identifica melhor o sistema, quando os sinais de referência estão contaminados com ruído. Enquanto as redes recorrentes (se bem treinadas) podem substituir totalmente o sistema real, as redes *feedforward* devem sempre atuar em paralelo com o sistema real (e sensores de ruído) os quais fornecem o vetor de estado de entrada da rede (Moran, 1994).

5.3 – Neuro-Veículo

Um veículo é um sistema muito complexo, possui várias entradas, incertezas e distúrbios, que devem ser levados em conta no projeto do controle das características dinâmicas do mesmo. Quando desejamos identificar um sistema de um veículo, vários problemas são encontrados, como por exemplo: não é fácil selecionar uma classe de modelo apropriada devido ao complexo relacionamento entre as entradas, incluindo as excitações, e saídas; o grau elevado de não linearidades dos sistemas de um veículo; e a existência de incertezas paramétricas internas e externas (Figura 5.2). Técnicas alternativas baseadas no uso das RNA's podem ser empregadas. Estas podem ser usadas na identificação e controle dos sistemas de um veículos de várias maneiras: para representar as características não lineares de um pneu, “neuro-pneu”; para representar as características de um sistema de suspensão não linear, “neuro-suspensão” ou “neuro-veículo”; para representar o sistema de controle de uma suspensão, “neuro-controlador da suspensão”; e assim segue podendo ser aplicadas ainda para o sistema de direção, controle de tração, sistemas de injeção de combustível, transmissão e sistemas de freios (El-Gindy, 1993), (Zadeh, 1997).

Podemos citar brevemente algumas características das RNA's que são de grande utilidade para o estudo de suspensões automotivas:

- Não linearidade: A rede neural é essencialmente um sistema não linear. Esta propriedade é extremamente importante, pois o modelamento realístico de uma suspensão automotiva é uma tarefa que envolve equações não lineares, por vezes de difícil ou impossível determinação. Os sub-sistemas de uma suspensão, como amortecedores ou uma mola, são comportamento não linear;

- Adaptatividade: As redes neurais têm a capacidade intrínseca de se adaptar para representar mudanças no ambiente que as envolve. Essencialmente, uma rede treinada em um ambiente específico, pode ser retreinada para ser mais sensível a mudanças neste ambiente, ou seja, a rede será capaz de se adaptar as variações dos parâmetros do sistema, o que é muito importante no caso de suspensões automotivas, pois seus componentes, como foi citado no item

anterior, apresentam características altamente não lineares e os seus parâmetros podem não apresentar valores fixos. Ou ainda, uma rede pode ser montada de forma a mudar os seus próprios pesos sinápticos de forma a se adaptar automaticamente a mudanças no sistema em que está acoplada (redes recorrentes);

- Mapeamento entrada-saída: Consiste no processo de aprendizado da rede a partir de um conjunto de sinais (estímulos entrada-saída), visando a minimização de uma dada função custo. Especialmente útil para casos em que o processamento exato (demorado) pode ser feito *a priori* e a rede pode ser treinada para reproduzir estes sinais (de forma mais rápida);

- Uniformidade para analisar e projetar: Neurônios são comuns a todas as redes neurais. Redes neurais são facilmente montadas a partir da integração de módulos. As duas últimas características são importantes quando se quer montar uma rede para resolver um problema específico, pois as diversas arquiteturas de redes neurais possíveis só diferem na maneira com que seus neurônios estão agrupados (Picado, 1998).

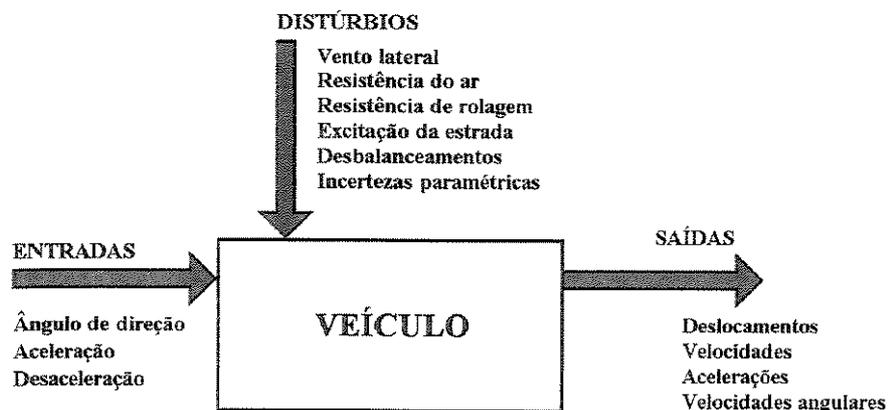


Figura 5.2 – Diagrama de blocos do sistema de um veículo

Aplicaremos as definições do item anterior no modelo de suspensão passiva e ativa de um automóvel de passeio, cuja formulação matemática foi descrita anteriormente no capítulo 3. A rede neural representará (identificará) a dinâmica da suspensão do veículo, tendo a capacidade de prever o vetor de estado futuro do sistema, e após realizado seu treinamento, quando as

características da suspensão estiverem armazenadas nos pesos sinápticos, será usada para treinar o neuro-controlador fazendo a retropropagação do erro e , conseqüentemente, permitindo a obtenção da lei de controle segundo o modelo de controlador proposto. Na figura 5.3 temos o diagrama de blocos do treinamento do neuro-veículo. Podemos observar também, o sinal de controle que está representado por $u(k)$, a excitação (rugosidade da estrada, buracos, lombadas, etc.) representado por $r(k)$, o vetor de estado $z(k+1)$ definido pela equação 3.6 e o vetor de estado $\hat{z}(k+1)$ estimado pela rede neural. Na figura 5.3 o “*tapped delay operator*” Δ é definido como um atraso da saída do sistema $z(k)$ (vetor de estado) que será usada como entrada do neuro-veículo. Adotaremos o método de treinamento direto proposto por (Moran, 1994) e (Park, 1996), pois sua implementação é mais simples que o método recorrente e atende nossos objetivos em relação à convergência, tempo de processamento e precisão desejada. Também faremos as simulações desconsiderando a presença de ruído, pois, caso estivéssemos considerando sua presença seria mais indicado o uso das redes recorrentes citadas anteriormente.

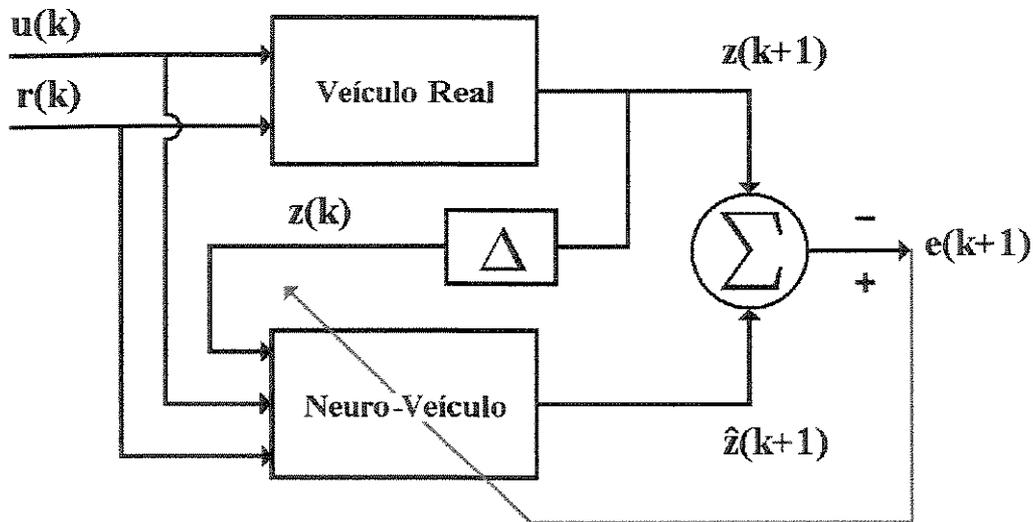


Figura 5.3 – Diagrama de blocos do treinamento do neuro-veículo

Na presente análise consideramos que antes de se projetar o neuro-controlador, uma rede neural representando a dinâmica do sistema a ser controlado (neuro-veículo) deve ser treinada. A precisão com a qual este neuro-veículo aproxima a dinâmica do veículo real, tem grande

influência no cálculo das requeridas derivadas parciais para treinar o neuro-controlador, como veremos a seguir, logo, justifica-se utilizarmos na identificação um algoritmo cuja performance seja comprovadamente superior ao *backpropagation*, além de permitir trabalhar com um grande número de padrões de treinamento. Usaremos portanto o **SCGM** no treinamento do neuro-veículo, para a obtenção dos pesos sinápticos do mesmo, até que a seguinte função custo seja minimizada (erro quadrático médio):

$$J = \frac{1}{2N} \sum_{k=1}^N (z(k+1) - \hat{z}(k+1))^2 \quad (5.4)$$

5.4 – Redes Neurais em Sistemas de Controle

Ultimamente tem sido desenvolvidas alguma pesquisas na aplicação de redes neurais em sistemas de controle e automação. Isso tem possibilitado novas técnicas que se inspiram na obtenção de melhores e mais eficientes ações de controle através do paradigma das redes neurais. Essas novas técnicas tentam aliar os benefícios das técnicas de controle ótimo e controle adaptativo.

De uma forma geral, o problema de controle de um processo dinâmico pode ser esboçado como na figura 5.4, sendo este esquema chamado de sistema de controle com realimentação das variáveis de estado (Cerqueira, 1996).

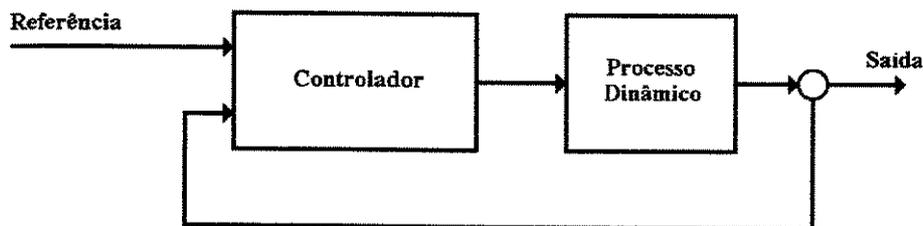


Figura 5.4 – Sistema de controle genérico

Um processo dinâmico ou sistema dinâmico é um conjunto de elementos coordenados entre si, e que formam uma estrutura organizada como se fosse um sistema único. Ele pode ser visto com mais detalhes na figura 5.5. Ele é basicamente composto por:

Sistema de transformação – Responsável por fazer operações de transformações quaisquer. Possui como entradas os sinais de saída dos acionadores e variáveis externas de perturbação, que podem ser classificadas em mensuráveis e não mensuráveis. Os efeitos das transformações são refletidos nas chamadas variáveis de saída. Os sistemas de transformação apresentam comportamento não lineares e algumas vezes, apesar de terem os seus modelos matemáticos conhecidos teoricamente, apresentam parâmetros com valores desconhecidos ou pouco confiáveis e/ou variantes. Porém, muitas vezes, é possível fazer-se linearizações e utilizar o modelo em questão para o projeto das estratégias de controle do sistema.

Sensores – Responsáveis pela medição das variáveis de saída bem como as externas e informá-las ao controlador. Em sistemas físicos reais os sensores apresentam um ligeiro atraso na medição, não linearidades e sujeição a ruídos, mesmo nos casos de sensores projetados com base em tecnologias avançadas. No entanto, em geral é possível desprezar-se esses efeitos e considerar os sinais apresentados pelos sensores ao controlador em um dado instante de tempo, como sendo os valores reais das variáveis de saída.

Acionadores – Responsável pela tarefa de receber o sinal de saída do controlador e prover a potência necessária à operação do sistema de transformação. Esta tarefa também pode ser feita pela modificação ou atuação em variáveis de entrada do processo dinâmico. Assim como os sensores, os acionadores possuem problemas de atraso de tempo, não linearidades e sujeição a ruídos e distorções de sinais de baixa potência, gerados pelo controlador (Cerqueira, 1996).



Figura 5.5 – Processo dinâmico

Para efeito de projeto de um sistema de controle, são considerados em conjunto os modelos matemáticos dos sistemas de transformação, dos sensores e dos acionadores. Assim, teremos um único modelo o que facilita o estudo de todo o processo dinâmico. O controlador é o elemento responsável em fazer com que as variáveis de saída do sistema tendam ou rastreem os valores desejados para os mesmos, atendendo também à especificações de desempenho que estejam preestabelecidas.

Usualmente o projeto de controladores tradicionais envolvem análises matemáticas complexas e possuem muita dificuldades em controle de sistemas altamente não lineares. Para superar estas dificuldades, um número cada vez maior de trabalhos usando redes neurais para controle de sistemas tem sido desenvolvidos nos últimos anos. O uso da capacidade de aprendizado das redes neurais tem ajudado no projeto de controladores mais flexíveis, especialmente quando a dinâmica do sistema é complexa e altamente não linear. Tomando diretamente o problema de controle não linear, vários estudos experimentais e simulações tem mostrado que o mapeamento entrada-saída e a capacidade de aproximação de funções não lineares das redes neurais podem ser usadas para projetar controladores que podem ser baseados na dinâmica inversa do sistema à controlar (Pham, 1999) (Moran, 1994), ou que sejam capazes de identificar o dinâmica de um modelo de referência (Moran, 1993), (Choromanski, 1996), (Park, 1996), (Kasparian, 1998), ou ainda, controladores capazes de otimizar uma pré-especificada performance medida do sistema (Moran, 1997), (Watanabe, 1999). Estas habilidades (capacidade de aprendizado e de processamento paralelo (MacLoone, 1997)) são úteis para o projeto de controladores auto-ajustáveis, permitindo adaptações às mudanças de hardware ou das funções de custo.

5.5 – Neuro-Controlador

Como foi citado anteriormente, os métodos para o projeto de neuro-controladores podem ser classificados dentro de três grupos gerais: neuro-controladores treinados para identificar a dinâmica de um modelo de referência (*model reference neuro-controllers*); neuro controladores representando a dinâmica inversa do sistema a ser controlado; e neuro controladores treinados para otimizar o desempenho do sistema (Moran, 1994).

O controlador proposto neste trabalho pertence ao primeiro grupo, que possuem a tarefa de conduzir um sistema de acordo com uma resposta desejada. O projeto dos controladores baseados em um modelo de referência pode ser visto como um problema de identificação dinâmica onde o objetivo é treinar o neuro-controlador para que o sistema em “*loop-fechado*” siga as características de um modelo de referência com uma performance já conhecida e desejada. O controlador é projetado para fazer o mapeamento entre o espaço de estado e os sinais de controle. As trajetórias do sistema de referência são as entradas do controlador e as saídas são os controles desejados. O controlador pode ser treinado *off-line* com trajetórias nominais, seja, através de simulações ou através do desempenho do sistema real. Caso não exista um conjunto de padrões entrada-saída entre o vetor de estado e os sinais do atuador, podemos usar como alternativa para este problema o uso combinado do controlador com uma rede *feedforward* que represente a dinâmica do sistema, ou seja, o neuro-veículo é necessário para se obter uma estimativa da sensibilidade do sistema com relação às suas entradas, pois em muitos casos práticos a dinâmica do sistema pode não estar disponível para o projeto do controlador. Algum tipo de aprendizado supervisionado pode ser usado como algoritmo de treinamento (Balakrishnan, 1996) (Kasparian, 1998).

Os mesmos métodos de treinamento apresentados anteriormente para a neuro-identificação do sistema podem ser usados para treinar o neuro-controlador. Neste caso o modelo do neuro-veículo proposto na figura 5.3 é substituído por uma rede aumentada composta pelo neuro-controlador e um neuro-veículo como mostrado na figura 5.6, onde \mathbf{z}_r representa a saída do modelo de referência que deve ser igualada pela saída do sistema real, representado pelo neuro-veículo $\hat{\mathbf{z}}$ (Park, 1996), (Moran, 1994). Neste modelo também observa-se que a saída do modelo

de referência foi atrasada para servir de entrada para o neuro-controlador e para o neuro-veículo. Durante o treinamento do neuro-controlador, o neuro-veículo já treinado permanece fixo (seus pesos sinápticos já treinados não se alteram) e é usado apenas para retropropagar o erro da saída do neuro-veículo para o neuro-controlador, treinando assim os pesos sinápticos do mesmo.

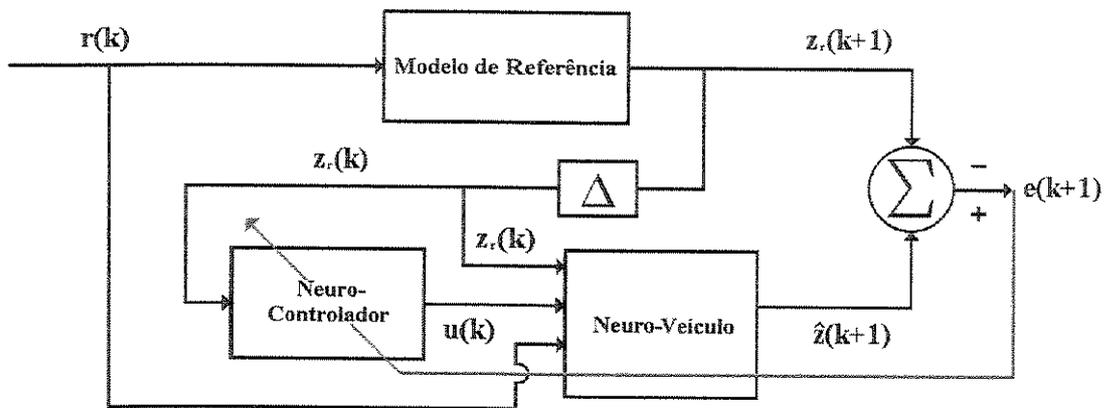


Figura 5.6 – Diagrama de blocos do treinamento do neuro-controlador

Em (Park, 1996), este tipo de controlador é denominado como *Feedforward Neuro-Controller*, ou simplesmente **FFNC**. Um novo conceito de mapeamento inverso foi desenvolvido para o projeto do neuro-controlador proposto, usando como foi dito anteriormente um neuro-veículo para retropropagar o erro durante a atualização dos pesos sinápticos da rede que representa controlador. O treinamento do controlador **FFNC** pode ser entendido então, como um processo de aproximação da dinâmica inversa do sistema.

Na figura 5.7 temos uma representação de duas redes neurais colocadas em série representando o neuro-controlador e o neuro-veículo. As redes são do tipo **MLP**, sendo que a primeira rede (neuro-controlador) possui duas camadas intermediárias e a segunda rede (neuro-veículo) possui apenas uma camada intermediária (Moran, 1994). A rede que representa o neuro-controlador, como um mapeamento inverso do sistema pode ser desenvolvido através do uso do neuro-veículo como mostrado na figura 5.7, isto é

$$\hat{z} = F(z_r, u, r, w) \quad (5.5)$$

$$u = G(z_r, w) \quad (5.6)$$

onde \mathbf{u} é a entrada de controle e $\hat{\mathbf{z}}$ é a saída do neuro-veículo. O treinamento do neuro-controlador é realizado para ajustar os seus pesos sinápticos, até que a saída do neuro-veículo se aproxime de uma dada saída de referência \mathbf{z}_r e quando o treinamento estiver terminado, \mathbf{u} se aproximará de um \mathbf{u}_r , relativo ao modelo de referência adotado.

Para o entendimento de como o erro é retropropagado da saída do neuro-veículo para atualizar os pesos do neuro-controlador faremos o uso do modelo da figura 5.7 abaixo:

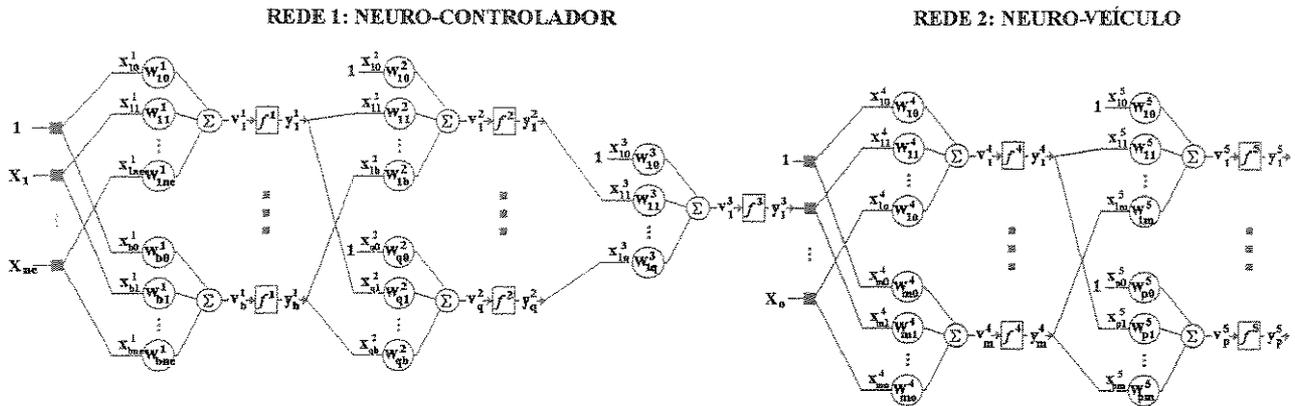


Figura 5.7 – Rede neural: Neuro-Controlador + Neuro-Veículo

O objetivo do treinamento desta rede é a atualização dos pesos sinápticos do neuro-controlador até que a função custo definida pela equação 5.4, substituindo o vetor de estado do veículo real $\mathbf{z}(\mathbf{k}+1)$, pelo vetor de estado do modelo de referência $\mathbf{z}_r(\mathbf{k}+1)$. Primeiramente vamos considerar que os pesos sinápticos do neuro-veículo foram treinados pelo método de aprendizagem direto descrito anteriormente. Sejam \mathbf{W}_{NV1} e \mathbf{W}_{NV2} as matrizes de pesos do neuro-veículo das camadas intermediária e de saída respectivamente, onde teremos

$$\mathbf{W}_{NV2} = \mathbf{w}_{ij}^5 = \begin{bmatrix} w_{10}^5 & w_{20}^5 & w_{30}^5 & \dots & w_{p0}^5 \\ w_{11}^5 & w_{21}^5 & w_{31}^5 & \dots & w_{p1}^5 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ w_{1m}^5 & w_{2m}^5 & w_{3m}^5 & \dots & w_{pm}^5 \end{bmatrix}_{(m+1,p)} \quad (5.7)$$

onde $i = 1, 2, \dots, p$ e $j = 0, 1, 2, \dots, m$; sendo p o número de neurônios da camada de saída e m o número de neurônios da camada intermediária do neuro-veículo.

$$w_{NV1} = w^4_{jl} = \begin{bmatrix} w^4_{10} & w^4_{20} & w^4_{30} & \dots & w^4_{m0} \\ w^4_{11} & w^4_{21} & w^4_{31} & \dots & w^4_{m1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ w^4_{1o} & w^4_{2o} & w^4_{3o} & \dots & w^4_{mo} \end{bmatrix}_{(o+1,m)} \quad (5.8)$$

onde $j = 1, 2, \dots, m$ e $l = 0, 1, 2, \dots, o$; sendo o número de entradas do neuro-veículo.

