

Rodrigo Rizzi Starr

Nota final

9,9 (nove e nove)

  
22/01/04

## PROJETO DE SUPERVISORES INTELIGENTES EM CONTROLE DE SISTEMAS NÃO-LINEARES

Texto apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do título de Engenheiro Mecânico.

SÃO PAULO

2003

RODRIGO RIZZI STARR

PROJETO DE SUPERVISORES  
INTELIGENTES EM CONTROLE DE  
SISTEMAS NÃO-LINEARES

Texto apresentado à Escola  
Politécnica da Universidade  
de São Paulo para obtenção  
do título de Engenheiro  
Mecânico.

Orientador:  
Agenor de Toledo Fleury

Dezembro 2003

*À minha mãe, que não pode ver isto pronto.*

## *AGRADECIMENTOS*

Ao orientador, Prof. Dr. Agenor de Toledo Fleury, pelo estímulo, apoio, e pela paciência em suportar meus momentos de teimosia.

Ao Prof. Dr. Fernando e Freitas Jr., pela ajuda com o modelo do motor, e por fornecer os programas originais para a simulação.

Ao Dr. Pedro Sérgio Pereira Lima, pela grande ajuda com relação a redes neurais.

A turma de 98 de mecatrônica, pelo companheirismo, e a todos que de alguma forma ajudaram na realização deste trabalho.

## Resumo

O objetivo deste trabalho é aplicar técnicas de lógica fuzzy e redes neurais no controle de emissões de poluentes em motores de combustão interna movidos a gás natural.

A técnica utilizada consiste em cobrir a região de operação do motor com uma série de controladores LQ, válidos localmente, e utilizar lógica fuzzy para realizar uma interpolação entre esses controladores, sintetizando assim um controlador que apresente um bom desempenho sobre toda a região de operação. Sobre este sistema, utilizou-se um algoritmo de retropropagação para sintonizar o controlador para produzir respostas exatas onde estas fossem conhecidas.

Os resultados das simulação mostraram-se bastante satisfatórios, apresentando um desempenho superior a um controlador utilizando um esquema de chaveamento simples sobre os mesmos controladores LQ. Considera-se que a abordagem é promissora, e possa vir a ser utilizada em outros sistemas com características semelhantes.

## Abstract

The goal of this work is the application of fuzzy logic and neural networks techniques in the control of pollutant gases generated by internal combustion engines powered by natural gas.

The technique used consists in covering the operation region of the engine with a series of LQ controllers, and use fuzzy logic to interpolate these controllers, thus creating a controller that presents a good performance all over the operation region. Upon this system, it was used an backpropagation algorithm to fine tune the controller, so as to produce exact responses where they were known.

The simulation results have been quite satisfactory, showing a performance consistently superior to that of a controller using a simple switching schema over the same LQ controllers. It was considered that this was a good approach, and it may be used in other systems with similar characteristics.

# Sumário

<b>Resumo</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Modelo do motor</b>	<b>4</b>
<b>3 Sistemas de controle não-lineares</b>	<b>7</b>
3.1 Múltiplos modelos e chaveamento . . . . .	9
3.2 Lógica fuzzy . . . . .	12
3.3 Redes Neurais . . . . .	13
<b>4 Lógica fuzzy</b>	<b>15</b>
4.1 Introdução . . . . .	15
4.2 Conjuntos fuzzy . . . . .	18
4.3 Operações básicas . . . . .	19
4.3.1 Conjunto vazio . . . . .	20
4.3.2 Conjunto fuzzy normal . . . . .	20
4.3.3 Igualdade de conjuntos fuzzy . . . . .	20
4.3.4 União de conjuntos fuzzy . . . . .	20
4.3.5 Intersecção de conjuntos fuzzy . . . . .	20
4.3.6 Complemento de um conjunto fuzzy . . . . .	21
4.3.7 Produto de dois conjuntos fuzzy . . . . .	21
4.3.8 Potência de um conjunto fuzzy . . . . .	21
4.3.9 alfa-Cortes . . . . .	21
4.4 Relações fuzzy . . . . .	22
4.5 Números fuzzy . . . . .	22