Usando a equação 4.11 podemos determinar o sinal de erro, onde teremos

$$e_i = e^5_i = z_r - \hat{z} \quad (5.9)$$

Antes do cálculo do gradiente local do neurônio da camada de saída do neuro-controlador, vamos calcular o gradiente local dos neurônios da camada de saída e da camada intermediária do neuro-veículo.

Seja $i = 1, 2, \dots, p$; como foi definido anteriormente, usaremos a equação 4.24 para determinarmos o gradiente local $\delta_i = \delta^5_i$ dos neurônio da camada de saída do neuro-veículo.

$$\delta_i = \delta^5_i = e^5_i f^5_i'(v_i^5) \quad (5.10)$$

Para a camada intermediária do neuro-veículo fazendo $j = 1, 2, \dots, m$; usaremos a equação 4.36 para o cálculo de gradiente local $\delta_j = \delta^4_j$ dos neurônio da mesma.

$$\begin{aligned} \delta_j = \delta^4_j &= f^4_j'(v_j^4) \sum_{i=1}^p [\delta^5_i w^5_{ij}] \\ &= f^4_j'(v_j^4) \sum_{i=1}^p [e^5_i f^5_i'(v_i^5) w^5_{ij}] \end{aligned} \quad (5.11)$$

Podemos agora calcular o gradiente local $\delta_1 = \delta^3_1$ do neurônio da camada de saída no neuro-controlador Usaremos a equação 4.36, pois devemos considerar o neurônio da camada de saída do neuro-controlador como um neurônio de uma camada intermediária da rede formada pelo neuro-veículo e pelo neuro-controlador, onde teremos

$$\begin{aligned} \delta_1 = \delta^3_1 &= f^3_1'(v_1^3) \sum_{j=1}^m [\delta^4_j w^4_{j1}] \\ &= f^3_1'(v_1^3) \sum_{j=1}^m [f^4_j'(v_j^4) \sum_{i=1}^p [e^5_i f^5_i'(v_i^5) w^5_{ij}] w^4_{j1}] \end{aligned} \quad (5.12)$$

onde w^4_{j1} corresponde à segunda linha da matriz w_{NV1} do neuro-veículo.

Para a atualização da matriz de pesos da camada de saída do neuro-controlador w_{NC3} , usaremos a equação 4.25. Seja $r = 0, 1, 2, \dots, q$; onde q é o número de neurônios da 1ª camada intermediária do neuro-controlador, teremos então

$$w_{NC3} = w^3_{1r} = \begin{bmatrix} w^3_{10} \\ w^3_{11} \\ \vdots \\ w^3_{1q} \end{bmatrix}_{(q+1,1)} \quad (5.13)$$

$$\frac{\partial \varepsilon}{\partial w_{NC3}} = \frac{\partial \varepsilon}{\partial w^3_{1r}} = -\delta^3_1 y^2_r \quad (5.14)$$

Observando a figura 4.6 e a figura 5.7, teremos

$$y^2_r = X^3_r \quad (5.15)$$

onde y_0^2 corresponde ao "bias" e $w_{NC3} = w_{1r}^3$ é uma matriz $(q+1, 1)$.

Substituindo a equação 5.15 na 5.14, teremos

$$\frac{\partial \varepsilon}{\partial w_{NC3}} = \frac{\partial \varepsilon}{\partial w_{1r}^3} = -\delta^3 \mathbf{X}^3_r \quad (5.16)$$

Usaremos o mesmo procedimento descrito anteriormente para o cálculo do gradiente local $\delta_r = \delta^2_r$ dos neurônios da 1ª camada intermediária do neuro-controlador. Seja $r = 1, 2, \dots, q$, teremos então

$$\begin{aligned} \delta_r = \delta^2_r &= f^2_r'(v_r^2) \sum_1 [\delta^3_1 w^3_{1r}] \\ &= f^2_r'(v_r^2) \sum_1 [f^3_1'(v_1^3) \sum_{j=1}^m [f^4_j'(v_j^4) \sum_{i=1}^p [e^5_i f^5_i'(v_i^5) w^5_{ij}] w^4_{j1}] w^3_{1r}] \end{aligned} \quad (5.17)$$

Para a atualização da matriz de pesos da 1ª camada intermediária do neuro-controlador w_{NC2} , onde $a = 0, 1, 2, \dots, b$; sendo b o número de neurônios da 2ª camada intermediária do neuro-controlador. Teremos então

$$w_{NC2} = w^2_{ra} = \begin{bmatrix} w^2_{10} & w^2_{20} & w^2_{30} & \dots & w^2_{q0} \\ w^2_{11} & w^2_{21} & w^2_{31} & \dots & w^2_{q1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ w^2_{1b} & w^2_{2b} & w^2_{3b} & \dots & w^2_{qb} \end{bmatrix}_{(b+1,q)} \quad (5.18)$$

$$\frac{\partial \varepsilon}{\partial w_{NC2}} = \frac{\partial \varepsilon}{\partial w^2_{ra}} = -\delta^2_r \mathbf{X}^2_{ra} \quad (5.19)$$

Para o cálculo do gradiente local $\delta_a = \delta_a^1$ dos neurônios da 2ª camada intermediária do neuro-controlador. Seja $a = 1, 2, \dots, b$, teremos então

$$\begin{aligned} \delta_a &= \delta_{ra}^1 = f_{a'}^1(v_a^1) \sum_{r=1}^q [\delta_r^2 w_{ra}^2] = \\ &= f_{a'}^1(v_a^1) \sum_{r=1}^q [f_{r'}^2(v_r^2) \sum_1 [f_{i'}^3(v_i^3) \sum_{j=1}^m [f_{j'}^4(v_j^4) \sum_{i=1}^p [e_i^5 f_{i'}^5(v_i^5) w_{ij}^5] w_{j1}^4] w_{1r}^3] w_{ra}^2] \end{aligned} \quad (5.20)$$

Para a atualização da matriz de pesos da 2ª camada intermediária do neuro-controlador w_{NC1} , onde $s = 0, 1, 2, \dots, ne$; sendo ne o número de neurônios da camada de entrada do neuro-controlador. Teremos então

$$w_{NC1} = w_{as}^1 = \begin{bmatrix} w_{10}^1 & w_{20}^1 & w_{30}^1 & \dots & w_{b0}^1 \\ w_{11}^1 & w_{21}^1 & w_{31}^1 & \dots & w_{b1}^1 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ w_{1ne}^1 & w_{2ne}^1 & w_{3ne}^1 & \dots & w_{bne}^1 \end{bmatrix}_{(ne+1,b)} \quad (5.21)$$

$$\frac{\partial \varepsilon}{\partial w_{NC1}} = \frac{\partial \varepsilon}{\partial w_{as}^1} = -\delta_a^1 X_{as}^1 \quad (5.22)$$

Mais detalhes sobre o algoritmo de treinamento dos pesos do neuro-controlador, assim como a figura 5.7 ampliada, encontram-se nos anexos 3 e 4.

Capítulo 6

Simulações e Resultados

6.1 - Introdução

Este capítulo apresenta as simulações computacionais dos modelos propostos anteriormente no capítulo 3, empregando a metodologia apresentada e desenvolvida nos capítulos 4 e 5. Foi analisado também um novo modelo proposto, com a substituição da mola linear do modelo original, por outra, com características não lineares, visando demonstrar a capacidade das redes neurais quando empregada na identificação e no controle de sistemas não lineares.

Primeiramente foi feita a identificação *off-line* da dinâmica de uma suspensão ativa, utilizando uma rede **MLP**, cujo modo de treinamento, número de padrões, arquitetura da rede, algoritmo utilizado, convergência, foram devidamente analisados. Em seguida foi discutida a performance do neuro-controlador proposto, seu treinamento e os testes necessários à verificação de seu desempenho, usando como parâmetros para análise, o espaço de trabalho da suspensão, a aceleração vertical da massa suspensa e a força de contato do pneu com a via, critérios de projeto de um sistema de suspensão automotiva. Destacou-se a capacidade do neuro-controlador em realizar um processo de aproximação da dinâmica inversa do sistema, sua adaptatividade em relação às variações dos parâmetros e a identificação de sistemas não lineares, sem a necessidade do emprego de técnicas de linearização.

6.2 – Considerações Iniciais

Como os sistemas serão simulados utilizando as suas equações de movimento escritas na forma de estado, obteremos assim diretamente do vetor de estado (equação 3.6) todas as variáveis necessárias para o cálculo dos parâmetros que serão analisados posteriormente: espaço de trabalho da suspensão, aceleração da massa suspensa e força de contato do pneu com a via. Como foi visto anteriormente, a escolha destes parâmetros está relacionada com os critérios de projeto das suspensões automotivas.

Nas simulações serão utilizados os valores da tabela 6.1 para simular a resposta do modelo de $\frac{1}{4}$ de veículo proposto segundo uma dada condição inicial. Tais valores se referem a um modelo descrito em Müller e Schiehlen, 1976.

Variável	Valor	Unidade	Descrição
Ms	600	kg	Massa suspensa ($\frac{1}{4}$ de veículo)
mu	40	kg	Massa do sistema eixo-roda ($\frac{1}{4}$ de veículo)
km	15000	kN/m	Constante de rigidez da mola da suspensão
kp	160000	kN/m	Constante de rigidez do pneu
ca	2400	Ns/m	Fator de amortecimento do amortecedor da suspensão

Tabela 6.1 – Valores dos parâmetros usados na simulação.

Para as simulações dos sistemas dinâmicos foram utilizados os recursos computacionais disponíveis. Todos os programas foram elaborados e executados utilizando o programa MATLAB® versão 5.3 e um computador Pentium III de 550 MHz. Dentre estes programas, foi desenvolvido um especificamente para as simulações das suspensões automotivas descritas anteriormente, empregando algumas funções do MATLAB®, como o ODE45 para a resolução de sistemas de equações diferenciais, com o objetivo de calcular e apresentar organizadamente os gráficos das respostas dinâmicas de tais modelos às excitações que serão descritas a seguir.

Podemos ver abaixo na figura 6.1 a janela de entrada de dados para as simulações.

Figura 6.1 – Janela de entrada de dados para as simulações.

6.2.1 –Excitação em degrau:

O degrau é uma das formas mais comuns de excitação para simulação de sistemas de controle do tipo regulador. Constituem uma condição adversa para os algoritmos de integração numérica. Sua expressão é simples e apresenta respostas de rápida identificação e análise quanto a estabilidade do sistema, além de conseguir varrer uma grande faixa de frequências. A sua expressão matemática é simples e dada por,

$$R = \begin{cases} 0 & , t \leq t_0 \\ r & , t > t_0 \end{cases} \quad | \quad t \in R^+ \quad (6.1)$$

Onde r pode assumir qualquer valor constante. Foi usada a excitação em degrau para o treinamento do neuro-veículo e do neuro-controlador. Este sinal será fornecido à rede como uma das entradas para o treinamento. Adotou-se um degrau de amplitude de 10 cm, ou seja, $r = 0.1$ m.

6.2.2 –Excitação tipo pulso:

Este tipo de excitação foi empregada para o teste do treinamento das redes neurais, como também nos resultados da análise das características dos sistemas. No teste do treinamento das RNA's deve-se utilizar parâmetros diferentes daqueles empregados anteriormente (condições iniciais e excitações), a fim de se verificar a capacidade de generalização e adaptatividade das redes neurais, o que caracterizaria um treinamento realizado com sucesso. Sua expressão matemática é dada por

$$R = \begin{cases} 0 & , t \leq t_i \\ r & , t_i < t < t_f \\ 0 & , t \geq t_f \end{cases} \quad \left| \quad t \in R^+ \right. \quad (6.2)$$

Considerando-se o veículo a uma velocidade igual a 7,2 km/h submetido a uma excitação em forma de um pulso de 10 cm de altura e 20 cm de comprimento, tem-se $r = 0.1$ m e $\Delta t = t_f - t_i = 0,1$ s (duração do pulso). A rigor este tipo de excitação pode ser entendida como uma composição de duas funções degrau, onde a segunda é o inverso da primeira (Alves, 1997).

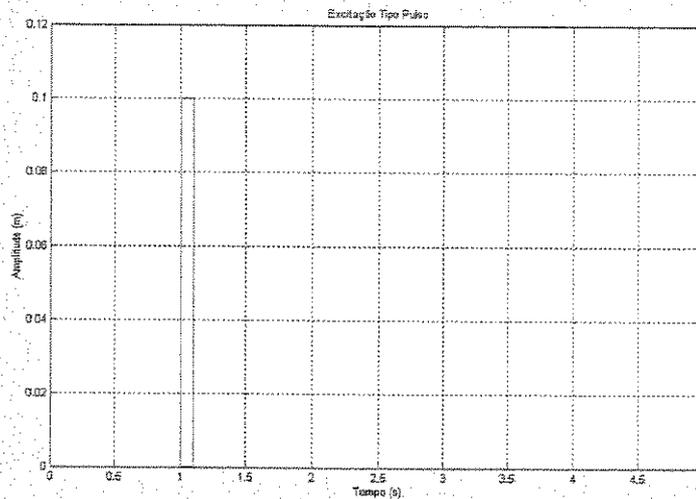


Figure 6.2 – Excitação tipo pulso.

Foi criada uma lombada para o teste do treinamento das redes neurais e para a análise da resposta da suspensão quando submetida a este tipo de excitação. Esta é gerada a partir de um pulso, com 10 cm de altura e 260 cm, servirá de excitação para o modelo que atingirá a mesma à uma velocidade de 7,2 km/h. Tem-se então $r = 0.1$ m, e $\Delta t = 1.3$ s. O objetivo é aplicar outra descontinuidade ao sistema logo após este ter estabilizado.

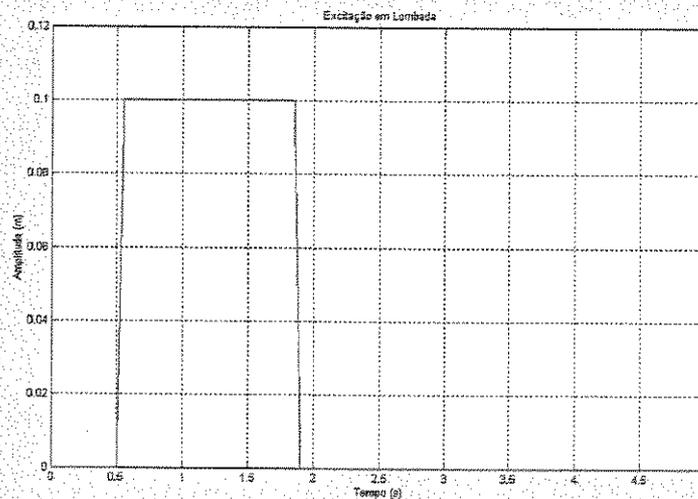


Figure 6.3 – Excitação em lombada.

6.3 – Identificação *Off-Line* da Dinâmica Direta de Uma Suspensão Ativa Veicular (Treinamento do Neuro-Veículo).

Este experimento foi desenvolvido conforme o exposto na seção 5.3. Foi feita a identificação de um modelo dinâmico multivariável para uma suspensão ativa veicular descrita no capítulo 3.

A RNA utilizada é do tipo MLP com uma camada de entrada, uma camada de saída e uma camada intermediária como mostrado na figura 5.7. O número de neurônios nas camadas de entrada e saída são diretamente determinados do número das variáveis de estado, controle e excitação envolvidas no problema específico. O número de neurônios na camada intermediária é escolhido para formar uma rede com uma arquitetura suave evitando assim problemas de convergência lenta no processo de treinamento, afetando negativamente as propriedades de generalização das redes neurais (Moran, 1997). Embora, tenha sido provado matematicamente que uma camada intermediária é suficiente para realizar um mapeamento entrada-saída não linear para aproximação de alguma função contínua, experiências sugerem que o uso de duas camadas intermediárias fornecem maior flexibilidade e convergência para o processo de treinamento das redes (Tamura, 1997). Deve-se lembrar que dependendo do tipo de sistema ou do algoritmo de treinamento empregado, a utilização de uma única camada intermediária tem demonstrado resultados excelentes.

Como foi descrito anteriormente no capítulo 4 e no capítulo 5, o treinamento do neuro-veículo foi realizado de acordo com o método de aprendizado direto, onde o algoritmo de treinamento adotado foi o SCGM (*Gradiente Conjugado Escalonado Modificado*), devido a sua alta velocidade de convergência, se comparado com o BPA (*Backpropagation*), principalmente quando o conjunto de treinamento é formado por uma grande quantidade de padrões. Como foi citado anteriormente, uma boa identificação do sistema, implicará numa maior precisão no treinamento do neuro-controlador, justificando assim o uso de um algoritmo de treinamento mais poderoso, como é o caso do SCGM.

O modelo do neuro-veículo foi dimensionado com 6 neurônios de entrada (4 variáveis de estado, 1 sinal de controle e 1 excitação), 4 neurônios de saída e 20 neurônios na camada intermediária (Moran, 1994). O experimento foi desenvolvido como um procedimento *off-line*, onde o sistema foi excitado com sinais apropriados e colhidos os sinais de saída, obtendo-se assim as matrizes de entrada e de saída desejada, com os padrões usados no treinamento do neuro-veículo. Estes padrões de treinamento são obtidos do histórico de operação do sistema (Park, 1996). O intervalo de amostragem foi de 0.05 segundos e o tempo de duração do experimento foi de 5 segundos. O processo de treinamento foi realizado com um conjunto de 400 padrões entrada-saída para diferentes condições iniciais, que foram escolhidas aleatoriamente. A escolha das condições iniciais deve ser feita dentro de um conjunto de valores específicos para cada variável de estado. O espaço de trabalho da suspensão pode ser limitado a um valor de no máximo 25 cm e a velocidade máxima atingida pelo amortecedor é de 1.5 m/s. Usou-se o intervalo $[-0.125, 0.125]$ para X_1 e X_2 (m) e, $[-0.75, 0.75]$ para \dot{X}_1 e \dot{X}_2 (m/s), a fim de se determinar as condições iniciais usadas no treinamento da rede. O sinal de controle também foi limitado a uma faixa de operação do atuador escolhido, visto que não dispomos da lei de controle empregada, que será posteriormente obtida após o treinamento do neuro-controlador, para a obtenção dos valores necessários à formação do conjunto de padrões de treinamento da rede. Após o treinamento do neuro-veículo este foi testado com excitações e condições iniciais diferentes daquelas usadas no treinamento da rede, lembrando mais uma vez que foi utilizada uma excitação em degrau durante o treinamento do neuro-veículo.

O tempo médio de treinamento foi de 1 hora e 30 minutos (referente ao tempo de processamento em um computador Pentium III, 550 MHz). Pode-se ver a variação do EQM (erro quadrático médio) ao longo do tempo na figura 6.4 abaixo,

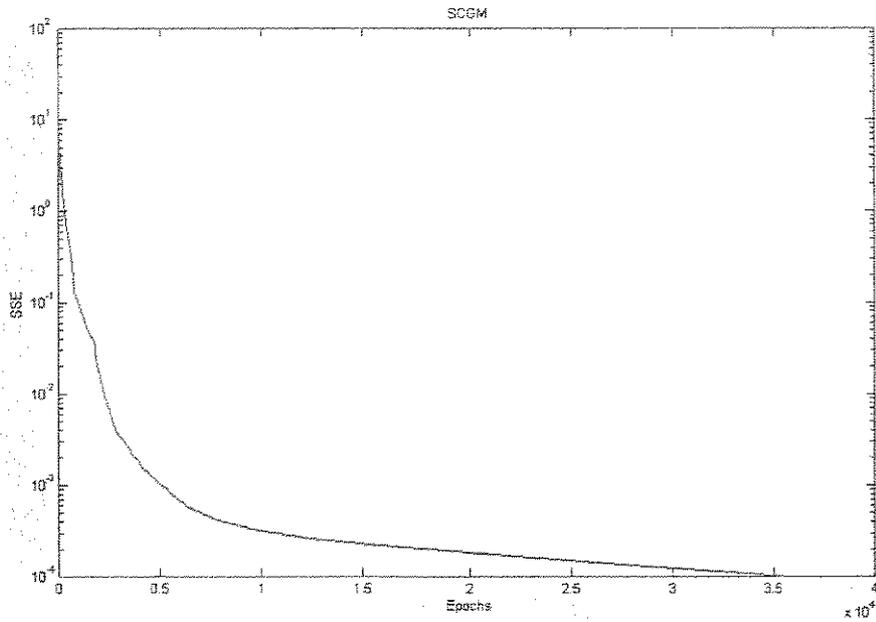


Figura 6.4 – Treinamento do neuro-veículo (EQM = 9.95×10^{-5}).