4.5.1	Adição . . . . .	24
4.5.2	Subtração . . . . .	24
4.5.3	Multiplicação . . . . .	24
4.5.4	Divisão . . . . .	24
4.6	Inferência fuzzy . . . . .	25
4.7	Quantidade de imprecisão . . . . .	26
4.8	Controladores fuzzy . . . . .	27
<b>5</b>	<b>Redes Neurais</b>	<b>30</b>
5.1	Memória associativas . . . . .	30
5.2	Retropropagação . . . . .	32
<b>6</b>	<b>Arquitetura do controlador</b>	<b>34</b>
6.1	Controlador LQ existente . . . . .	34
6.2	Síntese do controlador . . . . .	38
6.2.1	Definição dos parâmetros do controlador . . . . .	42
6.3	Detalhes de implementação . . . . .	52
<b>7</b>	<b>Resultados</b>	<b>54</b>
7.1	Análise dos resultados . . . . .	62
<b>8</b>	<b>Conclusões</b>	<b>63</b>
	<b>Bibliografia</b>	<b>65</b>



# Lista de Figuras

1.1	Gráficos de eficiência do catalizador e emissões do MCI . . . . .	2
3.1	Abrangência de cada modelo com relação ao espaço de estados da planta. . . . .	10
3.2	Arquitetura do sistema de controle com chaveamento . . . . .	11
4.1	Representação clássica da função de pertinência do conjunto “temperatura confortável” . . . . .	17
4.2	Representação fuzzy da função de pertinência do conjunto “temperatura confortável” . . . . .	18
4.3	Comparação entre uma função de pertinência normal, uma não-normal e uma não-convexa . . . . .	23
4.4	Estrutura básica de um controlador fuzzy . . . . .	28
5.1	Modelo geral de uma rede neural associativa de múltiplas camadas	31
6.1	Representação das regiões de validade dos controladores e da linha de chaveamento . . . . .	36
6.2	Ângulo de abertura da borboleta . . . . .	37
6.3	Variação da relação estequiométrica normalizada, medida na borboleta . . . . .	37
6.4	Variação da relação estequiométrica normalizada, medida no escape	37
6.5	Representação de uma função de pertinência gaussiana em um espaço bidimensional . . . . .	40
6.6	Controlador fuzzy-neural . . . . .	41
6.7	Escolha do controlador mais próximo no espaço de estados . . . . .	44

6.8	Simulação para uma variação do ângulo do pedal de 50 a 80 graus, com $\xi = 4$ .	45
6.9	Simulação para uma variação do ângulo do pedal de 50 a 80 graus, com $\xi = 2$ .	46
6.10	Simulação para uma variação do ângulo do pedal de 50 a 80 graus, com $\xi = 0,5$ .	46
6.11	Simulação para uma variação do ângulo do pedal de 50 a 80 graus, com $\xi = 0,3$ .	47
6.12	Diagrama do controlador	48
6.13	Erro médio de treinamento para a o controlador utilizando $\xi = 0,5$	50
6.14	Erro médio de treinamento para a o controlador utilizando $\xi = 0,3$	51
6.15	Erro médio de treinamento para a o controlador utilizando $\xi = 0,3$ , continuando a partir do gráfico 6.14, totalizando 10000 iterações	51
7.1	$\phi_e$ para a simulação 1 (de 50 a 80 graus)	55
7.2	$\phi_e$ para a simulação 1, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3	55
7.3	$\phi_e$ para a simulação 2 (de 85 a 40 graus). Esta é a única simulação em que o controlador LQR apresentou um desempenho superior aos fuzzy-neurais	56
7.4	$\phi_e$ para a simulação 2, ampliada para mostrar a resposta dos controladores de 1 a 3.	56
7.5	$\phi_e$ para a simulação 3 (de 40 a 85 graus)	57
7.6	$\phi_e$ para a simulação 3, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3	57
7.7	$\phi_e$ para a simulação 4 (de 33 a 88 graus)	58
7.8	$\phi_e$ para a simulação 4, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3. É interessante notar como o erro de <i>offset</i> é significativamente maior para o controlador 3. Isto se deve ao problema de interferência de aprendizado	58
7.9	$\phi_e$ para a simulação 5 (de 88 a 33 graus)	59
7.10	$\phi_e$ para a simulação 5, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3.	59