Após o treinamento foram feitos os testes usando excitações e condições iniciais diferentes daquelas usadas no processo de treinamento, onde foram obtidos os seguintes resultados:

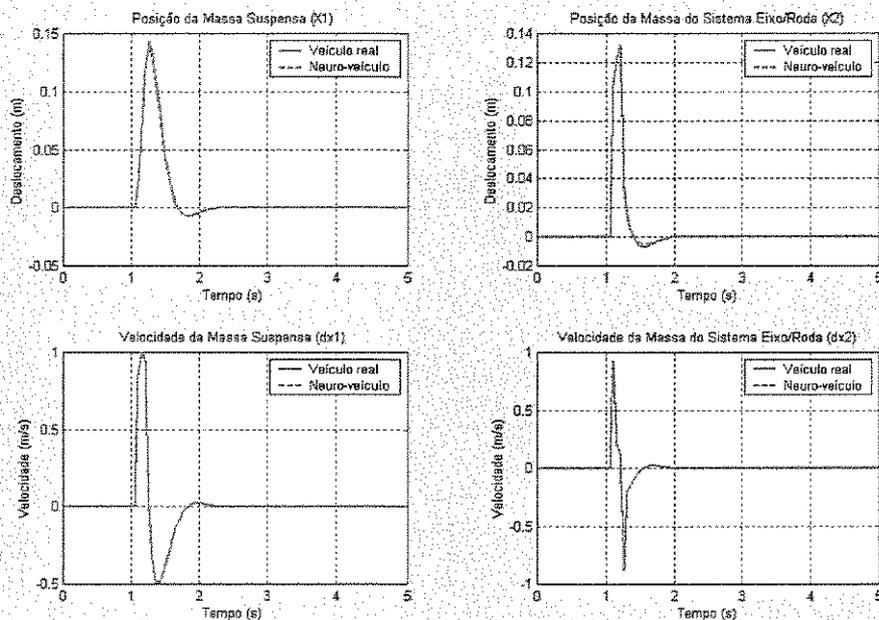


Figura 6.5 – Resposta à excitação tipo pulso. CI = [0 0 0 0].

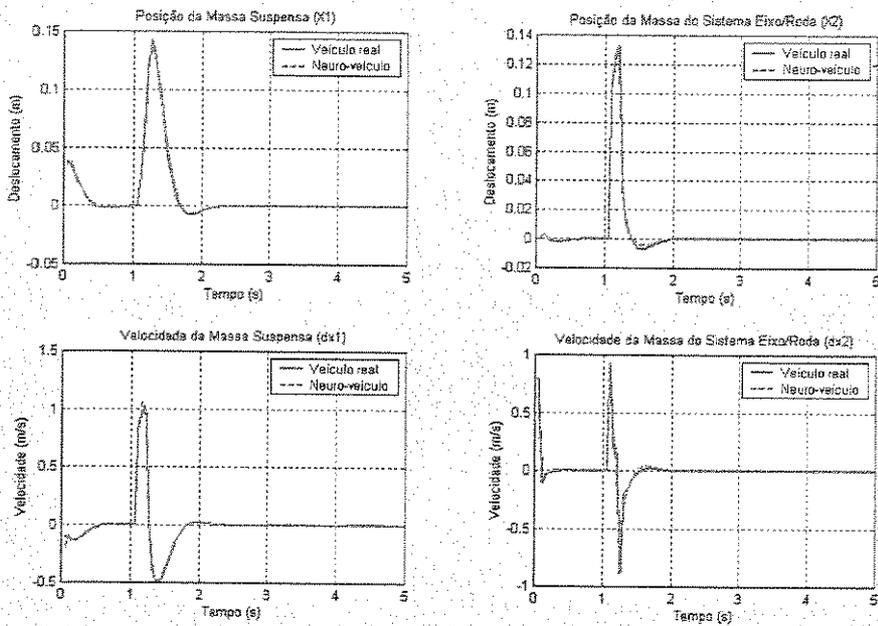


Figura 6.6 – Resposta à excitação tipo pulso. $CI = [0.05 \ -0.12 \ -0.74 \ 0.52]$.

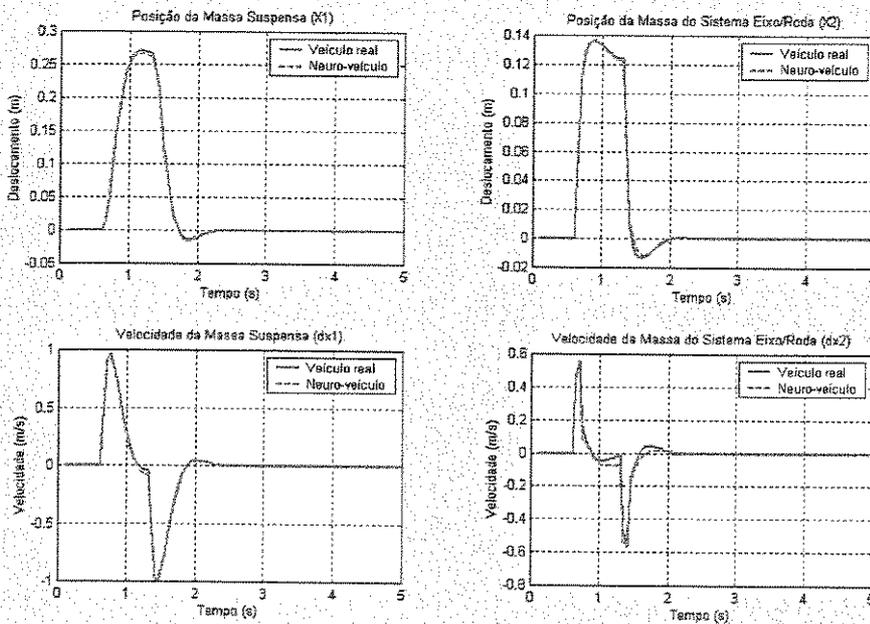


Figura 6.7 – Resposta à excitação em lombada. $CI = [0 \ 0 \ 0 \ 0]$.

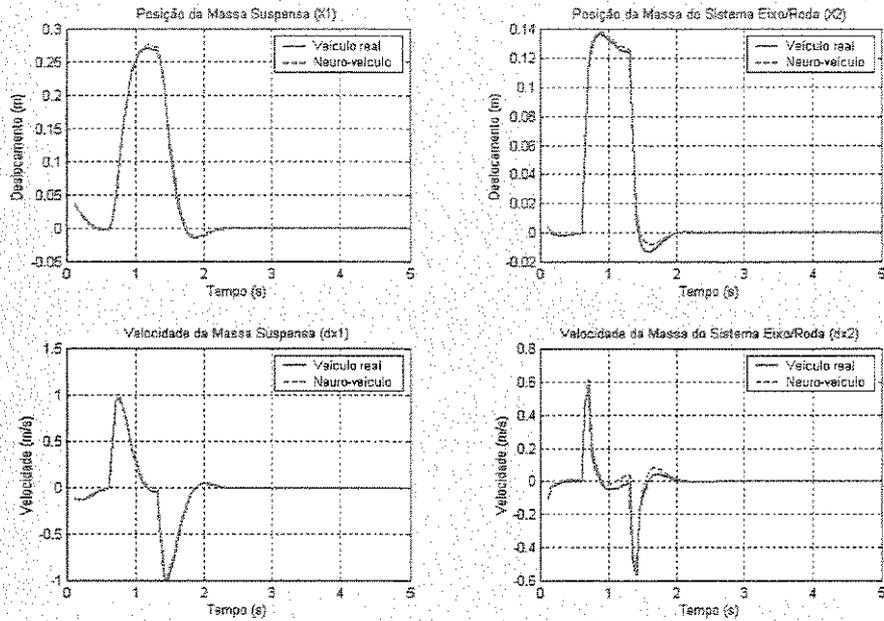


Figura 6.8 – Resposta à excitação em lombada. $CI = [0.05 \ -0.12 \ -0.74 \ 0.52]$.

Observando os gráficos das variáveis de estado do sistema ativo, pode-se concluir que o neuro-veículo foi bem treinado, sendo capaz de realizar a identificação da dinâmica da suspensão ativa proposta no capítulo 3, prevendo suas características futuras para diferentes valores de excitação e de condições iniciais. Uma vez treinado o neuro-veículo, os seus pesos sinápticos serão usados no treinamento do neuro-controlador a seguir.

6.4 – Neuro-Controlador Para Uma Suspensão Ativa Veicular.

Para o treinamento do neuro-controlador proposto na figura 5.6 foi utilizado como modelo de referência uma suspensão ativa com LQR conforme descrito nos capítulos 3 e 5, usando as constantes descritas na tabela 6.1. O objetivo é retropropagar o erro na saída da rede através do neuro-veículo para o treinamento dos pesos sinápticos do neuro-controlador (Park, 1996), (Moran, 1994). A RNA utilizada também é do tipo MLP com uma camada de entrada, uma

camada de saída e duas camadas intermediárias como mostrado na figura 5.7. O número de neurônios nas camadas de entrada e saída são diretamente determinados do número das variáveis de estado e de controle. A utilização de duas camadas neste caso, justifica-se pela necessidade de uma convergência mais precisa, devido a utilização do *backpropagation* como algoritmo de treinamento (Moran, 1994). Como foi descrito anteriormente no capítulo 4 e no capítulo 5, o treinamento do neuro-controlador foi realizado de acordo com o método de aprendizado direto, onde o algoritmo de treinamento adotado neste caso foi o **BPA** (*Backpropagation*), devido a facilidade de alterar internamente seu algoritmo, permitindo assim a atualização dos pesos sinápticos do neuro-controlador, mantendo-se fixos os pesos do neuro-veículo conforme descrito na seção 5.5.

O modelo do neuro-controlador foi dimensionado com 4 neurônios de entrada (4 variáveis de estado), 1 neurônio de saída (sinal de controle), 5 neurônios na primeira camada intermediária e 5 neurônios na segunda camada intermediária. O experimento foi desenvolvido também *off-line* segundo o que foi proposto anteriormente. O intervalo de amostragem foi mantido ($\Delta t = 0.05$ s), assim como o tempo de duração do experimento. O processo de treinamento foi realizado com o mesmo número de padrões do neuro-veículo, seguindo a mesma metodologia empregada, onde após o treinamento do mesmo, foram obtidas as respostas do sistema quando submetido à excitações e condições iniciais diferentes daquelas usadas no treinamento da rede.

O treinamento do neuro-controlador foi bem mais demorado que o do neuro-veículo, apresentando um tempo médio de 22 horas e 12 minutos (referente ao tempo de processamento em um computador Pentium III, 550 MHz), alcançando um EQM = 9.984×10^{-4} . Isto já era esperado devido a utilização do *backpropagation* cujas limitações em relação à taxa de convergência ficaram evidentes, sobretudo quando se aumentava o número de padrões de treinamento.

Outra consideração importante que deve ser citada é o fato que neste tipo de controlador, que realiza um mapeamento inverso, o erro ao passar pelo neuro-veículo fica bastante reduzido, diminuindo segundo taxas pequenas, aumentando assim o tempo de treinamento do neuro-controlador, como pode ser observado na figura abaixo.

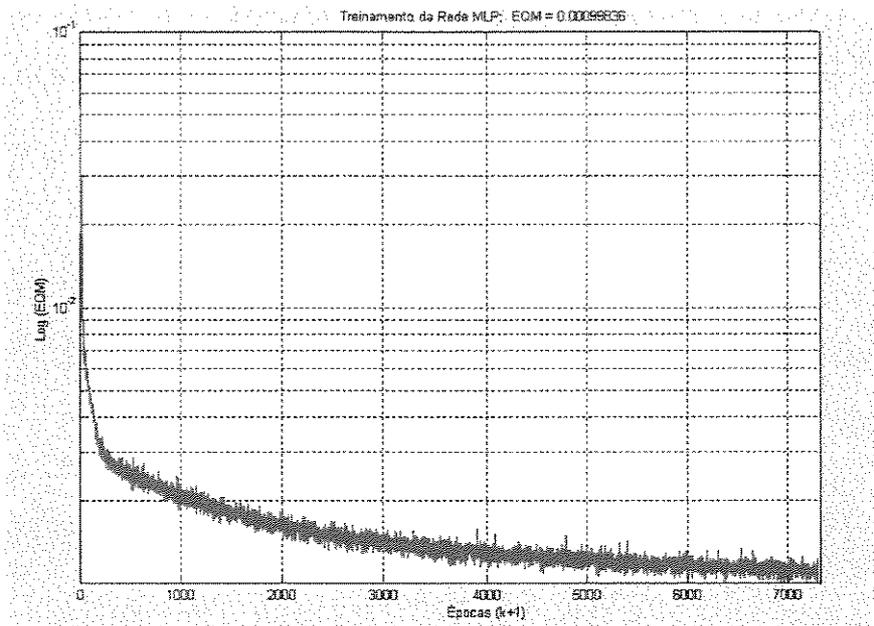


Figura 6.9 - Treinamento do neuro-controlador via *Backpropagation*.

Após o treinamento foram feitos os testes usando excitações e uma condição inicial diferente daquela usada no processo de treinamento, onde obteve-se para a lei de controle os seguintes resultados:

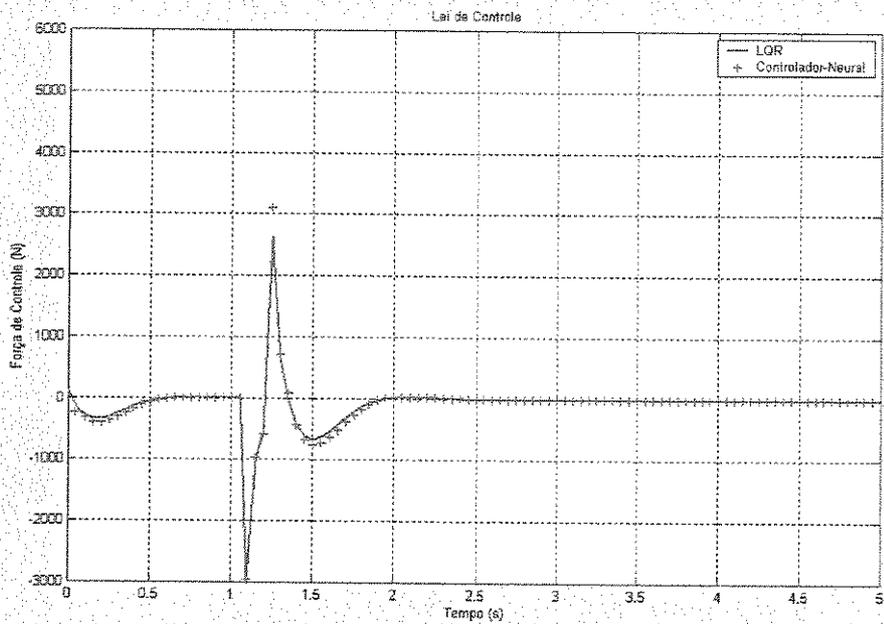


Figura 6.10 – Lei de controle à excitação tipo pulso. $CI = [0.05 \ -0.20 \ 0 \ 0]$.

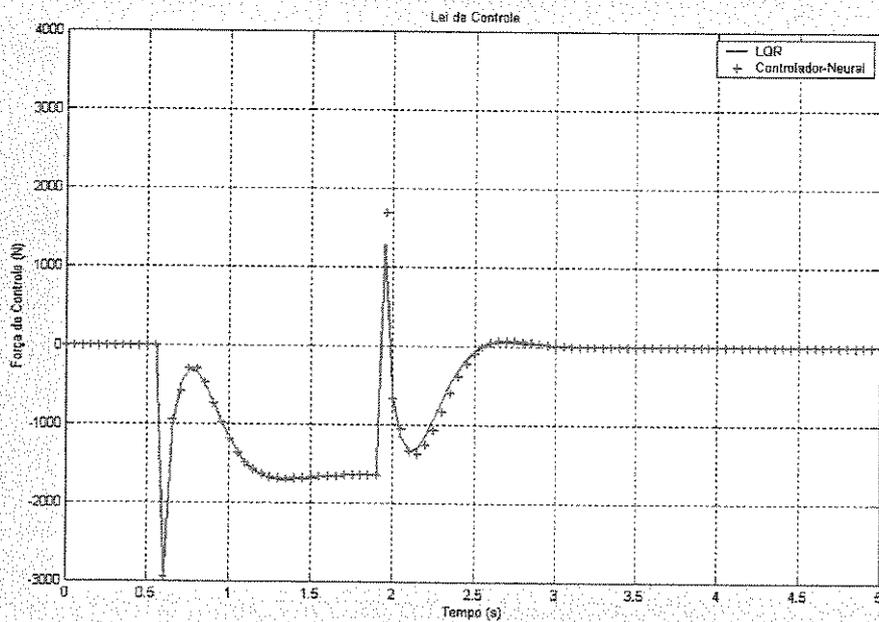


Figura 6.11 – Lei de controle à excitação em lombada. $CI = [0 \ 0 \ 0 \ 0]$.

Pode-se observar que o neuro-controlador responde bem aos testes, atendendo aos objetivos propostos, onde através do processo de aproximação da dinâmica inversa do sistema, conforme descrito no capítulo 5, o neuro-controlador consegue gerar uma lei de controle próxima à do modelo de referência adotado no treinamento. Observa-se então que o neuro-controlador reproduz a lei de controle do **LQR** utilizado como modelo de referência, comprovando sua capacidade neste processo de identificação inversa. Foi observado que se as condições iniciais variam muito, ou seja, se durante os testes forem escolhidas condições iniciais muito diferentes daquelas utilizadas durante o treinamento, dependendo da excitação empregada, a resposta neural nos pontos de descontinuidade não é tão boa. Isto pode ser melhorado aumentando o conjunto de treinamento através da inclusão de novas condições iniciais, o que implicará num aumento do tempo de treinamento das redes.

6.4.1 – Análise dos resultados:

Antes porém, deve-se fazer alguns esclarecimentos em relação ao desenvolvimento das simulações computacionais, sobretudo com a utilização do **LQR**, tanto para o treinamento das redes neurais, como também para a análise dos resultados a seguir. O **LQR** por ser um controlador do tipo **PD** (proporcional e derivativo) apresenta um erro estacionário, ou seja, uma deflexão estacionária que é ocasionada no sistema devido a uma excitação degrau ou rampa ou devido a forças aplicadas no corpo, que pode ser eliminado com a incorporação de integradores no sistema (Alves, 1997) (Ogata, 1998), o que não foi feito neste trabalho. Além disso foram utilizadas nas simulações, funções do Matlab, pacotes de programas prontos (por exemplo, ODE45), que podem contribuir para um aumento da imprecisão dos resultados, sobretudo nos pontos de descontinuidade. Outro fator que também não foi considerado, foi um estudo da discretização empregada, sobretudo em relação a estas descontinuidades, pois isto também tem influência nos resultados obtidos em torno destes pontos.

Será feito agora uma análise comparativa entre os modelos de suspensão veicular (passiva e ativa) propostos no capítulo 3, usando para isso os critérios de projeto de uma suspensão, ou seja,

o espaço de trabalho da suspensão (equação A.1.12), a aceleração vertical da massa suspensa e a força resultante do contato do pneu com a via (deve-se obter a força resultante como sendo a soma entre o peso total do modelo de ¼ de veículo e a força de contato do pneu calculada nas simulações), como os parâmetros que devem ser analisados. Obteve-se então as seguintes respostas:

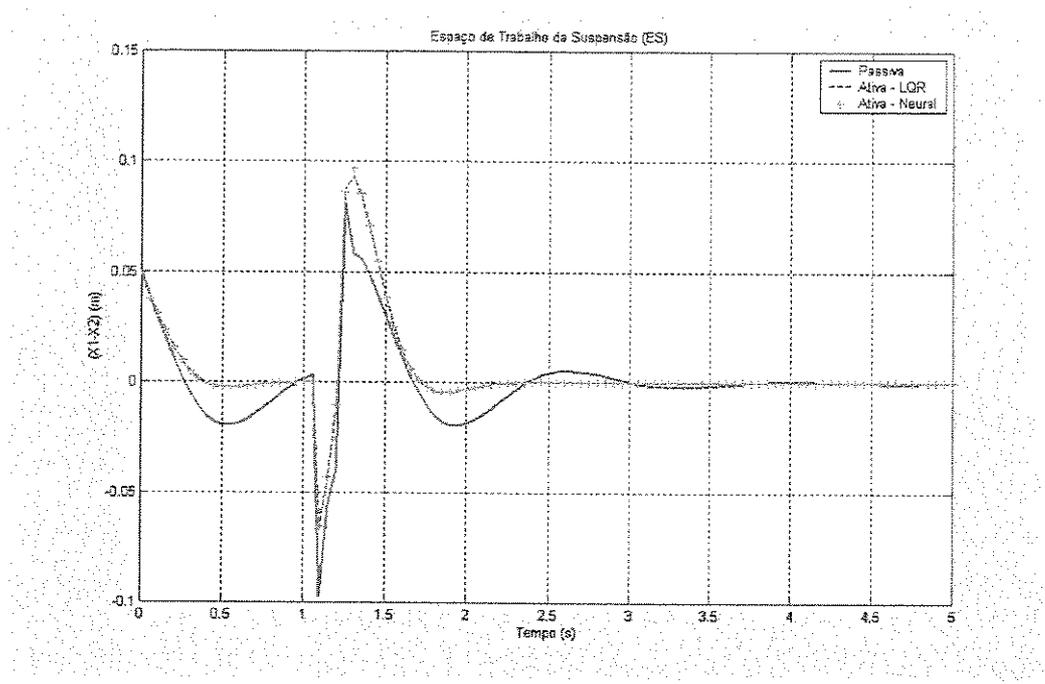


Figura 6.12 – Resposta à excitação tipo pulso (espaço de trabalho da suspensão).

$$CI = [0.05 \ 0 \ -0.20 \ 0].$$

Considerando o espaço de trabalho da suspensão a partir da figura acima, observa-se que o valor máximo para o sistema passivo sofre uma redução no sistema ativo, no primeiro ponto de descontinuidade, o que caracterizaria uma suspensão mais compacta. Além disso para o sistema ativo a oscilação é também bem menor, possuindo então um maior amortecimento. A rede neural consegue acompanhar a resposta do modelo de referência, cuja excitação e condição inicial são diferentes daquelas usadas no treinamento. Isto já era esperado uma vez que o neuro-controlador conseguiu reproduzir com êxito a lei de controle do modelo de referência, conforme foi visto anteriormente.

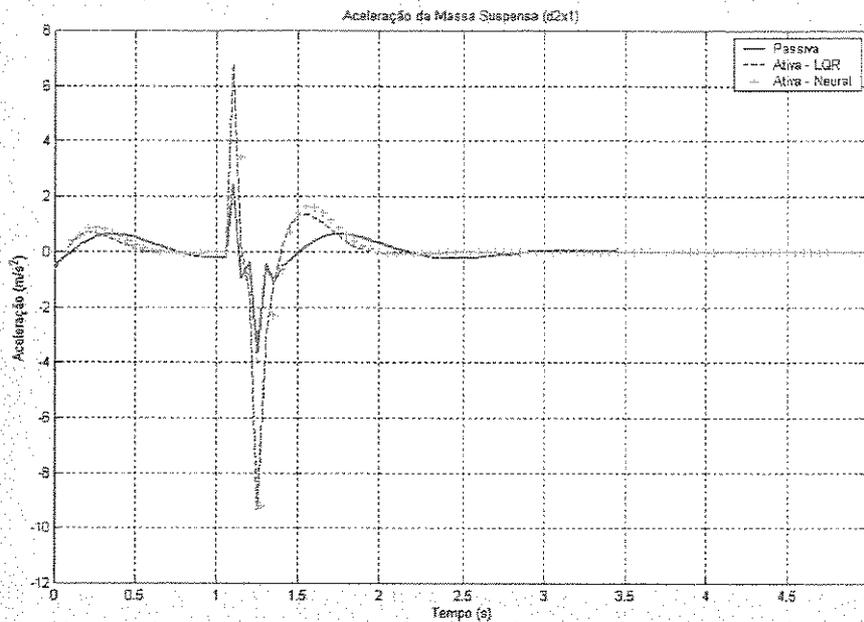


Figura 6.13 – Resposta à excitação tipo pulso (aceleração da massa suspensa).

$$CI = [0.05 \ 0 \ -0.20 \ 0].$$

A aceleração vertical da massa suspensa, utilizada como medida do conforto do passageiro, apresenta valores superiores ao do sistema passivo após o pneu passar pelo obstáculo conforme visto na figura 6.13. A presença do erro estacionário concorre para um aumento destes valores. Observa-se também que a mesma é amortecida mais rapidamente. A rede neural consegue acompanhar o modelo de referência e atinge um valor de pico menor no início do pulso, isto se deve provavelmente, a algum erro no processo de obtenção das variáveis de estado, devido aos motivos citados no parágrafo inicial deste item. Lembrando que para o projeto de suspensões automotivas segundo o critério de conforto dos passageiros, deve-se ter valores baixos de aceleração e rápido amortecimento, justifica-se a utilização dos integradores para a eliminação do erro estacionário na implementação prática do controlador proposto.

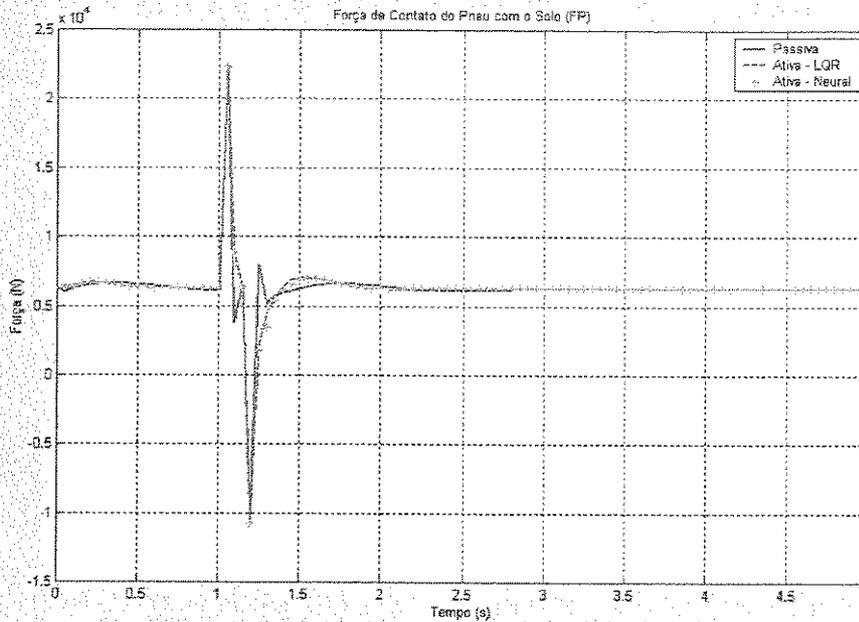


Figura 6.14 – Resposta à excitação tipo pulso (força de contato do pneu com a via).

$$CI = [0.05 \ 0 \ -0.20 \ 0].$$

Conforme o que foi dito anteriormente, na obtenção da força de contato do pneu com a via, utilizada como parâmetro de medida da segurança do veículo, o seu valor é somada à força estática (massa total do modelo), tendo-se assim a força resultante. A análise do comportamento dessas forças possibilita a previsão da perda de contato roda-via através da análise dos seus sinais. Isto é feito sabendo-se que a perda de contato ocorre quando tais forças possuem sinais negativos, indicando que houve inversão no sentido de atuação. Observa-se que a força de contato assume valores menores se comparado à suspensão passiva durante a aplicação da excitação. A força de contato apresenta um amortecimento mais rápido no sistema ativo se comparado com o passivo. Aqui também a rede neural acompanha a resposta do modelo de referência.

Aplicando agora uma excitação em lombada, pode-se observar uma pequena diferença em relação ao modelo neural, na aceleração da massa suspensa e na força de contato do pneu, sendo porém perfeitamente aceitável.

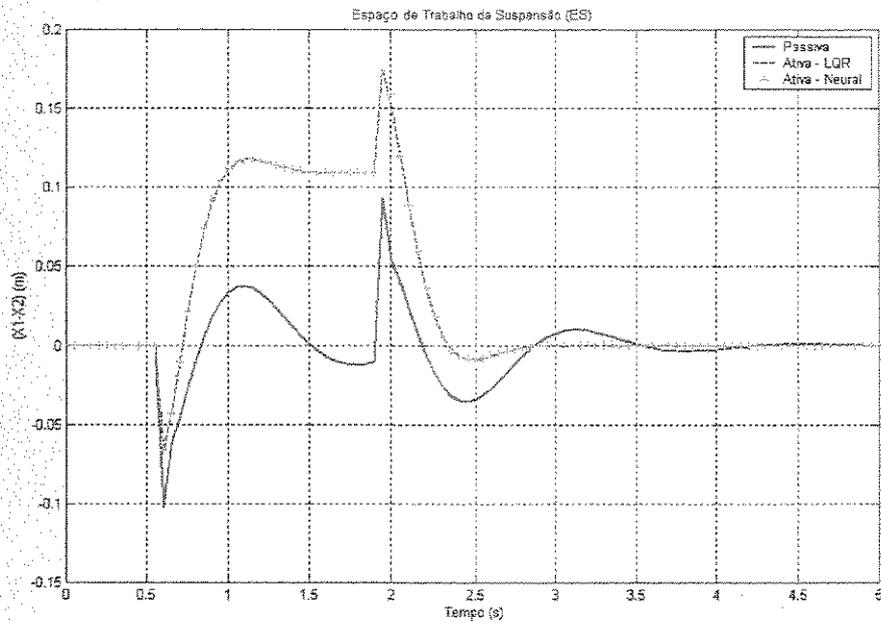


Figura 6.15 – Resposta à excitação em lombada (espaço de trabalho da suspensão).

$$CI = [0 \ 0 \ 0 \ 0]$$

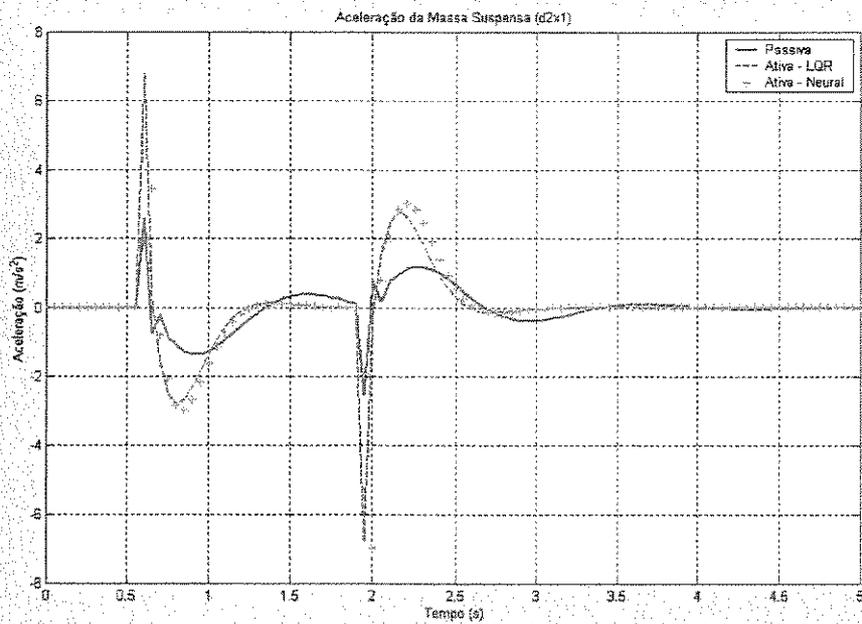


Figura 6.16 – Resposta à excitação em lombada (aceleração da massa suspensão).

$$CI = [0 \ 0 \ 0 \ 0]$$

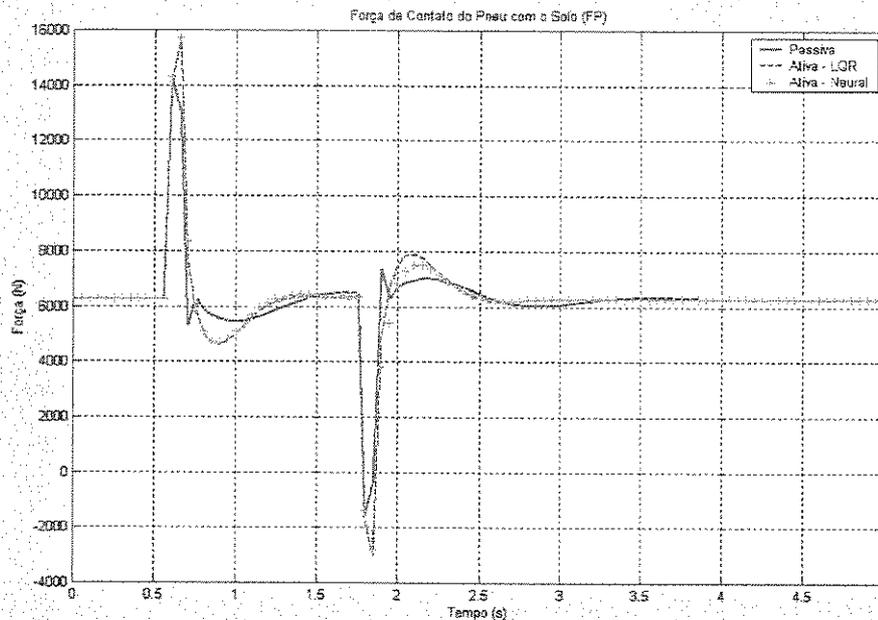


Figura 6.17 – Resposta à excitação em lombada (força de contato do pneu com a via).

$$CI = [0 \ 0 \ 0 \ 0]$$

Pode-se observar claramente na análise do espaço de trabalho da suspensão, quando o sistema está submetido a uma excitação em lombada, a presença do erro estacionário, que afeta a resposta do sistema levando-o a um comportamento indesejável em relação ao sistema passivo. O uso de integradores resolverá este problema conduzindo a suspensão ativa a um comportamento ótimo sob todas as condições de operação, quando comparada aos sistemas passivos.

6.4.2 – Construção e análise de um modelo de suspensão ativa não linear:

Como foi destacado anteriormente as redes neurais apresentam a capacidade de aprender ou reconhecer uma grande variedade de funções não lineares com entradas e saídas limitadas à uma precisão arbitrária. Assim sendo, pode-se aplicar as redes neurais para identificar a dinâmica de uma suspensão ativa com componentes não lineares sem a necessidade de algum tipo de técnica

de linearização. Após a identificação do modelo não linear é feito o treinamento do neuro-controlador através da retropropagação do erro da saída do neuro-veículo até a entrada da rede. Os pesos do neuro-veículo são chamados de matrizes de “sensibilidade” pois armazenam, e são capazes de reconhecer, as características não lineares da suspensão, sendo usados para impor tais características ao treinamento do neuro-controlador e conseqüentemente na obtenção da lei de controle, através de um processo de aproximação inversa, que levará o sistema a se comportar de modo semelhante ao modelo de referência adotado. Segundo Moran, 1994, o modelo adotado deve ser definido pelo projetista de acordo com os critérios de projeto da suspensão (conforto do passageiro e segurança). Pode ser utilizado então um modelo conhecido com características lineares ou não, cujo desempenho satisfaça tais critérios.

Utilizaremos o modelo da suspensão ativa de ¼ de veículo com dois graus de liberdade conforme exposto na figura 3.1, com a substituição da mola por outra não linear. Foi assumido que a mola da suspensão tem a seguinte característica não linear:

$$f_s = k_{ms} \operatorname{sgn}(\Delta x) |\Delta x|^{nl} \quad (6.5)$$

onde f_s é a força dinâmica da mola, k_{ms} é o coeficiente de rigidez equivalente, Δx é o movimento relativo de ambos os extremos da mola ($X_1 - X_2$), $\operatorname{sgn}(\cdot)$ é o sinal da força (compressão ou tração) e nl é um expoente representando a não linearidade da mola e usualmente varia entre 1 e 1.45 quando Δx é dado em cm e k_{ms} em N/cm (Moran, 1994). Como as unidades utilizadas no trabalho estão em metros, deve-se multiplicar f_s por um fator de correção $\alpha = 10^{2(nl-1)}$ para garantirmos a proporcionalidade entre os modelos. Na figura abaixo pode-se observar a relação não linear entre Δx e f_s conforme expresso pela equação 6.5 e sua comparação com as características de uma mola linear. Usou-se neste modelo $nl = 1.1$ para a simulação do modelo não linear.

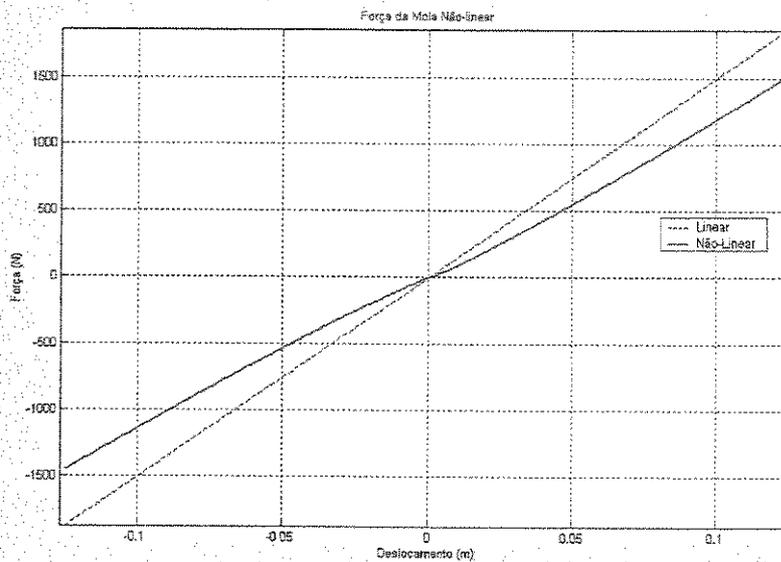


Figura 6.18 – Mola não linear.

As equações diferenciais que descrevem o movimento vertical do veículo podem ser descritas por:

$$M s \ddot{X}_1 + c a (\dot{X}_1 - \dot{X}_2) + [\alpha k m s (X_1 - X_2)^{nl}] + f = 0 \quad (6.6)$$

$$m \ddot{X}_2 - c a (\dot{X}_1 - \dot{X}_2) - [\alpha k m s (X_1 - X_2)^{nl}] - f + k p X_2 = k p r \quad (6.7)$$

Para a análise e simulação deste tipo de sistema ativo deve-se utilizar algum método de integração para a resolução de sistemas de equações diferenciais não lineares a fim de se obter a resposta do sistema à diferentes tipos de excitações. Utilizando o vetor de estado definido pela equação 3.6 pode-se montar o seguinte sistema de equações diferenciais:

$$\begin{cases} \dot{z}_1 = \dot{X}_1 \\ \dot{z}_2 = \dot{X}_2 \\ \dot{z}_3 = \ddot{X}_1 = -\left(\frac{ca}{Ms}\right) (\dot{X}_1 - \dot{X}_2) - [\alpha \left(\frac{kms}{Ms}\right) (X_1 - X_2)^{nl}] - f \\ \dot{z}_4 = \ddot{X}_2 = \left(\frac{ca}{mu}\right) (\dot{X}_1 - \dot{X}_2) + [\alpha \left(\frac{kms}{mu}\right) (X_1 - X_2)^{nl}] + \left(\frac{kp}{mu}\right) r + f \end{cases} \quad (6.8)$$

Inicialmente foi feita a identificação da dinâmica do modelo não linear. O modelo do neuro-veículo (arquitetura das redes) foi dimensionado de forma idêntica ao do modelo linear. O experimento também foi desenvolvido como um procedimento *off-line*, obtendo 200 padrões de treinamento a partir de diferentes condições iniciais escolhidas aleatoriamente. Após o treinamento do neuro-veículo este foi testado com excitações e condições iniciais diferentes daquelas usadas no treinamento da rede. O tempo médio de treinamento foi de 3 horas e 30 minutos (referente ao tempo de processamento em um computador Pentium III, 550 MHz). Pode-se ver a variação do EQM (erro quadrático médio) ao longo do tempo na figura abaixo,

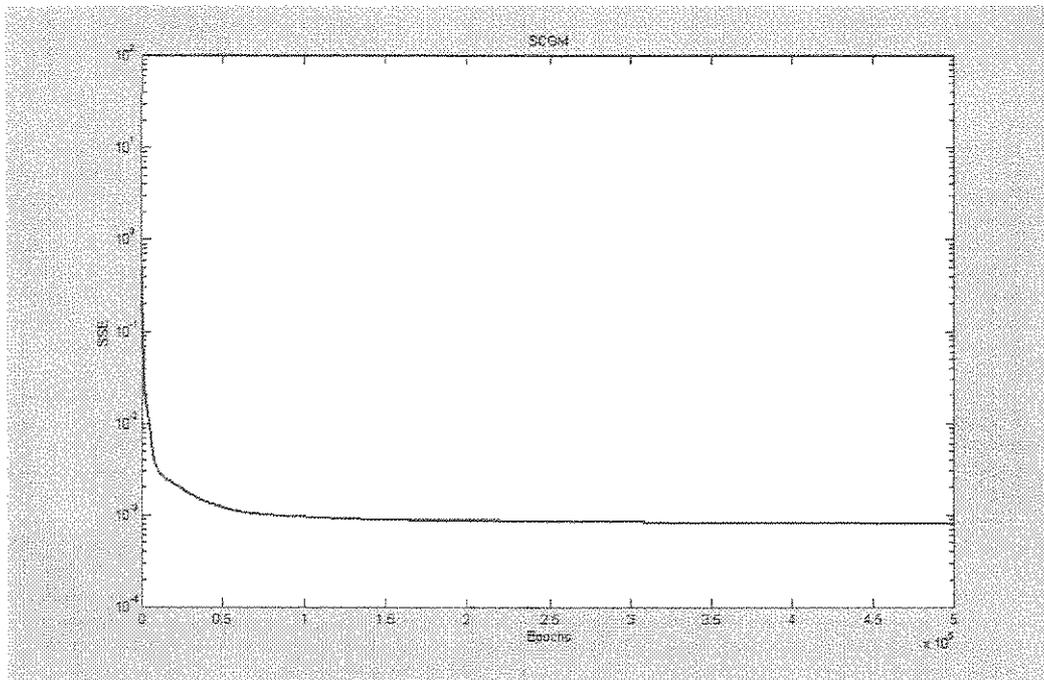


Figura 6.19 – Treinamento do neuro-veículo (EQM = 8.154 x 10⁻⁴).

Após o treinamento foram feitos os testes usando excitações e condições iniciais diferentes daquelas usadas no processo de treinamento, onde obteve-se os seguintes resultados:

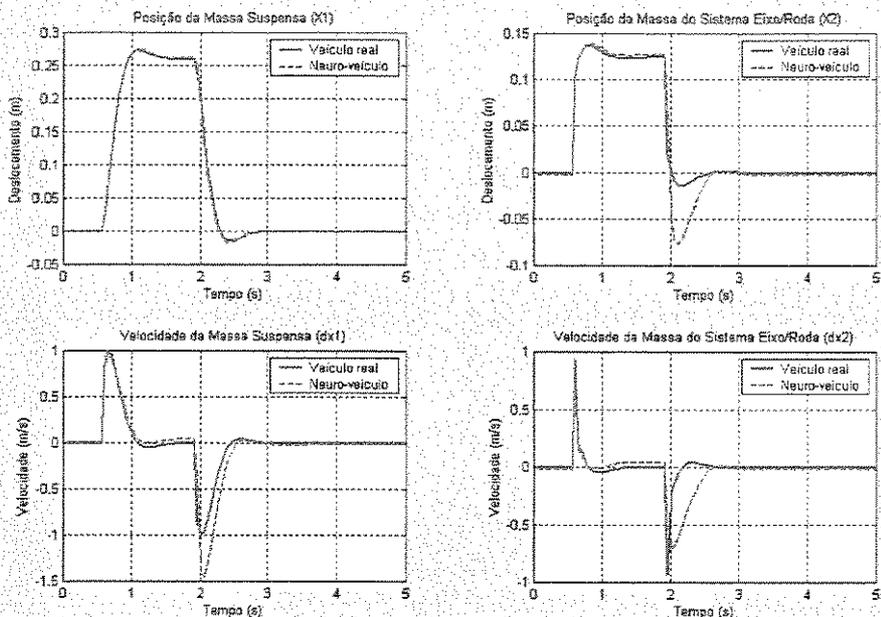


Figura 6.20 – Resposta à excitação em lombada. $CI = [0 \ 0 \ 0 \ 0]$.