7.11 $\phi_e$ para a simulação 6 (de 50 a 88 graus) . . . . .	60
7.12 $\phi_e$ para a simulação 6, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3. . . . .	60
7.13 $\phi_e$ para a simulação 7 (de 88 a 50 graus) . . . . .	61
7.14 $\phi_e$ para a simulação 7, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3. . . . .	61

## Lista de Tabelas

2.1	Notação utilizada na modelagem do motor . . . . .	6
5.1	Influência do valor de $\alpha$ no aprendizado. . . . .	33
6.1	Valores dos parâmetros de desempenho para o controlador LQ com chaveamento brusco . . . . .	38
6.2	Valores de estabilização . . . . .	43
6.3	Comparação entre o número de operações realizadas para o contro- lador LQ com chaveamento brusco e para o controlador fuzzy-neural	52
7.1	Tabela com os valores utilizados nas simulações e com os parâmet- ros de performance obtidos. . . . .	54

# Capítulo 1

## Introdução

Uma abordagem que recebeu bastante atenção nas últimas duas décadas é a chamada “soft computing”, ou “knowledge based engineering”. Esta abordagem, que agrupa uma série de tecnologias (redes neurais artificiais, lógica fuzzy, algoritmos genéticos etc) se caracteriza pelo uso de mecanismos de processamento de informação e de dados inspirados nos mecanismos utilizados por sistemas biológicos. Desse modo, as redes neurais foram inspiradas no funcionamento do sistema nervoso central, a lógica fuzzy surgiu como um modo de descrever modos de raciocínio utilizando conceitos pouco definidos, e algoritmos genéticos surgiram como um modelo do funcionamento da teoria da evolução.

O objetivo deste trabalho é testar esta abordagem na realização do controle de um problema prático, e comparar os resultados com os obtidos utilizando outras técnicas de controle. Neste aspecto, a tese de Freitas Junior (2003) é um ótimo ponto de partida, pois compara o desempenho de várias técnicas distintas para o controle de um motor a combustão interna movido a gás natural. Portanto decidiu-se utilizar como plataforma de teste o modelo de motor citado nesta tese, e proposto originalmente por Lopes & Fleury (1995).

O modelo do MCI é bastante interessante pois trata-se de um sistema não-linear e de dinâmica complexa, e com objetivos de controle bastante restritivos. Além disso, existe um interesse social muito grande quanto a melhora das tecnologias relacionadas a MCIs à gás natural, pois estes vem se mostrando uma alternativa economicamente e ecologicamente viável aos MCIs movidos à gasolina

ou diesel.

Apesar de ainda ser baseado em uma fonte de energia não renovável, o preço do gás natural é bastante inferior ao de outros combustíveis fósseis. Também a queima deste combustível é mais eficiente, e, para uma mesma massa, ele possui mais energia se comparado à gasolina, por exemplo. Do ponto de vista da emissão de poluentes, vê-se que a queima do gás natural produz uma quantidade de particulados muito menor que o óleo diesel, por exemplo, e também não produz derivados de enxofre, como a gasolina.

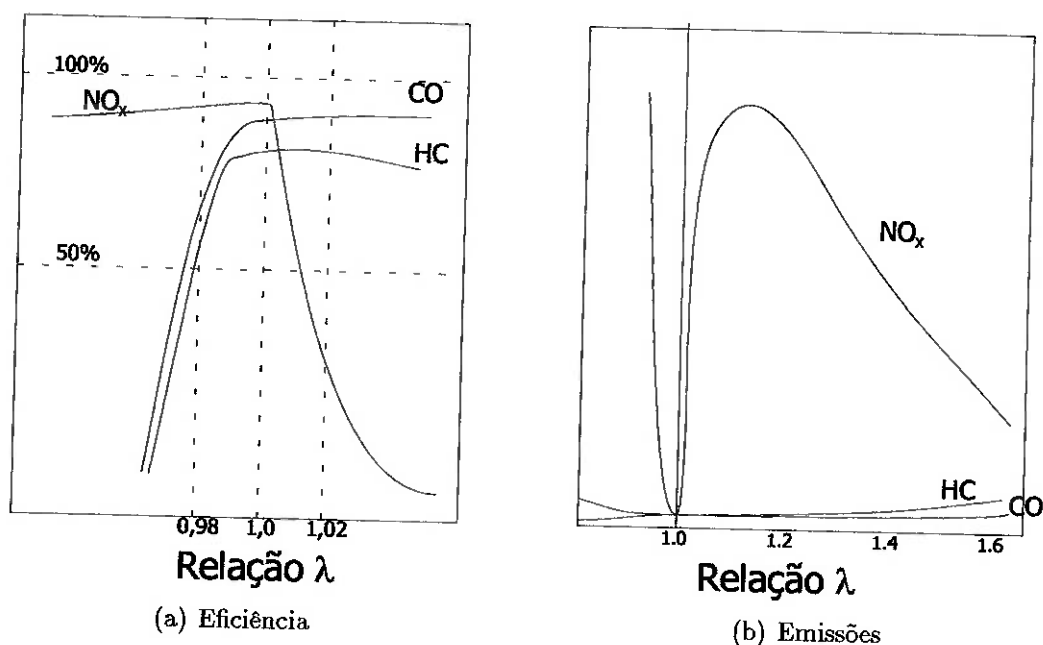


Figura 1.1: Gráficos de eficiência do catalizador e emissões do MCI

No entanto, ainda resta o problema da produção de óxidos de nitrogênio ( $NO_x$ ), hidrocarbonetos (HCs) e monóxido de carbono (CO). Uma solução possível para estes problemas é o uso de catalizadores. O grande problema desta abordagem é que a faixa de eficiência para os catalizadores é muito restrita, sobretudo para a redução do  $NO_x$ . Vê-se na figura 1.1 que para se manter uma eficiência de funcionamento do catalizador em torno de 50%, a variação da relação estequiométrica não deve ser maior que  $\pm 1\%$  da relação ideal. Esta faixa é relativamente simples de ser mantida para um MCI trabalhando em regime, porém, para um MCI sujeito a muitos transitórios, a operação se dá basicamente

fora desta faixa ideal. Deste modo se justifica o esforço de desenvolvimento de um sistema de controle que mantenha a relação estequiométrica da saída em um nível satisfatório.

Este texto está organizado do seguinte modo: No próximo capítulo será dada uma introdução a algumas técnicas de controle linear e não-linear. No capítulo 4 será dada uma introdução sobre lógica fuzzy. No capítulo 5 serão descritas as principais características da rede neural que será utilizada no projeto. No capítulo 2, será mostrado o modelo do motor, que é o sistema a ser controlado. No capítulo 6 será dada uma definição mais rigorosa do problema de controle, e será mostrado o processo de síntese da arquitetura do controlador e da definição dos parâmetros do mesmo. Finalmente, no capítulo 7 serão mostrados os resultados comparativos destes controladores. Finalmente, em 8 estão as principais conclusões obtidas deste estudo.

## Capítulo 2

### Modelo do motor

Este trabalho será realizado sobre um modelo de motor de combustão interna, desenvolvido por Lopes & Fleury (1995). O modelo representa um motor diesel de 6 cilindros, convertido para gás natural, e com injeção eletrônica monoponto. O carregamento do motor é realizado através de uma que aumenta linearmente com a rotação.

Este modelo enfoca sobretudo as características dinâmicas do motor em si, não tendo sido modeladas as perdas de carga no filtro de ar e no escapamento, e também não tendo sido modeladas as dinâmicas dos atuadores nem dos sensores. Portanto, no projeto do sistema de controle, considerar-se-á uma velocidade de atuação infinitamente rápida<sup>1</sup>.