Neste caso, assim como para a excitação tipo pulso, observa-se que a rede neural não se comporta de maneira idêntica na identificação dos vetores de estado no sistema ativo não linear. A precisão diminui para a posição do sistema eixo/roda e para as velocidades da massa suspensa e do sistema eixo/roda. A explicação pode estar no critério de parada adotado (quando o EQM atingir um valor desejado) que beneficia apenas um parâmetro ou então na entrada de excitação (degrau) usada no treinamento que para o caso não linear não consegue varrer uma faixa suficiente de frequências.

Para tentar solucionar o problema foi feito um novo treinamento aumentando o número de padrões para 1600, através da diminuição do intervalo de amostragem. O tempo médio de treinamento foi de 8 horas e 45 minutos. Pode-se ver a variação do EQM (erro quadrático médio) ao longo do tempo na figura abaixo,

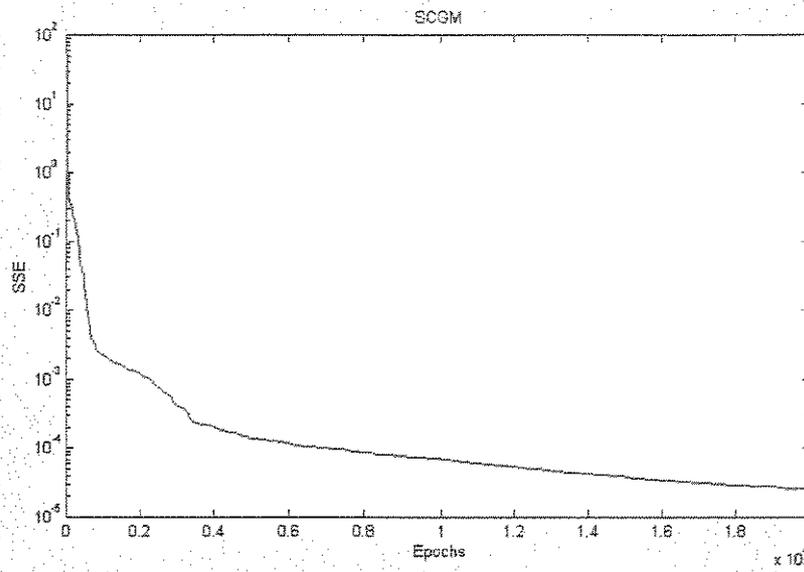


Figura 6.21 – Treinamento do neuro-veículo (EQM = 1.6×10^{-5}).

Após o treinamento foi repetida a metodologia anterior, onde obteve-se os seguintes resultados:

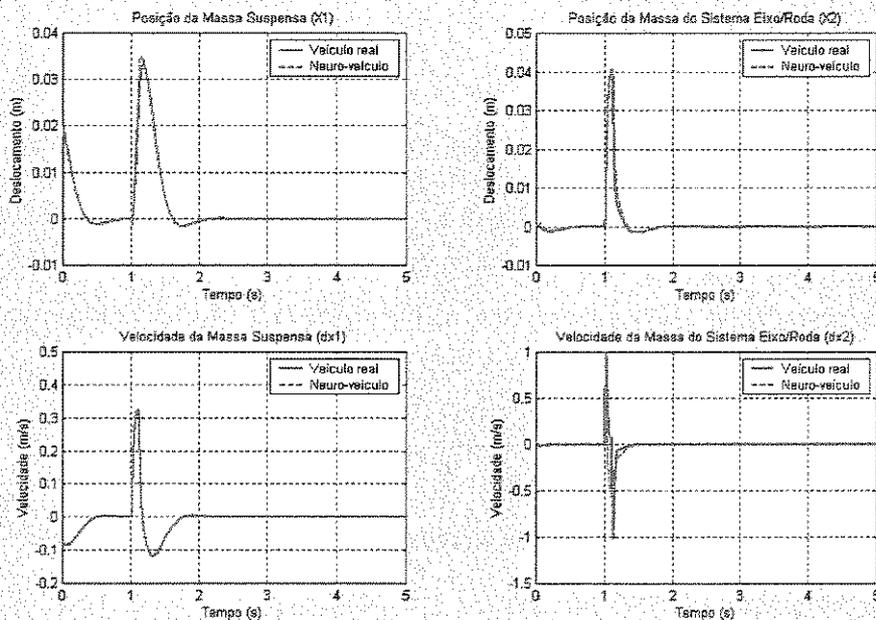


Figura 6.22 – Resposta à excitação tipo pulso. CI = $[0.05 \ 0 \ -0.2 \ 0]$.

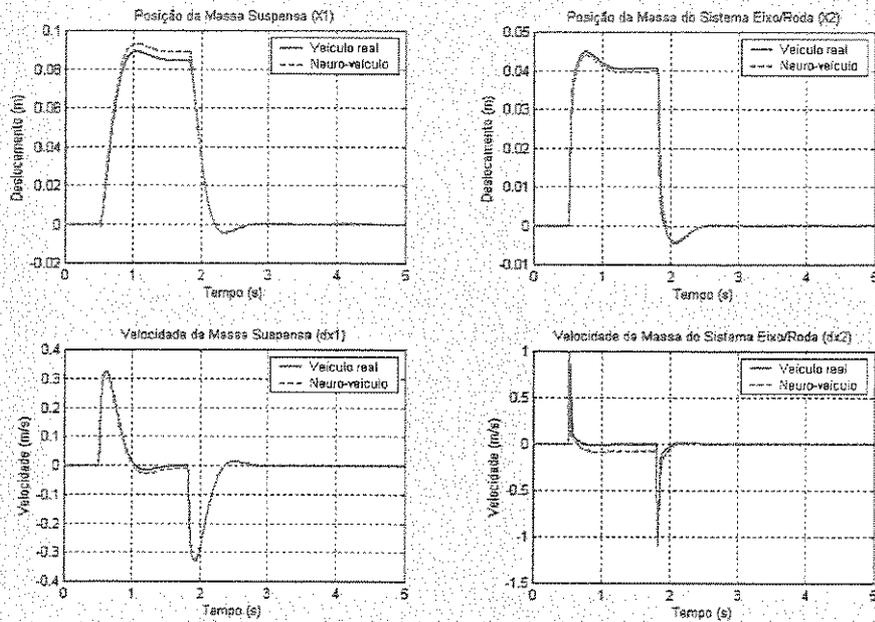


Figura 6.23 – Resposta à excitação em lombada. $CI = [0 \ 0 \ 0 \ 0]$.

Pode-se observar agora que a rede consegue identificar a dinâmica não linear da suspensão ativa. Os pesos sinápticos do neuro-veículo carregam as informações do sistema não linear que serão usadas no treinamento no neuro-controlador a seguir.

O modelo do neuro-controlador foi dimensionado de forma idêntica ao caso anterior. O processo de treinamento foi realizado com o mesmo número de padrões do neuro-veículo, seguindo a mesma metodologia empregada, onde após o treinamento do mesmo, foram obtidas as respostas do sistema quando submetido à excitações e condições iniciais diferentes daquelas usadas no treinamento da rede.

O treinamento do neuro-controlador foi bem mais demorado que o do neuro-veículo, apresentando um tempo médio de mais de 30 horas (referente ao tempo de processamento em um computador Pentium III, 550 MHz), alcançando um $EQM = 1.83 \times 10^{-3}$, sendo um valor superior ao conseguido para o modelo linear. Isto já era esperado devido a presença da não linearidade.

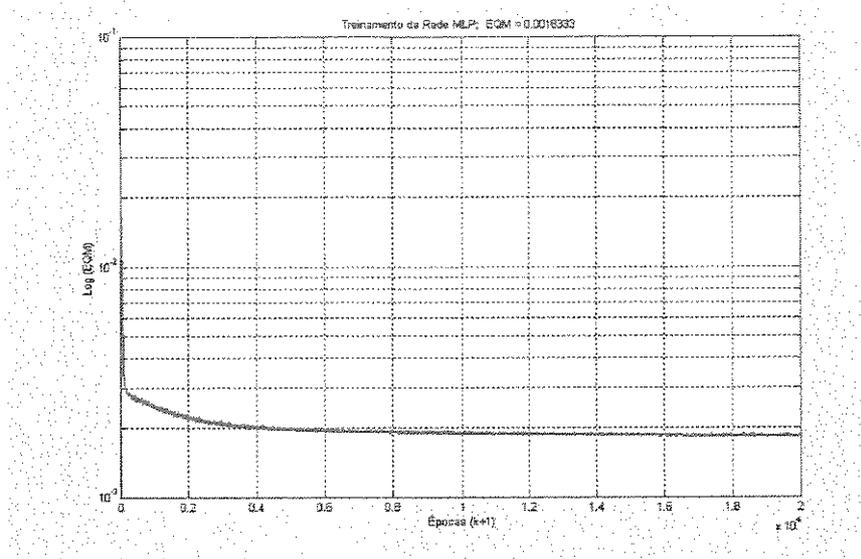


Figura 6.24 - Treinamento do neuro-controlador via *Backpropagation*.

Após o treinamento foram feitos os testes usando excitações e uma condição inicial diferente daquela usada no processo de treinamento, onde obteve-se para a lei de controle o seguinte resultado:

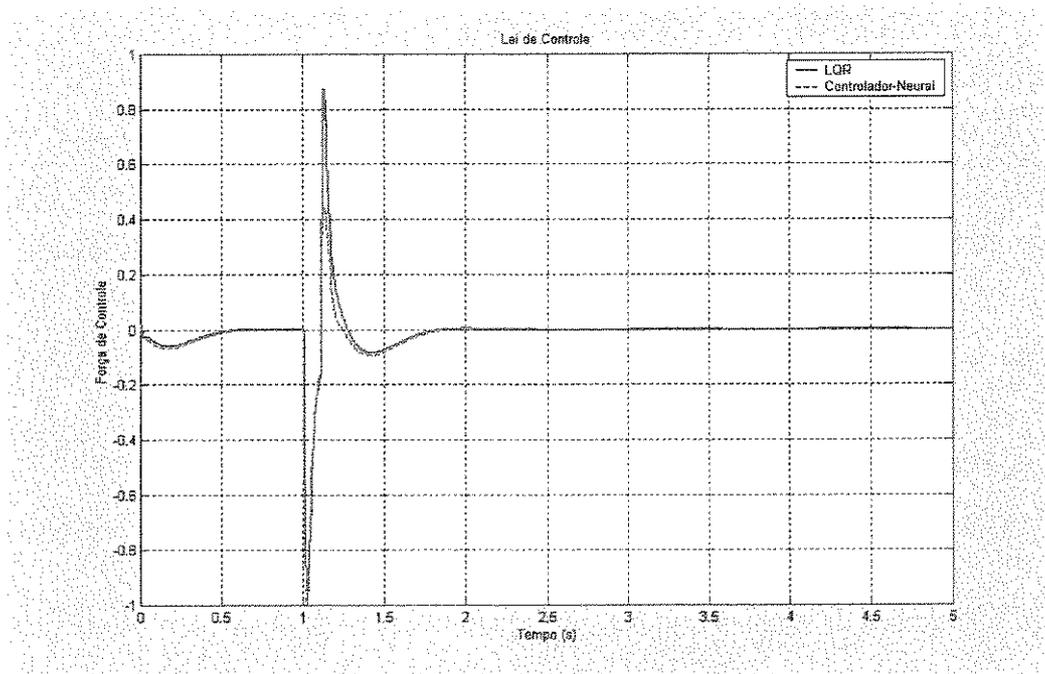


Figura 6.25 – Lei de controle à excitação tipo pulso. $CI = [0.05 \ 0 \ -0.2 \ 0]$.

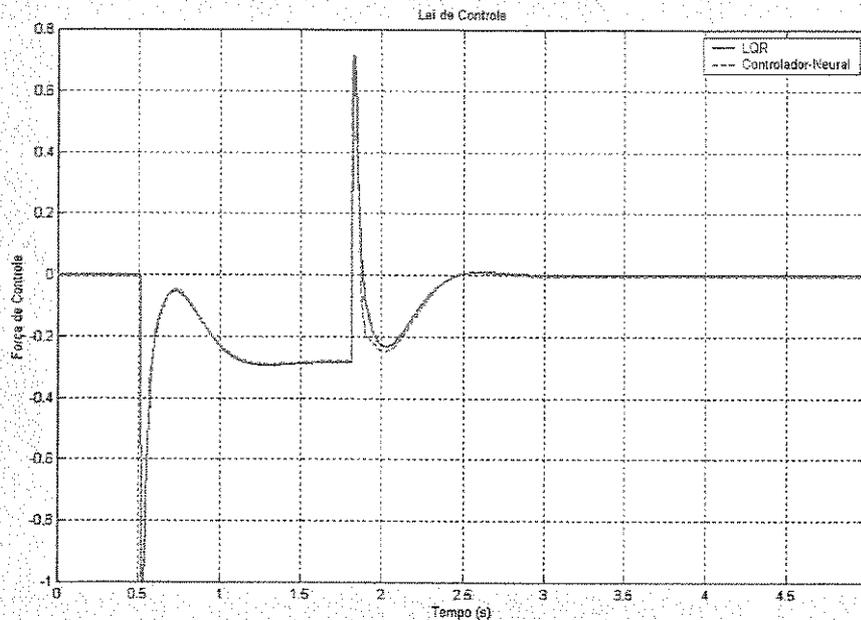


Figura 6.26 – Lei de controle à excitação em lombada. $CI = [0 \ 0 \ 0 \ 0]$.

Estes resultados demonstram a capacidade da rede neural em gerar uma lei de controle para o sistema não linear seguindo um dado modelo de referência, através de um processo de aproximação inversa da dinâmica do sistema. Voltou-se a observar uma diferença no ponto que representa a segunda descontinuidade, isto conforme foi dito anteriormente deve ser resultado de imprecisões no processo de integração ou na intervalo de amostragem adotado. Foram testadas outras condições iniciais e verificou-se a necessidade de aumentar o conjunto de treinamento (número de padrões), pois a rede se perde na identificação dos pontos iniciais para alguns casos da simulação.

Será feito agora uma análise comparativa entre os modelos de suspensão veicular (passiva e ativa não linear) como foi proposto anteriormente, levando-se em conta todas as considerações feitas anteriormente em relação a presença do erro estacionário. Obteve-se então as seguintes respostas:

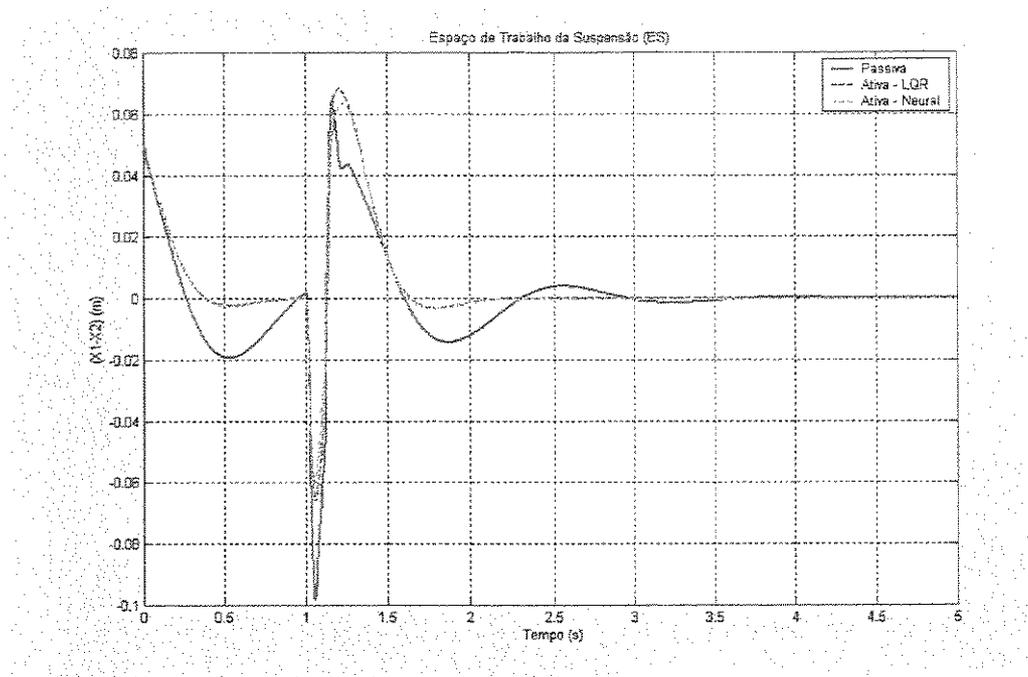


Figura 6.27 – Resposta à excitação tipo pulso (espaço de trabalho da suspensão).

$$CI = [0.05 \ 0 \ -0.20 \ 0].$$

Considerando o espaço de trabalho da suspensão a partir da figura acima, observa-se que o resultado é similar ao caso da suspensão linear. A rede neural consegue acompanhar a resposta do modelo de referência, cuja excitação e condição inicial são diferentes daquelas usadas no treinamento, apresentando uma pequena diferença de amplitude, caracterizando assim um espaço de trabalho menor, o que torna a suspensão mais compacta.

A aceleração vertical da massa suspensa, utilizada como medida do conforto do passageiro, apresenta valores superiores ao do sistema passivo após o pneu passar pelo obstáculo conforme visto na figura 6.28. Também observa-se um rápido amortecimento. A rede neural consegue acompanhar o modelo de referência, e atingindo valores de pico inferiores ao do sistema passivo e ativo linear.

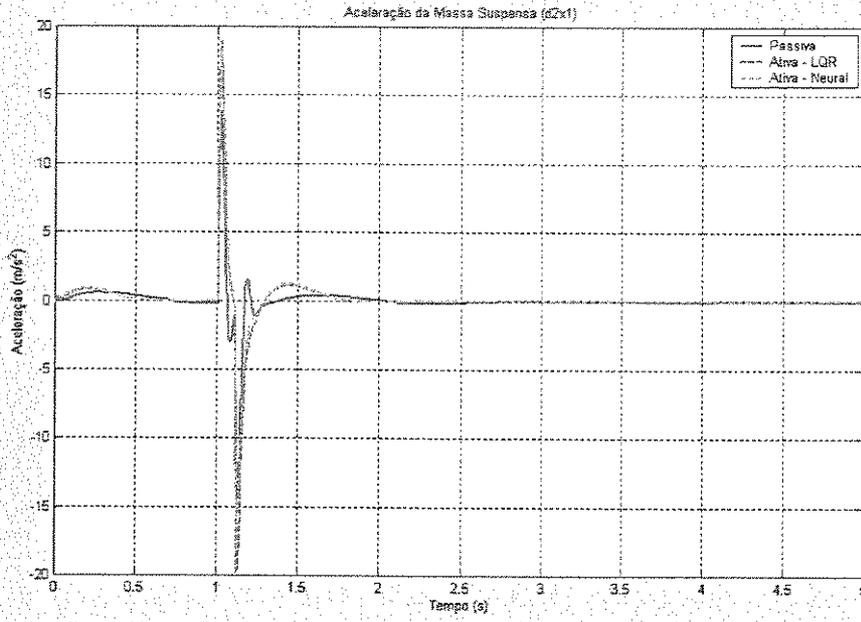


Figura 6.28 – Resposta à excitação tipo pulso (aceleração da massa suspensa).

$$CI = [0.05 \ 0 \ -0.20 \ 0].$$

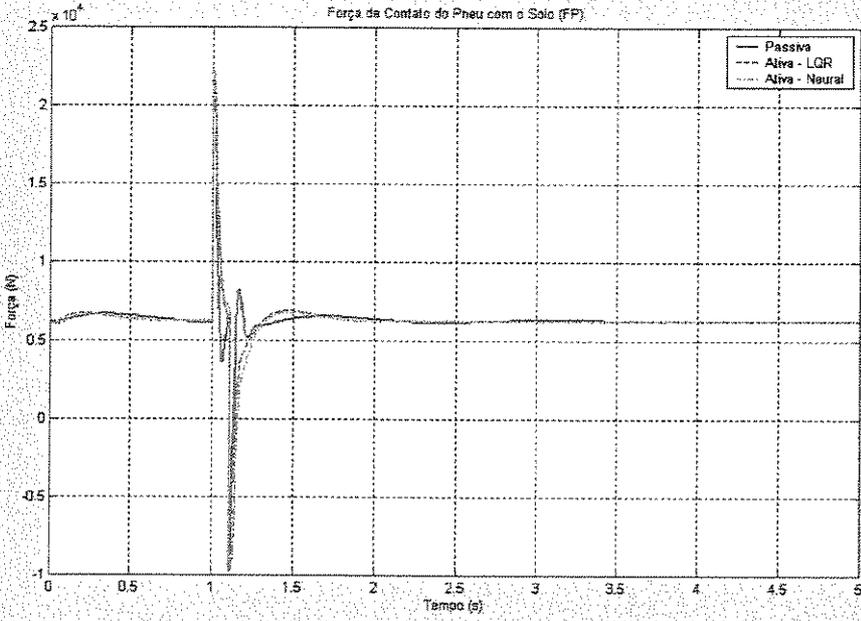


Figura 6.29 – Resposta à excitação tipo pulso (força de contato do pneu com a via).

$$CI = [0.05 \ 0 \ -0.20 \ 0].$$

Neste caso também a rede neural acompanha a resposta do modelo de referência.