O modelo em si não será mostrado aqui, por ser muito extenso, e estar disponível em outras fontes ((Lopes & Fleury 1995) e (Freitas Junior 2003)). As variáveis de entrada deste modelo são o avanço do ângulo de ignição ( $\beta$ ), a vazão mássica de combustível na borboleta ( $vc_b$ ) e o ângulo de abertura da borboleta ( $\alpha$ ). As variáveis medidas são a razão estequiométrica na borboleta e na saída do motor (respectivamente  $\phi_b$  e  $\phi_e$ ). Este modelo foi implementado como um diagrama de blocos utilizando o Simulink<sup>™</sup>.

Uma das desvantagens de se possuir um modelo implementado apenas em digrama de blocos, ou seja, não analítico, é que fica-se privado de uma série de ferramentas para a análise do comportamento do sistema. Para resolver este

---

<sup>1</sup>Pode-se dizer que isto de certa forma corresponde à realidade, já que o tempo de resposta de um bico injetor é muito mais rápido que as outras dinâmicas envolvidas



problema, Freitas Junior (2003) empreendeu a transformação deste modelo em um modelo analítico no espaço de estados. É este o modelo que será utilizado no presente trabalho. Como algumas funções do modelo original eram descontínuas, para tornar o modelo analítico estas funções foram interpoladas. No entanto isto resultou em pouca diferença nos resultados das simulações.

Este modelo analítico é composto por 6 variáveis de estado, sendo que 4 delas modelam os vários atrasos de transporte presentes, uma é a velocidade de rotação do motor e a outra é a pressão no coletor de admissão. As equações estão representadas abaixo.

$$\dot{x}_1 = -300x_1 - 20000x_2 + vc_b \quad (2.1)$$

$$\dot{x}_2 = x_1 \quad (2.2)$$

$$\dot{x}_3 = -100x_3 + \frac{2530192 \cdot 10^8 \pi}{27np} x_2 \quad (2.3)$$

$$\dot{x}_4 = -200x_4 + 100x_3 \quad (2.4)$$

$$\dot{p} = 4,364 \cdot 10^7 va_b(\alpha, p) - \frac{1,35}{\pi} np \quad (2.5)$$

$$\dot{n} = \frac{1}{120} \left( \frac{8 \cdot 10^{11} x_2}{n} E_f(x_1, x_2, x_3, p, n, \beta) - 2,22n + 66,3 \right) \quad (2.6)$$

As saídas são dadas pelas equações

$$\phi_b = 14,5 \frac{vc_b}{va_b} \quad (2.7)$$

$$\phi_e = 400x_4 - 4nx_3 + \frac{1265093 \cdot 10^6 \pi}{27np} (-x_1 + 200x_2) \quad (2.8)$$

Definindo

$$A(\alpha) = 3 - 2 \cos(\alpha) - \cos^2(\alpha) \quad (2.9)$$

E  $va_b$  é dado por

$$va_b = \begin{cases} 0,000010159375\pi A(\alpha) p^{0,714} \left(1 - \frac{p}{10^5}\right)^{0,143} & \text{Se } p > 5,28 \cdot 10^4 \\ 0,029180921875\pi A(\alpha) & \text{Se } p \leq 5,28 \cdot 10^4 \end{cases} \quad (2.10)$$

A tabela 2.1 explica os símbolos utilizados nestas equações.

Este modelo foi linearizado por Fernando Freitas Jr. ao redor de vários pontos de operação. O critério utilizado foi considerar o ângulo de abertura da borboleta

SÍMBOLO	SIGNIFICADO FÍSICO
$\alpha$	Ângulo de abertura da válvula borboleta
D	Diâmetro da tubulação na válvula borboleta
$A(\alpha)$	Área da seção na válvula borboleta
$\beta$	Ângulo de avanço da ignição (antes do ponto morto superior)
$vc_b$	Vazão de combustível na válvula borboleta
$va_b$	Vazão de ar na válvula borboleta
$vc_a$	Vazão de combustível na válvula de admissão
$va_a$	Vazão de ar na válvula de admissão
$\phi_b$	Relação combustível/ar na válvula borboleta
$\phi_a$	Relação combustível/ar na válvula de admissão
$\phi_c$	Relação combustível/ar na câmara de combustão
$\phi_e$	Relação combustível/ar no escape
n	Rotação do motor
p	Pressão no coletor