Aplicando agora uma excitação em lombada, pode-se observar que o comportamento dos parâmetros sob análise não apresentam grandes diferenças em relação ao caso anterior, a resposta neural, que consegue acompanhar a resposta do modelo de referência. Aqui também podemos observar a presença do erro estacionário de forma mais clara.

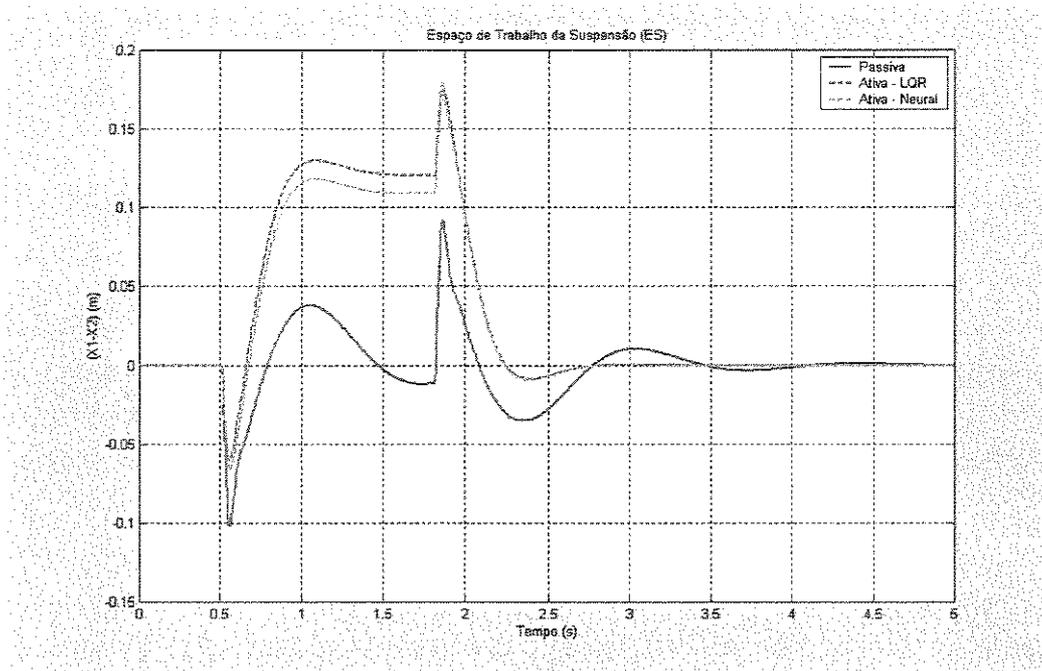


Figura 6.30 – Resposta à excitação em lombada (espaço de trabalho da suspensão).

$$CI = [0 \ 0 \ 0 \ 0]$$

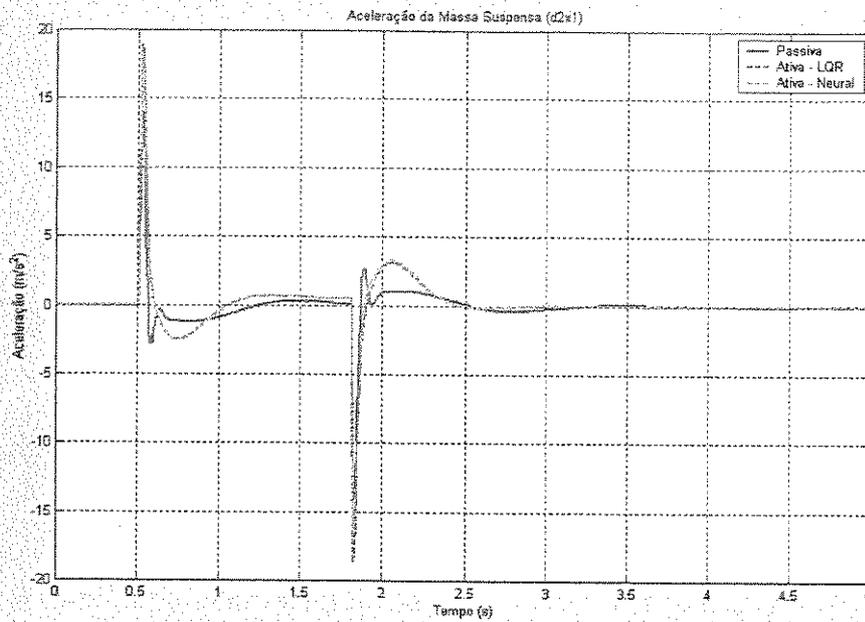


Figura 6.31 – Resposta à excitação em lombada (aceleração da massa suspensa).

$$CI = [0 \ 0 \ 0 \ 0]$$

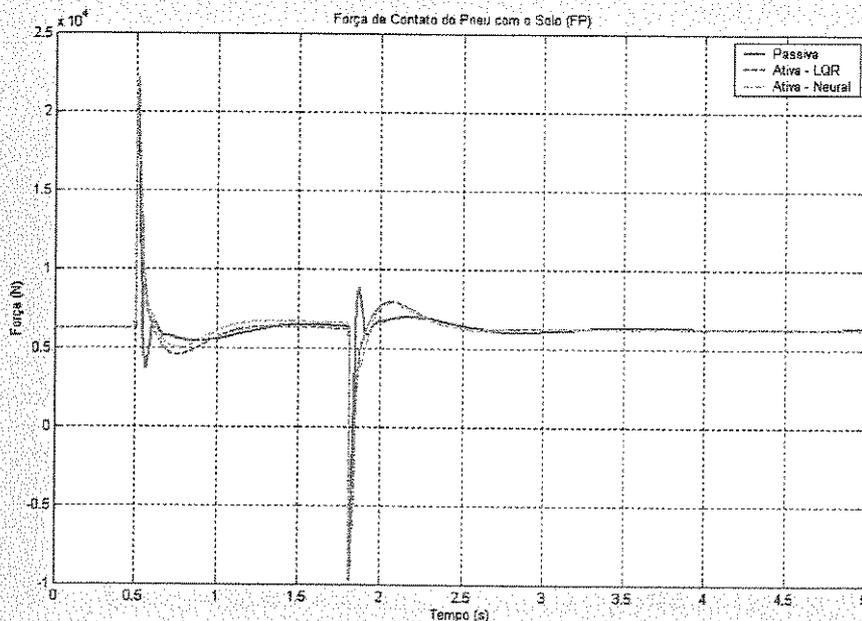


Figura 6.32 – Resposta à excitação em lombada (força de contato do pneu com a via).

$$CI = [0 \ 0 \ 0 \ 0]$$

De um modo geral observa-se que a rede neural consegue gerar uma lei de controle similar à do modelo de referência, impondo assim à suspensão ativa não linear um comportamento próximo ao do modelo de referência adotado. A escolha do modelo de referência, como foi dito anteriormente, depende dos critérios de projeto da suspensão. Usou-se um modelo linearizado, baseado no **LQR** como modelo de referência, para verificar a capacidade do controlador neural em gerar a lei de controle através do processo de identificação inversa, porém outros modelos de referência, lineares ou não, podem ser empregados, desde que o mesmo tenha um desempenho satisfatório.

6.4.3 – Alterando o modelo de referência e os parâmetros do sistema original:

Visando demonstrar a capacidade de adaptatividade das redes neurais, ou seja, a capacidade da rede adaptar-se a novas condições de comportamento ou a variações nos parâmetros do sistema, nos processos de identificação e controle de sistemas dinâmicos, foram realizados dois novos testes.

Primeiramente verificou-se a capacidade do neuro-controlador em acompanhar um modelo de referência qualquer, com características diferentes do sistema original. Para o novo modelo de referência o valor da constante de amortecimento foi reduzida para **1800 Ns/m** (ver valores originais na tabela 6.1). Usando os pesos sinápticos gerados na identificação do sistema original, não linear, ou seja, não foi realizado um novo treinamento do neuro-veículo, foi feito o treinamento do neuro-controlador, empregando para tal uma arquitetura de rede e um número de padrões idênticos ao caso anterior, segundo este novo modelo de referência.

O tempo médio de treinamento do neuro-controlador foi próximo ao do caso anterior sendo que o EQM estabilizou em torno do valor de 2.59×10^{-3} . Foi observado que esta estabilização sempre ocorria e o valor do EQM dependia de quão longe os parâmetros do novo modelo estavam do modelo original.

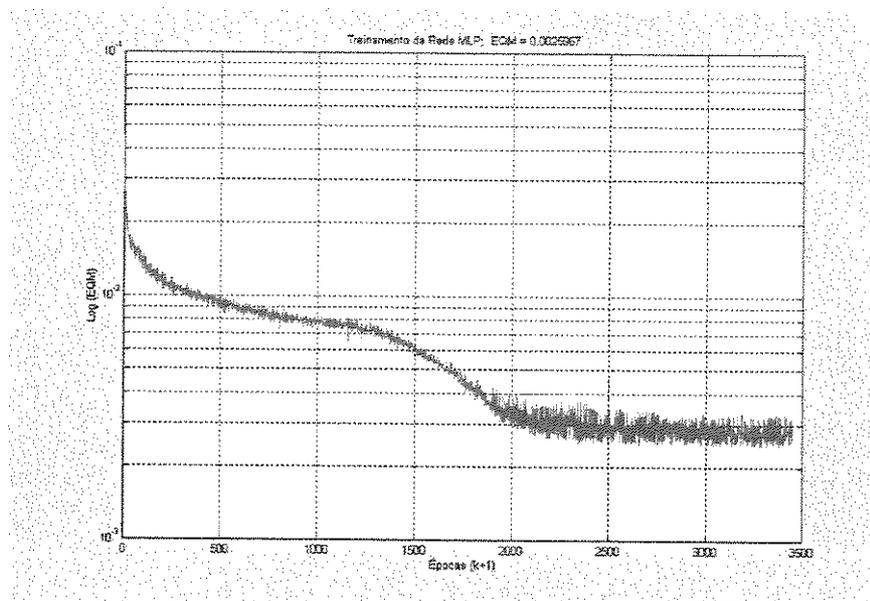


Figura 6.33 - Treinamento do neuro-controlador para o novo modelo de referência.

Após o treinamento foram feitos os testes usando uma excitação e uma condição inicial diferente daquela usada no processo de treinamento, onde obteve-se os seguintes resultados:

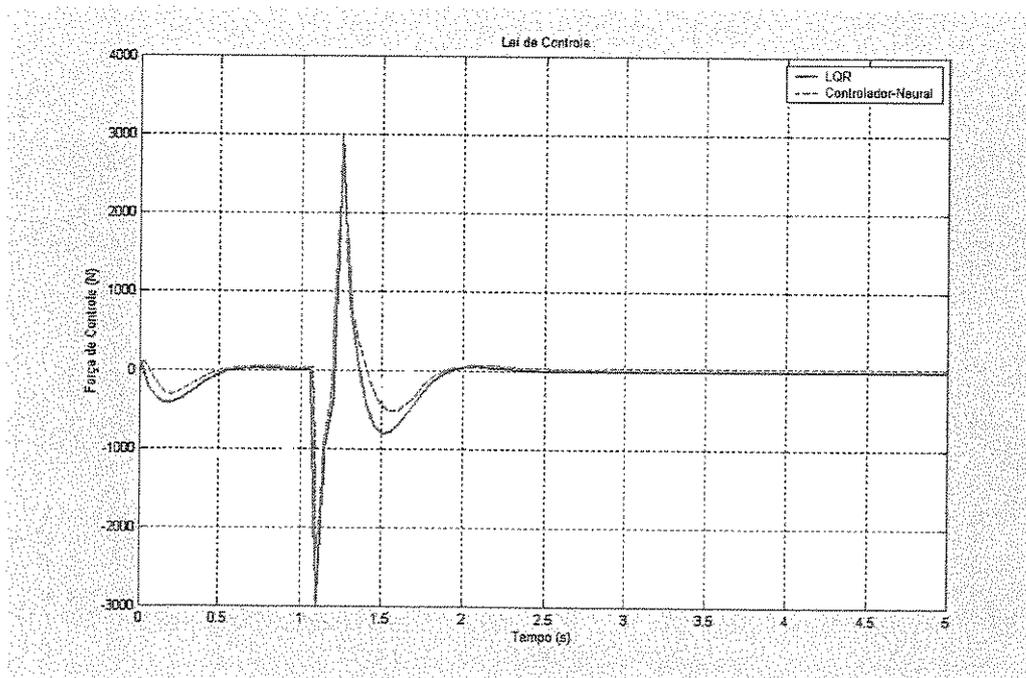


Figura 6.34 – Lei de controle à excitação tipo pulso. $CI = [0.01 \ 0 \ -0.2 \ 0]$.

A lei de controle gerada pelo neuro-controlador a partir do novo modelo de referência apresenta algumas diferenças de amplitude. Isto está relacionado com a estabilização do erro durante o treinamento do neuro-controlador, assim como também às imprecisões nas simulações comentadas anteriormente. Deve-se lembrar que o objetivo do neuro-controlador é gerar uma lei de controle para produzir no sistema original um comportamento similar ao do novo modelo de referência através de um processo de aproximação inversa da dinâmica do sistema.

Aplicando-se esta lei de controle ao sistema original e comparando com a resposta do novo modelo de referência, tem-se:

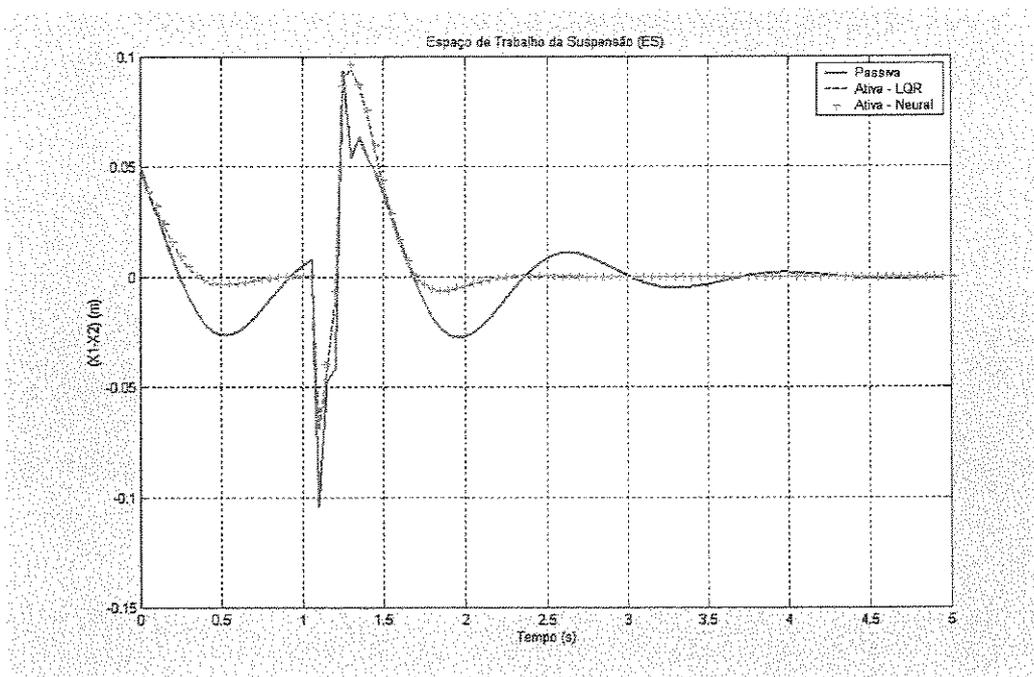


Figura 6.35 – Resposta à excitação tipo pulso (espaço de trabalho da suspensão).

$$CI = [0.05 \ 0 \ -0.2 \ 0]$$

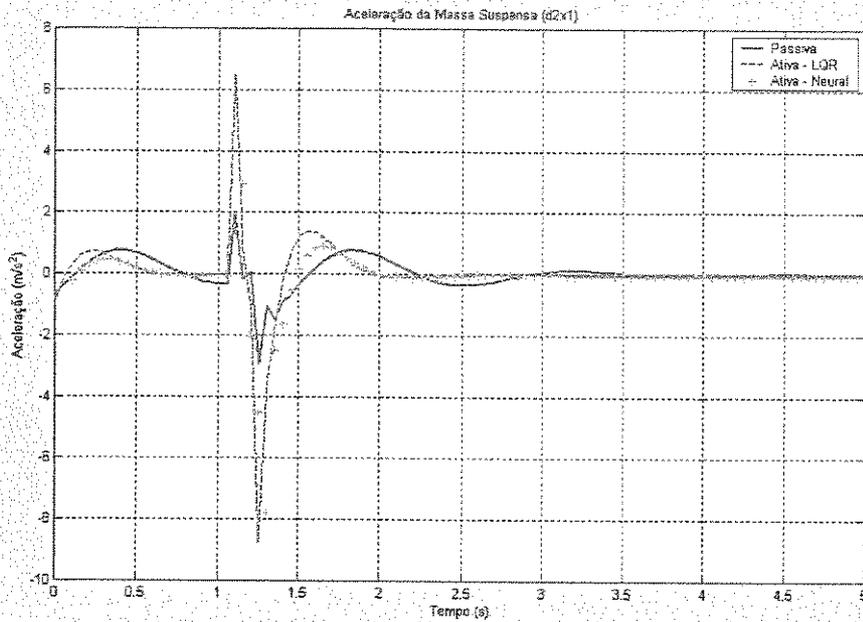


Figura 6.36 – Resposta à excitação tipo pulso (aceleração da massa suspensa).

$$CI = [0.05 \ 0 \ -0.2 \ 0]$$

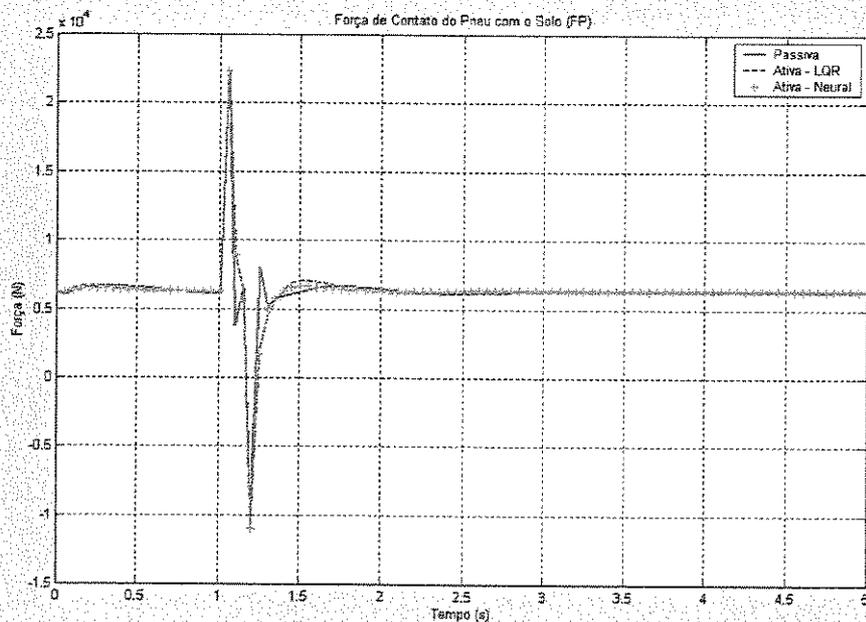


Figura 6.37 – Resposta à excitação tipo pulso (força de contato do pneu com a via).

$$CI = [0.05 \ 0 \ -0.2 \ 0]$$

Observa-se que o modelo neural consegue acompanhar a resposta do sistema ativo com uma boa precisão e desempenho superior ao do sistema passivo. No caso da aceleração (figura 6.36), a resposta neural apresentou valores de pico menores que o sistema ativo, para o novo modelo de referência.

Em seguida foi realizado outro teste. Partindo do sistema original, com o neuro-veículo e o neuro-controlador já treinados (para o sistema original não linear) foi analisado o comportamento da rede neural quando ocorre uma modificação de um dos padrões originais de treinamento. A massa suspensa teve seu valor modificado para, $M_s = 900 \text{ kg}$, simulando uma situação que pode ocorrer em casos práticos. Foram obtidos os seguintes resultados:

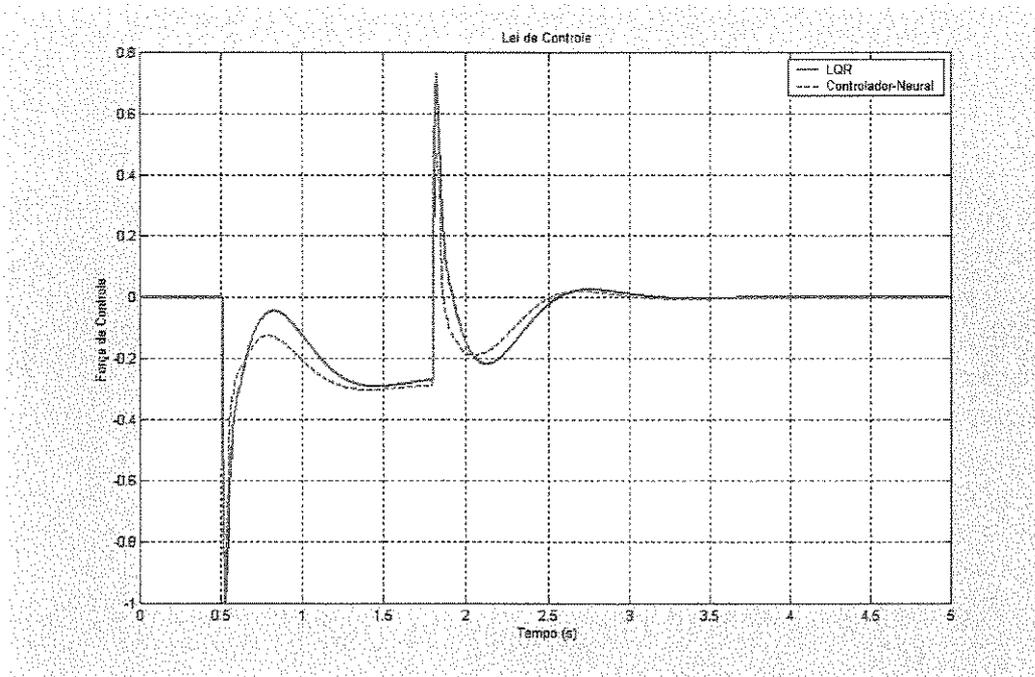


Figura 6.38 – Lei de controle à excitação em lombada. $CI = [0 \ 0 \ 0 \ 0]$.