Tabela 2.1: Notação utilizada na modelagem do motor

como uma perturbação do ambiente, e encontrar, para cada ambiente diferente, um valor de estabilização. Especificamente, de 30° a 90° de abertura da borboleta, foram feitos 25 modelos, homogeneamente espaçados, e para cada um destes modelos linearizados foi feito um controlador do tipo LQ (realimentação de estados). O objetivo de controle destes controladores é forçar  $\phi_b$  e  $\phi_e$  para 1, ou seja, manter uma relação estequiométrica ideal tanto na admissão quanto no escape. Este trabalho será construído sobre esses controladores.

## Capítulo 3

# Sistemas de controle não-lineares

De um modo geral, a maior parte das técnicas de controle lidam com a solução de problemas lineares. No entanto, o mundo real é essencialmente não linear, e da mesma forma a maior parte dos problemas de interesse em engenharia. Felizmente, para um conjunto relativamente grande desses problemas, pode-se tratá-los como se fossem lineares, e com isso obter-se resultados razoáveis.

Mesmo assim, existem limitações quanto a aplicação de técnicas de controle linear à um sistema não-linear. Por exemplo, um determinado sistema só pode ser linearizado ao redor de um ponto de equilíbrio, e, mesmo assim, o controlador só é válido próximo a este ponto. Caso as não-linearidades da planta sejam “pouco intensas”, a faixa de comportamento linear é extensa e as respostas da planta controlada satisfazem as especificações de performance. No entanto, em muitos casos as não-linearidades são bastante intensas na faixa de operação (e.g. sistemas com folga), ou a faixa de operação do sistema é muito ampla para que a linearização seja válida (e.g. sistemas com saturação).

Assim, para sistemas onde é necessário uma excursão muito grande nas entradas, ou que estão sujeitos a grandes perturbações, ou ainda, onde os parâmetros de performance são muito restritivos, a aproximação do sistema por um modelo linear pode mostrar-se absolutamente incorreta, até levando o sistema à instabilidade. Para este tipo de problema, é necessário utilizar alguma técnica de controle que leve em conta as não-linearidades presentes.

De acordo com (Cook 1993) e (Vincent & Grantham 1997), existem técnicas

que podem ser usadas para reduzir a influencia das não-linearidades na resposta final do sistema, como o uso de um sinal de *dither*, a utilização de uma não-linearidade no controlador de modo a cancelar a não-linearidade da planta, ou a análise utilizando uma função descritora de entrada exponencial.

Um problema com todas essas técnicas é que elas pressupõem um conhecimento perfeito do funcionamento do sistema, ou seja, que existe um conjunto de equações diferenciais analíticas que descreve exatamente o funcionamento do sistema, e que essas equações são conhecidas, se não com exatidão, pelo menos com precisão suficiente para tornar o modelo válido. E mesmo assim, existem casos em que, apesar das equações serem conhecidas, elas são muito complicadas para permitirem a síntese de um controlador satisfatório. Outras vezes a solução pode simplesmente ser tão complexa do ponto de vista numérico a ponto de impedir a implementação prática do controlador, devido a limitações de custo e velocidade de processamento.

Caso o problema a ser resolvido se encaixe em alguma destas categorias, faz-se necessário o uso de técnicas que ataquem especificamente estes problemas. Ou seja, é necessário técnicas que levem em consideração incertezas na definição dos sistemas, ou que permitam abordagens “não analíticas” ao problema, ou ainda que permitam a inclusão de conhecimentos de natureza prática ou intuitiva, mas de qualquer modo imprecisa, sobre o sistema. Abaixo se sumariza algumas técnicas existentes que abordam pelo menos um dos problemas acima.

- Chaveamento entre modelos;
- Controle adaptativo;
- Lógica fuzzy;
- Redes neurais artificiais (RNAs);
- Algoritmos genéticos e sistemas classificadores;
- Fusão de duas ou mais das técnicas anteriores.

Dentre estas técnicas, será usado neste trabalho a fusão entre uma técnica de chaveamento e adaptação proposta por Narendra, Balakrishnan & Ciliz (1995), um método de chaveamento “suave” usando lógica fuzzy proposto por Ruspini, Bonissone & Pedrycz (1998), e um método de aprendizado utilizando uma rede *fuzzy-neural*<sup>1</sup> descrita em (Branco & Dente 1998).

Estas técnicas serão explicadas brevemente aqui, e serão tratadas mais profundamente nos capítulos seguintes.

### 3.1 Múltiplos modelos e chaveamento

A idéia da utilização de múltiplas estratégias de controle para resolver problemas não-lineares tem uma faixa de aplicação muito grande. Por exemplo, (Franklin, Powell & Workman 1997) realiza o posicionamento da cabeça de leitura de um disco rígido utilizando um esquema de chaveamento entre uma lei de controle ótimo do tipo *bang-bang* com um controlador linear.

A idéia básica por traz do uso de múltiplos modelos consiste em cobrir a região de operação do espaço de estados da planta, que denotaremos por  $S_k$ , com  $K$  modelos diferentes e, baseando-se nos erros de previsão de cada modelo, escolher uma estratégia de controle  $C_k$  baseada no modelo com menor erro. Deste modo é possível realizar o controle de um sistema não-linear muito complexo abordando propriedades dinâmicas específicas de cada região do espaço de estados.

De acordo com esta estratégia, deve-se criar um conjunto controladores  $\{C_k\}$ , cada um deles válido em uma pequena região do espaço de estados da planta  $S_k$ , de modo que a união destes controladores cubra toda a região do espaço de estados correspondente a faixa de operação da planta.

Podem ser feitas muitas variações sobre esta idéia básica. Especificamente, Narendra et al. (1995) propõe que o uso de um número relativamente pequeno de modelos lineares, e dois modelos adaptativos permitem ótimos resultados no que se refere a transitórios (devido aos modelos lineares) e erros em regime (devido aos controladores adaptativos). Apesar da proposta deste artigo ser voltada para

<sup>1</sup>Na falta de uma tradução adequada, será utilizado o termo em inglês.

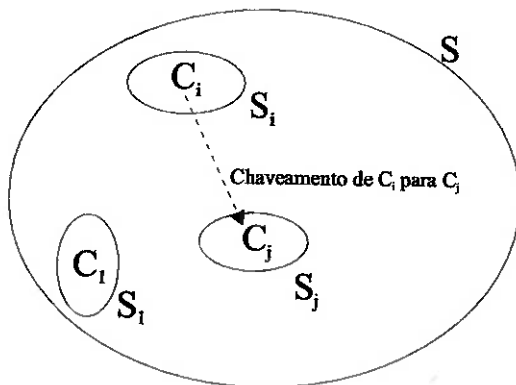


Figura 3.1: Abrangência de cada modelo com relação ao espaço de estados da planta.

plantas com condições ambientais variando rapidamente, acredita-se que esta idéia possa ser transportada para o problema do controle do motor a gás natural, já que a alteração no ângulo de abertura da borboleta pode ser considerada uma alteração ambiental.

Esta técnica parece atrativa para atacar este problema exatamente porque deseja-se obter uma resposta a transientes extremamente rápida, de modo a minimizar a excursão da relação estequiométrica dos gases de escape fora da faixa aceitável de  $\pm 1\%$ . Isto porque motores instalados em veículos de uso predominantemente urbano (o que representa o maior uso dos motores a gás natural) a condição de operação transiente é muito comum.

A arquitetura deste controlador está representada na figura 3.2. Ela é composta por  $N$  modelos de identificação, representados por  $\{I_i\}_{i=1}^N$ . A cada instante, uma medida de performance  $J_i(t)$ ;  $i = 1, 2, \dots, N$  dos erros de identificação  $e_i \triangleq \hat{y}_i - e$  é realizada, e o modelo correspondente a  $\min_i \{J_i(t)\}$  é escolhido para determinar a entrada de controle naquele instante para a planta.

Os modelos utilizados não necessitam obrigatoriamente ser do mesmo tipo. Pode-se utilizar modelos lineares assim como adaptativos ou não-lineares. Também é possível estabelecer alguns modelos que representem o comportamento da planta em situações de falha, e desta forma obter um controlador capaz de identificar falhas e tomar as atitudes mais corretas para a situação.

No mesmo artigo se propõe a seguinte medida de performance dos erros de

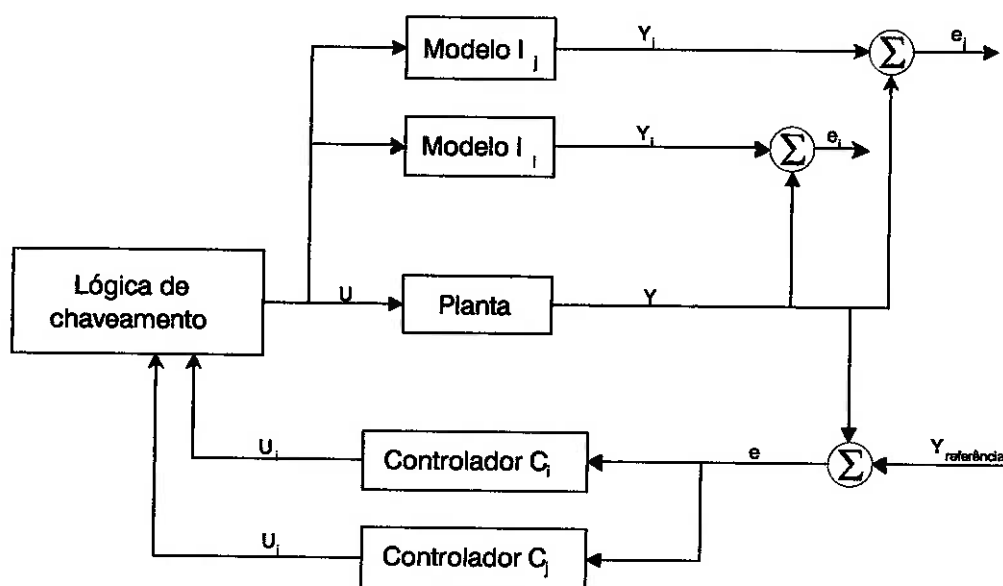


Figura 3.2: Arquitetura do sistema de controle com chaveamento

identificação:

$$J_i(t) = \alpha e_j^2(t) + \beta \int_0^t e^{-\lambda(t-\tau)} e_j^2(\tau) d\tau \quad (3.1)$$

Este índice de performance leva em consideração tanto o erro instantâneo quanto o erro acumulado<sup>2</sup>. Os parâmetros  $\alpha \geq 0$  e  $\beta > 0$  definem a importância relativa destas duas medidas, e o parâmetro  $\lambda > 0$  define o tamanho da “memória” do termo integral<sup>3</sup> (quanto maior  $\lambda$ , menor a memória).

Também se recomenda o uso de histerese, para evitar um chaveamento arbitrariamente rápido entre as leis de controle. Assim, se o par  $(I_j, C_j)$  está sendo usado em um instante  $t$ , e  $J_k(t) = \min_i \{J_i(t)\}$ , então  $(I_j, C_j)$  será retido se  $J_i(t) \leq J_k(t) + \delta$ , e chaveado para  $(I_k, C_k)$  em caso contrário. Nesta regra,  $\delta > 0$  é a constante de histerese.

<sup>2</sup>A integral da equação 3.1 é a convolução das funções  $e^{-\lambda t}$  e  $e_j^2(t)$ , ou seja, pode ser entendida como um filtro passa-baixa com frequência de corte  $\lambda$  aplicado sobre  $e_j^2(t)$ .