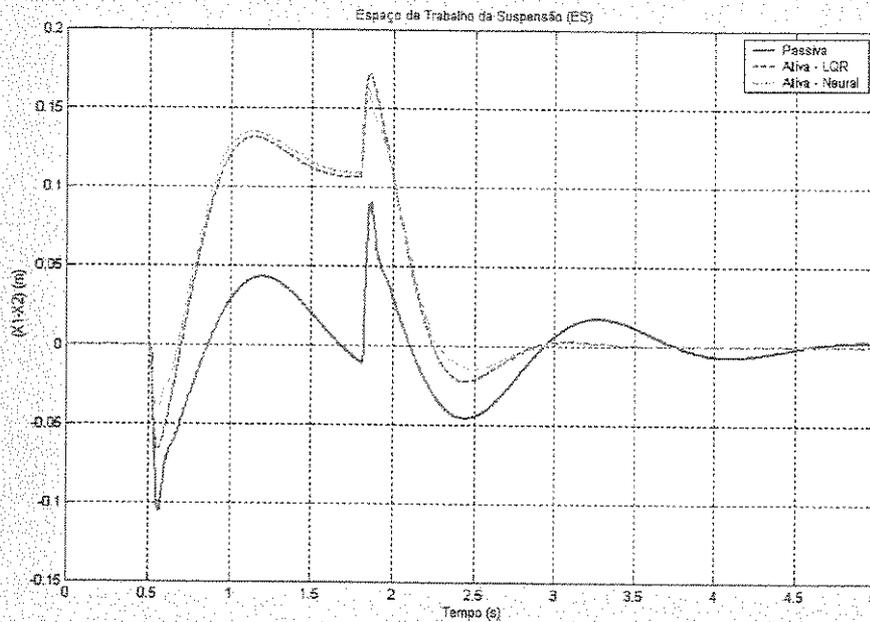


Figura 6.39 – Resposta à excitação em lombada (espaço de trabalho da suspensão).

$$CI = [0 \ 0 \ 0 \ 0]$$

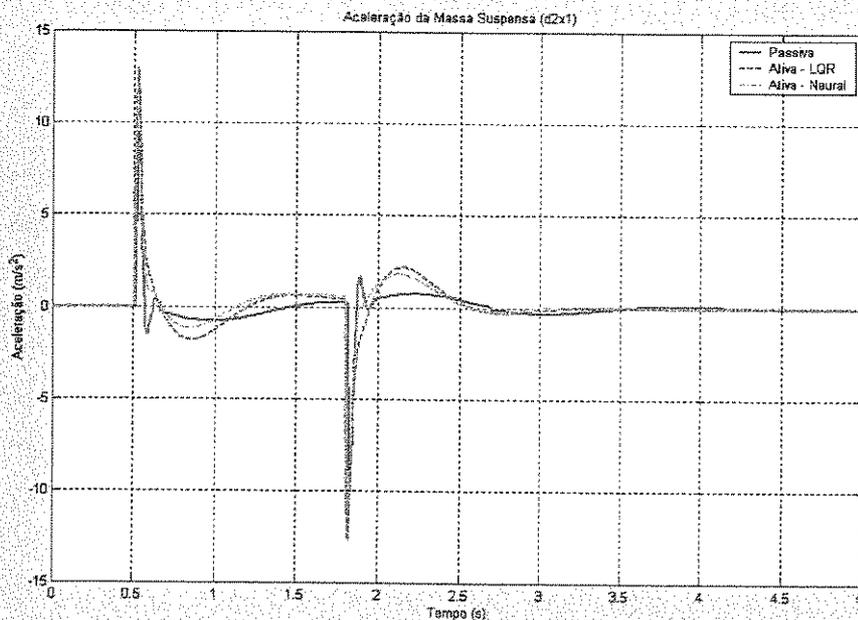


Figura 6.40 – Resposta à excitação em lombada (aceleração da massa suspensa).

$$CI = [0 \ 0 \ 0 \ 0]$$

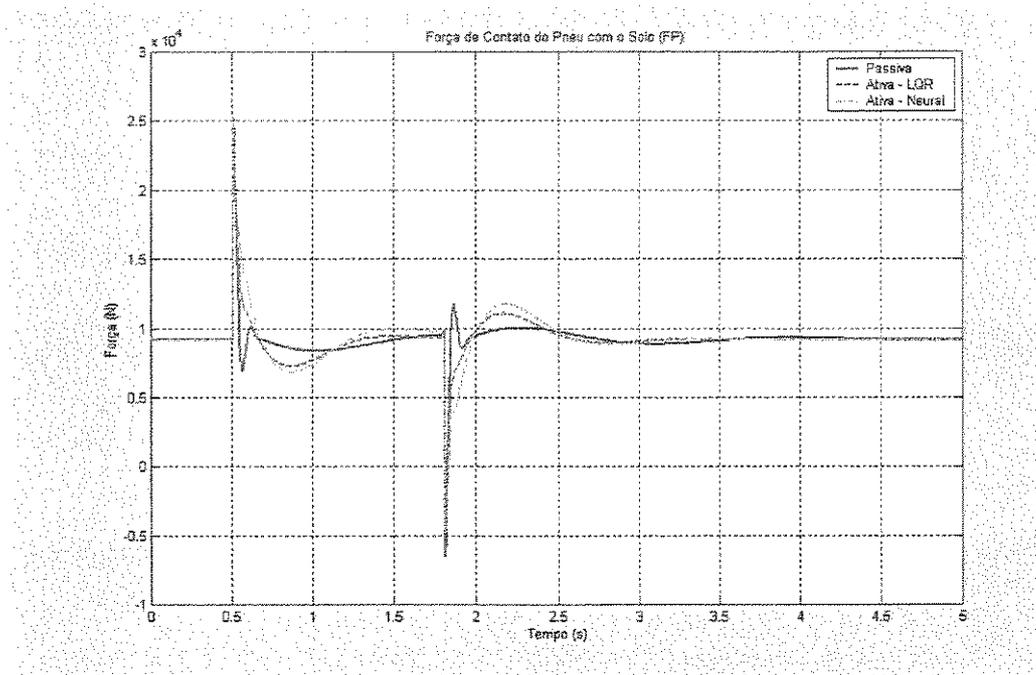


Figura 6.41 – Resposta à excitação em lombada (força de contato do pneu com a via).

$$CI = [0 \ 0 \ 0 \ 0]$$

Observando os gráficos anteriores pode-se concluir que o neuro-controlador é capaz de tolerar alterações nos parâmetros do modelo original usado no treinamento, dentro de certos limites. Isto justifica a utilização das redes neurais na identificação e no controle de sistemas não lineares, cujos parâmetros internos são impossíveis de serem determinados com precisão, onde através de sua capacidade de adaptar-se e de sua capacidade de generalização, as redes conseguem um desempenho melhor que o **LQR** (neste caso), conforme pode ser observado.

O modelo neural consegue acompanhar a resposta do sistema ativo com uma boa precisão, tendo uma melhoria nas amplitudes e com um amortecimento mais rápido, como também com um desempenho superior quando comparado com o sistema passivo. No três casos, a resposta neural apresentou valores de pico menores que o sistema ativo, onde não deve-se deixar de mencionar mais uma vez, que tais resultados contam com a presença do erro estacionário, o que

afeta a precisão das respostas, principalmente em relação ao desempenho dos modelos de suspensão ativo com o **LQR** e do modelo neural, quando comparados com o modelo passivo.

O objetivo principal destas simulações foi o de verificar se o neuro-controlador é capaz de através de um processo de aproximação inversa gerar uma lei de controle idêntica ou próxima à do modelo de referência adotado no treinamento, para com isso conduzir o sistema não linear a um comportamento próximo ao do tal modelo, atendendo assim os critérios de projeto de uma suspensão automotiva.

Capítulo 7

Conclusões e Perspectivas Futuras

Neste trabalho foi feito um estudo da dinâmica de um modelo de suspensão automotiva de $\frac{1}{4}$ de veículo. Considerou-se inicialmente um modelo cujos componentes possuíam características lineares, para em seguida ser introduzido um elemento não linear no sistema de suspensão. Utilizando a teoria de redes neurais artificiais, foi realizada a identificação e o controle da dinâmica do sistema ativo, demonstrando a capacidade das **RNA's** no tratamento de tais problemas, sobretudo quando estes apresentam características altamente não lineares.

Inicialmente foi feita a identificação do modelo de suspensão ativa, usando uma rede neural com três camadas, sendo uma intermediária e o **SCGM**, um algoritmo de aprendizado baseado em um método de segunda ordem, para o treinamento dos pesos sinápticos do chamado “neuro-veículo”. Durante o treinamento o **SCGM** demonstrou a sua eficiência, sobretudo quando os conjuntos de padrões de treinamento apresentavam dimensões elevadas, garantindo um custo computacional menor, quando comparado com métodos de treinamento padrões, como o *Backpropagation* por exemplo. Após o seu treinamento, o neuro-veículo conseguiu identificar a dinâmica tanto do sistema de suspensão linear, como também para o caso do sistema com características não lineares, sem que fossem necessárias técnicas de linearizações, permitindo assim uma identificação de tais modelos, mais próxima das condições reais de operação. A finalidade do neuro-veículo é retropropagar o erro durante o treinamento dos pesos de controlador neural proposto.

Em seguida foi desenvolvido um controlador neural para o sistema ativo de suspensão baseado em um dado modelo de referência, onde através de um processo de aproximação inversa, após seu treinamento, consegue gerar uma lei de controle que conduzirá o sistema a um desempenho próximo ao do modelo de referência. Uma rede neural composta de quatro camadas, com duas camadas intermediárias, foi utilizada para representar o controlador. O algoritmo de treinamento usado foi o *backpropagation*, devido a sua facilidade de implementação, uma vez que o mesmo deveria ser modificado, para permitir a retropropagação do erro desde a saída do neuro-veículo, utilizando os pesos do mesmo já treinados, para atualizar os pesos de neuro-controlador até que uma dada função de custo fosse minimizada. Isto implicou em um aumento do tempo de treinamento para valores de limiares maiores, que os alcançados durante o treinamento do neuro-veículo, o que já era esperado devido as limitações do algoritmo utilizado.

A escolha do modelo de referência baseou-se na teoria de controle ótimo (**LQR**), onde a lei de controle para este modelo foi baseada na realimentação dos estados através da equação de **Riccati**. As matrizes de ponderação foram obtidas a partir de parâmetros representativos do comportamento desejado do sistema (aceleração vertical da massa suspensa, a força de contato do pneu com a via e o espaço de trabalho da suspensão). Foi observada a presença de um erro estacionário, que poderia ser eliminado pela inclusão de integradores durante a análise e simulações de tais sistemas.

Os resultados das simulações apresentadas no capítulo 6 mostraram que a solução proposta para o problema de controle da dinâmica de uma suspensão ativa veicular é factível. A estrutura de controle sugerida, contendo um identificador e controlador a base de redes neurais artificiais, foi capaz de realizar, primeiro uma identificação inversa para obtenção da lei de controle, para em seguida, realizar um controle eficiente, atendendo a especificações de desempenho pré estabelecidas. As redes demonstraram suas capacidades de adaptatividade e generalização, sobretudo quando os parâmetros empregados na análise variavam. Os desenvolvimentos teóricos feitos no decorrer do trabalho com certeza fornecem os subsídios básicos para a continuidade de trabalhos de pesquisas em sistemas de identificação e controle através das **RNA's** para sistemas dinâmicos não lineares.

Como propostas para trabalhos futuros pode-se sugerir a substituição do modelo estático de rede, adotado anteriormente, por redes recorrentes, uma vez que as mesmas apresentam um desempenho superior, principalmente na presença de ruído, que contaminam os sinais colhidos pelos sensores e que serão posteriormente utilizados no treinamento, como também nas simulações ou experimentos propostos. Outra sugestão é aplicar um método de treinamento de segunda ordem no ajuste dos pesos sinápticos do neuro-controlador, a fim de diminuir o custo computacional de treinamento da rede, como também atingir uma precisão maior nos resultados. Deve-se sugerir também a aplicação de integradores a fim de se eliminar o erro estacionário, permitindo assim um melhor desempenho do sistema ativo de suspensão, quando comparado com o sistema passivo.

Referências Bibliográficas

- Alves, Paulo Sérgio Lima. *Análise Comparativa do Desempenho de uma Suspensão Veicular Considerando Elementos Passivos e Ativos*. Campinas-SP: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1997. 131 p. Dissertação (Mestrado)
- Balakrishnan, S. N., Weil, R. D. Neurocontrol: a literature survey. *Mathl. Comput. Modelling*, v. 23, ns. 1/2, p. 101-117, 1996
- Cerqueira, J. J. F. *Análise de Uma Classe de Neurônios Artificiais para Aplicações em Sistemas Robóticos*. Campinas – SP: Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 1996. 103 p. Tese (Mestrado)
- CETEDEM – Centro de Treinamento e Desenvolvimento Monroe. Guia técnico do expert em amortecedores, 2000
- Chen, C. T., Chang, W. D. A feedforward neural network with function shape autotuning. *Neural Networks*, v. 9, n. 4, p. 627-641, 1996
- Choromanski, W. Application of neural network for intelligent wheelset and railway vehicle suspension designs. *Vehicle System Dynamics Supplement*, v. 25, p. 87-98, 1996
- De Castro, L. N. *Análise e síntese de estratégias de aprendizado para redes neurais artificiais*. Campinas – SP: Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 1998a. 248 p. Tese (Mestrado)

- De Castro, L. N., et. al. Feedforward neural network initialization: na evolutionary approach. *In: Vth Brazilian Symposium on Neural Networks, 12, 1998, Belo Horizonte, Brazil. Proceedings IEEE - Computer Society*, p. 43-48, 1998b
- De Castro, L. N., Von Zuben, F. J. A hybrid paradigm for weight initialization in supervised feedforward neural network learning. *International Computer Symposium – Workshop on Artificial Intelligence*, p. 30-37, 1998c
- De Castro, L. N., Von Zuben, F. J. Optimised training techniques for feedforward neural networks. *Technical Report DCA-RT 03/98*, 1998d
- El-Gindy, M. Neural networks representation of a vehicle model: ‘Neuro-vehicle (NV)’. *Int. Journal of Vehicle Design*, v.17, n. 1, p. 55-75, 1996
- El-Gindy, M., Palkovics, L. Possible application of artificial neural networks to vehicle dynamics and control: a literature review. *Int. Journal of Vehicle Design*, v.14, ns. 5/6, p. 592-614, 1993
- Elsken, T. Smaller nets may perform better: special transfer functions. *Neural Networks*, v. 12, p. 627-645, 1999
- Gupta, J. N. D., Sexton, R. Comparing backpropagation with a genetic algorithm for neural network training. *Omega*, v. 27, p. 679-684, 1999
- Haykin, Simon. *Neural Networks*. 2. ed. Reading: PTR Prentice-Hall, 1999
- Hrovat, D. Survey of advanced suspension developments and related optimal control applications. *Automatica*, v. 33, n. 10, p. 1781-1817, 1997

- Hwang, S. H., et. al. Design and evaluation of semi-active suspension control algorithms using hardware-in-the-loop simulations. *Int. Journal of Vehicle Design*, v.19, n. 4, p. 540-551, 1998
- Iyoda, Eduardo Masato. *Inteligência Computacional no Projeto Automático de Redes Neurais Híbridas e Redes Neurofuzzy Heterogêneas*. Campinas – SP: Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 2000. 162 p. Tese (Mestrado)
- Juang, Jer-Nan. *Applied System Identification*. Reading: PTR Prentice-Hall, 1994. Cap.2: Time-Domain Models, p. 15-35
- Karnopp, D., et. al. Vibration control using semi-active force generators. *Journal of Engineering for Industry*, p. 619-626, 1974
- Kasparian, V., Batur, C. Model reference based neural network adaptive controller. *ISA Transactions*, v. 37, p. 21-39, 1998
- Krishnapura, V. G., Jutan, A. A neural adaptive controller. *Chemical Engineering Science*, v. 55, p. 3803-3812, 2000
- MacLoone, S., Irwin, G. W. Fast parallel off-line training of multilayer perceptrons. *IEEE Transactions on Neural Networks*, v. 8, n. 3, p. 646-653, may 1997
- Martins, F. G., Coelho, M. A. N. Application of feedforward artificial neural networks to improve process control of PID-based control algorithms. *Computers and Chemical Engineering*, v. 24, p. 853-858, 2000
- Mitsui, Y., et. al. Application of artificial neural networks to load identification. *Computers and Structures*, v. 69, p. 63-78, 1998

- Moran, A. , et. al. Identification and control of nonlinear active pneumatic suspension for railway vehicles, using neural networks. *Control Eng. Praticce*, v. 5, n. 8, p. 1137-1144, 1997
- Moran, A. , Nagai, M. Optimal active control of nonlinear vehicle suspensions using neural networks. *JSME International Journal*, serie C, v. 37, n. 4, p. 707-718, 1994
- Moran, A. , Nagai, M. Optimal preview control of rear suspension using nonlinear neural networks. *Vehicle System Dynamics*, v. 22, p. 321-334, 1993
- Müller, P. C., Schiehlen, W. O. *Lineare Schwingungen*. Reading: Akademische Verlagsgesellschaft, 1976
- Nagai, et. al. Nonlinear design approach to four-wheel-steering systems using neural networks. *Vehicle System Dynamics*, v. 24, p. 329-342, 1995
- Nagai, M. Researches on active suspensions for ground vehicles. *JSME International Journal*, serie C, v. 36, n. 2, p. 161-170, 1993
- Narendra, K. S. , Parthasarathy, K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, v. 1, n. 1, p. 4-27, march 1990
- Nguyen, D. H., Widrow, B. Neural networks for self-learning control ssysyems. *Int. J. Control*, v. 54, n. 6, p. 1439-1451, 1991
- Ogata, Katsuhiko. *Engenharia de Controle Moderno*. 3 ed. Reading: PTR Prentice-Hall do Brasil, 1998
- Park, Y., Choi, M., Lee, K. Y. An optimal tracking neuro-controller for nonlinear dynamic systems. *IEEE Transactions on Neural Networks*, v. 7, n. 5, p. 1099-1110, september 1996
- Pham, D. T., Oh, S. J. Identification of plant inverse dynamics using neural networks. *Artificial Intelligence in Engineering*, v. 13, p. 309-320, 1999

- Picado, Ricardo Migueis. *Controle Semi-ativo de Suspensões Automotivas*. Campinas-SP: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1998. 87 p. Dissertação (Mestrado)
- Poznyak, A., et. al. Stability analysis of dynamic neural control. *Expert Systems With Applications*, v. 14, p. 227-236, 1998
- Rivals, et. al. Modeling and control of mobile robots and intelligent vehicles by neural networks. *Intelligent Vehicles Symposium, Proceedings, IEEE*, 1994. p. 137-142
- Safak, K. K., Turkay, O. S. Experimental identification of universal motor dynamics using neural networks. *Mechatronics*, v. 10, p. 881-896, 2000
- Stroeve, S. An analysis of learning control by backpropagation through time. *Neural Networks*, v. 11, p. 709-721, 1998
- Tamura, S., Tateishi M. Capabilities of a four-layered feedforward neural network: four layers versus three. *IEEE Transactions on Neural Networks*, v. 8, n. 2, p. 251-255, march 1997
- Tan, H. S., Bradshaw, T. Model identification of an automotive hydraulic active suspension system. *AACC*, 1997
- Truscott, A. J., Wellstead, P. E. Adaptive ride control in active suspension system. *Vehicle System Dynamics*, v. 24, p. 197-230, 1995
- Verma, Brijesh. Fast training of multilayer perceptrons. *IEEE Transactions on Neural Networks*, v. 8, n. 6, p. 1314-1320, november 1997
- Vogl, T. P., et. al. Accelerating the convergence of the back-propagation method. *Biological Cybernetics*, v. 59, p. 257-263, 1988

- Von Zuben, F. J. *Redes Neurais I*. Campinas-SP: Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 2 ° semestre de 1999. Notas de aula.
- Watanabe, Y. , Sharp, R. S. Neural network learning control of automotive active suspension systems. *Int. Journal of Vehicle Design*, v.21, ns. 2/3, p. 124-147, 1999
- Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, v. 78, n. 10, p. 1550-1560, october 1990
- Williams, R. A. Automotive active suspensions. Part 1: basic principles. Part 2: practical considerations. *Proc. Instn. Mech. Engrs.*, v. 211, part D, p. 415-444, 1997
- Zadeh, A. G., Fahim, A., El-Gindy, M. Neural network and fuzzy logic applications to vehicle systems: literature survey. *Int. Journal of Vehicle Design*, v.18, n. 2, p. 132-193, 1997
- Zhang, T., et. al. Design and performance analysis of a direct adaptive controller for nonlinear systems. *Automatica*, v. 35, p. 1809-1817, 1999

Anexos

A.1 – Obtenção das Matrizes de Ponderação

Definindo-se um sistema controlável como aquele que pode alterar todos os estados do sistema através da força de controle, tem-se que para o sistema de ciclo fechado com realimentação de estado através da matriz de ganho calculada acima ser estável, é necessário antes de utilizar a equação de Ricatti verificar se o sistema é controlável ou não. Isto se deve ao fato de que caso o sistema seja não controlável, haverá estados que não poderão ser controlados através dos elementos ativos, embora esta força envolva todas as variáveis de estado (Alves, 1997). Utilizando o teorema que comprove a controlabilidade ou não do sistema através da construção da matriz de controlabilidade \mathbf{M} , definida como função das matrizes \mathbf{A} e \mathbf{B} , tem-se:

$$\mathbf{M} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{(n-1)}\mathbf{B}] \quad (\text{A.1.1})$$

onde n representa a ordem do sistema. O Sistema é então dito controlável se a matriz \mathbf{M} tiver *rank* igual a ordem do vetor estado.

A segunda condição para estabilidade deste sistema de ciclo fechado é que o mesmo seja observável, isto é, se cada variável de estado puder ser determinada a partir da saída do sistema. Utilizando o teorema que verifique a observabilidade do sistema através da construção de uma matriz de observabilidade \mathbf{Q}_e , tem-se:

$$Q_c = [C \quad A^T C \quad (A^T)^2 C \quad \dots \quad (A^T)^{(n-1)} C] \quad (A.1.2)$$

O sistema é então dito observável se a matriz Q_c tiver *rank* igual a ordem do vetor de estado.

Sendo o sistema totalmente controlável e observável, logo, todos os estados do sistema são estabilizados através da matriz de ganho obtida pela equação de **Ricatti** e o sistema pode ser otimizado para as matrizes de ponderação consideradas.

O primeiro fator a ser ponderado é o conforto do motorista. Para isso, é necessário a sua representação através de um parâmetro que lhe sirva de medida. Considerando a aceleração como uma medida indireta do desconforto, tem-se então a aceleração vertical da massa suspensa como parâmetro a ser ponderado. Como as variáveis de estado são as coordenadas generalizadas de posição e velocidade, a representação deste parâmetro de ponderação \ddot{X}_1 é feita através de um vetor que multiplica a derivada de primeira ordem do vetor de estado (Alves, 1997).

$$P_{\ddot{X}_1} = C_{\ddot{X}_1} * \dot{z} \quad (A.1.3)$$

Onde,

$$C_{\ddot{X}_1} = [0 \quad 0 \quad 1 \quad 0] \quad (A.1.4)$$

Conforme a equação 3.35, as ponderações na função custo são na forma quadrática, assim as matrizes são obtidas fazendo-se:

$$P_{\ddot{X}_1}^2 = \dot{z}^T * C_{\ddot{X}_1}^T * C_{\ddot{X}_1} * \dot{z} \quad (A.1.5)$$

Como as ponderações são entre as variáveis de estado e o controlador ativo, torna-se necessário eliminar a derivada do vetor de estado na equação acima substituindo-se a equação 3.23 sem o termo de excitação:

$$\dot{z} = A * z + B * u \quad (\text{A.1.6})$$

Assim teremos,

$$P_{\ddot{X}_1}^2 = (A * z + B * u)^T * C_{\ddot{X}_1}^T * C_{\ddot{X}_1} * (A * z + B * u) \quad (\text{A.1.7})$$

Através da propriedade da transposta do produto de matrizes, temos:

$$P_{\ddot{X}_1}^2 = (z^T * A^T + u^T * B^T) * C_{\ddot{X}_1}^T * C_{\ddot{X}_1} * (A * z + B * u) \quad (\text{A.1.8})$$

Efetuando os produtos:

$$P_{\ddot{X}_1}^2 = z^T A^T C_{\ddot{X}_1}^T C_{\ddot{X}_1} A z + z^T A^T C_{\ddot{X}_1}^T C_{\ddot{X}_1} B u + u^T B^T C_{\ddot{X}_1}^T C_{\ddot{X}_1} A z + u^T B^T C_{\ddot{X}_1}^T C_{\ddot{X}_1} B u \quad (\text{A.1.9})$$

Fazendo,

$$\begin{aligned} Q_{\ddot{X}_1} &= A^T C_{\ddot{X}_1}^T C_{\ddot{X}_1} A \\ N_{\ddot{X}_1} &= A^T C_{\ddot{X}_1}^T C_{\ddot{X}_1} B \\ R_{\ddot{X}_1} &= B^T C_{\ddot{X}_1}^T C_{\ddot{X}_1} B \end{aligned} \quad (\text{A.1.10})$$

Tem-se finalmente,

$$P_{\ddot{X}_1}^2 = z^T Q_{\ddot{X}_1} z + z^T N_{\ddot{X}_1} u + u^T N_{\ddot{X}_1}^T z + u^T R_{\ddot{X}_1} u \quad (\text{A.1.11})$$

As matrizes de ponderação da aceleração vertical da massa suspensa são obtidas substituindo-se as matrizes **A** e **B** dadas pelas equações 3.4 e 3.24 e o vetor $C_{\ddot{x}_1}$ dado pela equação A.1.4 na equação A.1.10 (Alves, 1997).

Com a ponderação da aceleração vertical da massa suspensa, a mesma é então minimizada, implicando num aumento do conforto do veículo. Isto ocorre em suspensões com pequena rigidez, pois tem-se grandes deflexões da mola, como é o caso de suspensões de carros de luxo. Porém, considerando somente esta ponderação, a otimização levará a um sistema que possua pequena rigidez e deflexões muito grandes. Para que o processo de otimização não leve a uma solução em que as deflexões da suspensão aumentem indefinidamente, é necessário considerar essas deflexões nas ponderações da função custo. Isto é feito através da ponderação do espaço de trabalho da suspensão. A expressão para esta ponderação é dada por:

$$P_{ES} = X_1 - X_2 \quad (A.1.12)$$

Como esta expressão é função dos estados X_1 e X_2 , pode-se então representar esta ponderação através de um vetor que multiplica o vetor de estado, logo:

$$P_{ES} = C_{ES} * z \quad (A.1.13)$$

Onde,

$$C_{ES} = [1 \quad -1 \quad 0 \quad 0] \quad (A.1.14)$$

Para obtenção da matriz de ponderação, é necessário escrever a equação de ponderação acima na forma quadrática, assim:

$$P_{ES}^2 = z^T * C_{ES}^T * C_{ES} * z \quad (A.1.15)$$

Fazendo

$$\mathbf{Q}_{ES} = \mathbf{C}_{ES}^T * \mathbf{C}_{ES} \quad (\text{A.1.16})$$

Substituindo-se a equação A.1.16 em A.1.15 tem-se:

$$\mathbf{P}_{ES}^2 = \mathbf{z}^T * \mathbf{Q}_{ES} * \mathbf{z} \quad (\text{A.1.17})$$

As matrizes de ponderação do espaço de trabalho da suspensão, são:

$$\begin{aligned} \mathbf{Q}_{ES} &= \mathbf{C}_{ES}^T * \mathbf{C}_{ES} \\ \mathbf{N}_{ES} &= \mathbf{0} \\ \mathbf{R}_{ES} &= \mathbf{0} \end{aligned} \quad (\text{A.1.18})$$

Além dos fatores de conforto do motorista e espaço de trabalho da suspensão, é também necessário levar em consideração a segurança do veículo. Conforme feito anteriormente, este fator deve ser representado através de um parâmetro que lhe sirva de medida para que a sua matriz de ponderação possa ser obtida a partir dele. Este parâmetro é representado pela variação dinâmica das forças de contato do pneu com o solo. A sua ponderação fará com que essas variações sejam minimizadas. Isto forçará o sistema a se tornar mais estável e conseqüentemente mais seguro (Alves, 1997). Considerando a força de contato do pneu com o solo, obtém-se a expressão da ponderação a partir das equações abaixo:

$$\mathbf{P}_{Fp} = -k_p * \mathbf{X}_2 \quad (\text{A.1.19})$$

Escrevendo esta expressão como um produto entre um vetor e o vetor de estado, tem-se:

$$\mathbf{P}_{Fp} = \mathbf{C}_{Fp} * \mathbf{z} \quad (\text{A.1.20})$$

Onde,

$$\mathbf{C}_{F_P} = [0 \quad -k_P \quad 0 \quad 0] \quad (\text{A.1.21})$$

De modo similar ao que foi desenvolvido para os casos anteriores, representa-se a equação de ponderação A.1.20 na forma quadrática:

$$\mathbf{P}_{F_P}^2 = \mathbf{z}^T * \mathbf{C}_{F_P}^T * \mathbf{C}_{F_P} * \mathbf{z} \quad (\text{A.1.22})$$

A matriz de ponderação é então dada pelo produto entre o transposto do vetor \mathbf{C}_{F_P} e o próprio vetor:

$$\mathbf{Q}_{F_P} = \mathbf{C}_{F_P}^T * \mathbf{C}_{F_P} \quad (\text{A.1.23})$$

Assim, substituindo a equação A.1.23 em A.1.22, tem-se:

$$\mathbf{P}_{F_P}^2 = \mathbf{z}^T * \mathbf{Q}_{F_P} * \mathbf{z} \quad (\text{A.1.24})$$

As matrizes de ponderação em relação a segurança do veículo, são:

$$\begin{aligned} \mathbf{Q}_{F_P} &= \mathbf{C}_{F_P}^T * \mathbf{C}_{F_P} \\ \mathbf{N}_{F_P} &= \mathbf{0} \\ \mathbf{R}_{F_P} &= \mathbf{0} \end{aligned} \quad (\text{A.1.25})$$

Após a ponderação dos fatores representativos do conforto e da segurança do veículo, falta ainda ponderar a energia do controlador ativo introduzido no sistema, a qual permitirá a

realização do controle (Alves, 1997). Considerando-se novamente a ponderação quadrática tem-se para \mathbf{u} :

$$\mathbf{P}_u = \mathbf{u}^2 = \mathbf{u} * 1 * \mathbf{u} = \mathbf{u} * \mathbf{R}_u * \mathbf{u} \quad (\text{A.1.26})$$

Logo, as matrizes de ponderação em relação a energia do controlador, são:

$$\begin{aligned} \mathbf{Q}_u &= 0 \\ \mathbf{N}_u &= 0 \\ \mathbf{R}_u &= 1 \end{aligned} \quad (\text{A.1.27})$$

Para consideração de todas as ponderações simultaneamente, são utilizadas na função custo, matrizes obtidas a partir da soma de todas as matrizes de ponderação dadas acima (Alves, 1997). Tem-se então:

$$\begin{aligned} \mathbf{Q} &= \mathbf{Q}_{\ddot{\mathbf{x}}_1} + \mathbf{Q}_{ES} + \mathbf{Q}_{Fp} + \mathbf{Q}_u \\ \mathbf{N} &= \mathbf{N}_{\ddot{\mathbf{x}}_1} + \mathbf{N}_{ES} + \mathbf{N}_{Fp} + \mathbf{N}_u \\ \mathbf{R} &= \mathbf{R}_{\ddot{\mathbf{x}}_1} + \mathbf{R}_{ES} + \mathbf{R}_{Fp} + \mathbf{R}_u \end{aligned} \quad (\text{A.1.28})$$

A.2 – Algoritmo de Implementação do *Backpropagation*

A.2.1 - Processo Iterativo I – Método Padrão-a-Padrão (Von Zuben, 1999)

- defina uma condição inicial $\mathbf{w}(0)$ para o vetor de parâmetros ($\mathbf{w} \in \mathfrak{R}^{Np}$);
- atribua um valor fixo para η e defina um escalar $\varepsilon > 0$ arbitrariamente pequeno;
- faça $\mathbf{n} = 0$ e calcule $\mathbf{J}(\mathbf{w}(\mathbf{n}))$;
- enquanto $\sqrt{2 \mathbf{J}(\mathbf{w}(\mathbf{n})) / \mathbf{N}} \geq \varepsilon$, faça:

- ◆ ordene aleatoriamente os padrões de entrada-saída;
- ◆ para l variando de 1 até \mathbf{N} , faça:

apresente o padrão l de entrada à rede;

calcule $\mathbf{J}_l(\mathbf{w}(\mathbf{k}))$ e $\nabla \mathbf{J}_l(\mathbf{w}(\mathbf{n}))$;

$\mathbf{w}(\mathbf{n}+1) = \mathbf{w}(\mathbf{n}) - \eta \nabla \mathbf{J}_l(\mathbf{w}(\mathbf{n}))$;

- ◆ $\mathbf{n}=\mathbf{n}+1$;
- ◆ calcule $\mathbf{J}(\mathbf{w}(\mathbf{n}))$.

A.2.2 - Processo Iterativo II – Método em Lote ou Batelada (Von Zuben, 1999)

- defina uma condição inicial $\mathbf{w}(0)$ para o vetor de parâmetros ($\mathbf{w} \in \mathfrak{R}^{Np}$);
- atribua um valor fixo para η e defina um escalar $\varepsilon > 0$ arbitrariamente pequeno;
- faça $n = 0$ e calcule $J(\mathbf{w}(n))$;
- enquanto $\sqrt{2 J(\mathbf{w}(n)) / N} \geq \varepsilon$, faça:
 - ◆ ordene aleatoriamente os padrões de entrada-saída;
 - ◆ para l variando de 1 até N , faça:
 - apresente o padrão l de entrada à rede;
 - calcule $J_l(\mathbf{w}(k))$ e $\nabla J_l(\mathbf{w}(n))$;
 - $$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta/N \nabla J_l(\mathbf{w}(n));$$
 - ◆ $n=n+1$;
 - ◆ calcule $J(\mathbf{w}(n))$.

* Construção dos vektres iniciais de pesos Woell, Woel2 e Wosl :Camadas intermediárias e camada de saída do controlador respectivamente; Geração aleatória, com média 0 e distribuição uniforme ou então Gaussiana. Distribuição uniforme no intervalo [-0.1,+0.1].

```
if CM==1;
```

```
    Woella=zeros((Nwi1/a),a);
```

```
    Woel2a=zeros((Nwi2/q),q);
```

```
    Wosla=zeros((Nws/o),o);
```

```
end
```

```
Woe11=0.05*(randn((Nwi1/a),a));
```

```
Woe12=0.05*(randn((Nwi2/q),q));
```

```
Wos1=0.05*(randn((Nws/o),o));
```

* Carregando os pesos do identificador já treinado:

```
load pesos1
```

```
Woe2=w1;
```

```
Wos2=w2;
```

* Cálculo (Treinamento da Rede):

k=0; * k é o número de épocas.

EQM=1; * EQM é o erro quadrático médio.

```
while EQM >= Eps;
```

* Ordenamento aleatório dos padrões de entrada e saída.

```
    defindex=randperm(Np);
```

```
    for l=1:Np;
```

```
        index=defindex(1,l);
```

```
        X1i=Xe(index,:);
```

```
        X1c=X1i(1,2:5);
```

```
        Y1=Yde(index,:);
```

```
[Ys2,Vg11,Vg12,Vgs1]=backpropcontrol(a,q,o,m,p,X1i,X1c,Y1,Woe11,Woe12,Wos1,Woe2,Wos2);
```

```
    XEC(l,:)=X1c;
```

```
    YSI(l,:)=Ys2;
```

```
    YDI(l,:)=Y1;
```

```
    if CM==1;
```

* Treinamento via "Método do Gradiente":

1 Camada intermediária do controlador:

```
Woe11=Woe11+Beta*(Woe11-Woella)-alfa*Vg11;
```

2 Camada intermediária do controlador:

```
Woe12=Woe12+Beta*(Woe12-Woel2a)-alfa*Vg12;
```

* Camada de saída do controlador:

```
Wos1=Wos1+Beta*(Wos1-Wosla)-alfa*Vgs1;
```

* Atualização dos pesos:

```

    Wo11a=Wo11;
    Wo12a=Wo12;
    Wos1a=Wos1;
    Wo11=We11;
    Wo12=We12;
    Wos1=Ws1;
else
    % Treinamento via "Metodo do Gradiente":
    % 1ª Camada Intermediária do controlador:
    We11=Wo11-alfa*Vgi11;
    % 2ª Camada Intermediária do controlador:
    We12=Wo12-alfa*Vgi12;
    % Camada de saída do controlador:
    Ws1=Wos1-alfa*Vgs1;
    % Atualização dos pesos:
    Wo11=We11;
    Wo12=We12;
    Wos1=Ws1;
end
end

% Função Custo:

DIF=(YSI-YDI).^2;
SomaC=sum(DIF);
SomaT=sum(SomaC');
JWK=(1/2)*SomaT;

% Critério de Convergência:
EQM=sqrt(2*JWK/Np);
k=k+1;

% Gráfico de controle do treinamento:
ER(k,1)=EQM;
Limite(k,1)=Eps;
figure(2)
semilogy(0:k-1,ER(1:k,1),'m',0:k-1,Limite(1:k,1),'g');
grid
title(['Treinamento da Rede MLP; EQM = 'num2str(EQM)]);
xlabel('Épocas (k+1)');
ylabel('Log (EQM)');
drawnow
if k>Ninter;
    break
end
end
end

```

```

*****
**          REDE NEURAL PERCEPTRON WILTI-LAMNATAS COM BACKPROPAGATION          **
**          CONTROLADOR NEURAL                                             **
**                                                                              **
**          Algoritmo p/ aplicação de Backpropagation                       **
**          Método: Padrão-a-Padrão                                         **
**                                                                              **
**          Aluno:      Antonio Fernando Arceu de Andrade                   **
**                                                                              **
**          DMC/TEM/UNICAMP                                                  **
*****

```

```

function
[ys2,Vgi11,Vgi12,Vgs1]=backpropcontrol(a,q,o,m,p,Xli,Xlc,Yl,Woe11,Woe12,Wos1,W
oe2,Wos2);

```

```

* BACKPROPAGATION: FORWARD - CONTROLADOR

```

```

- 1ª Camada intermediária do controlador:

```

```

X1b11=[1 Xlc]; % Inclusão de "bias".

```

```

Ui11=X1b11*Woe11;

```

```

Zi11=tanh(Ui11);

```

```

- 2ª Camada intermediária do controlador:

```

```

X1b12=[1 Zi11]; % Inclusão de "bias".

```

```

Ui12=X1b12*Woe12;

```

```

Zi12=tanh(Ui12);

```

```

- Camada de saída do controlador:

```

```

Zib1=[1 Zi12]; % Inclusão de "bias".

```

```

Us1=Zib1*Wos1;

```

```

Ys1=Us1;

```

```

- BACKPROPAGATION: FORWARD - IDENTIFICADOR

```

```

* Camada intermediária do identificador:

```

```

X1b2=[1 Ys1 Xli]; % Inclusão de "bias".

```

```

Ui2=X1b2*Woe2;

```

```

Zi2=tanh(Ui2);

```

```

- Camada de saída do identificador:

```

```

Zib2=[1 Zi2]; % Inclusão de "bias".

```

```

Us2=Zib2*Wos2;

```

```

Ys2=Us2;

```

BACKPROPAGATION: BACKWARD - CONTROLADOR

```

erro=Ys2-Y1;

% Valor gradiente:
Vg11=[];
Vg12=[];
Vgs1=[];

% Camada de saída do controlador:
for j=1:m;
    for i=1:p;
        somalpar(1,i)=erro(1,i)*Wos2(j+1,i);
    end
    soma1=sum(somalpar);
    soma2par(1,j)=(1-(Zi2(1,j)^2))*soma1*Woe2(2,j);
end
soma2=sum(soma2par);
Vgs1=soma2*Zib1';

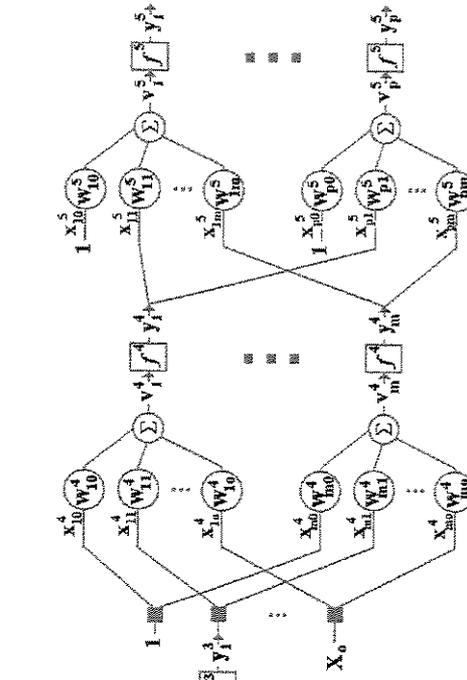
% 2ª Camada intermediária do controlador:
for r=1:q;
    Vg12(:,r)=(1-(Zi12(1,r)^2))*soma2*Wos1(r+1,1)*X1b12';
end

% 1ª Camada intermediária do controlador:
for s=1:a;
    for r=1:q;
        soma3par(1,r)=(1-(Zi12(1,r)^2))*soma2*Wos1(r+1,1)*Woe12(s+1,r);
    end
    soma3=sum(soma3par);
    Vg11(:,s)=(1-(Zi11(1,s)^2))*soma3*X1b11';
End

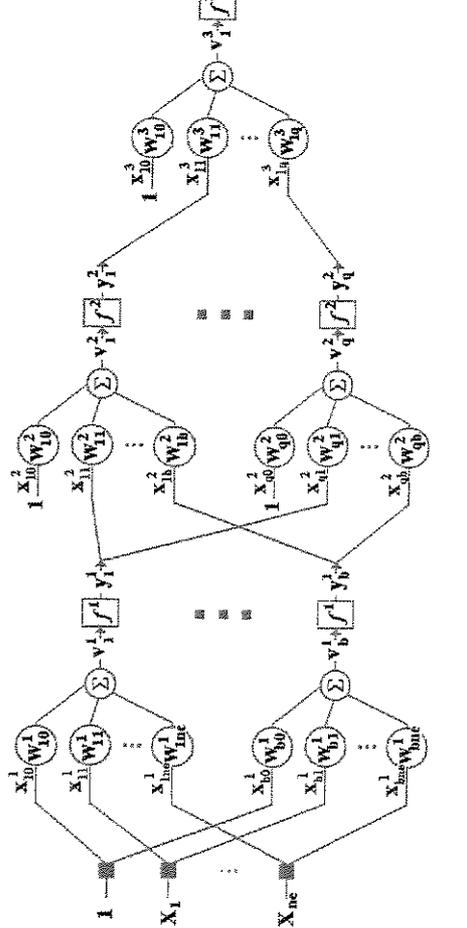
```

A.4 – Redes Neurais

REDE 2: NEURO-VEÍCULO



REDE 1: NEURO-CONTROLADOR



UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE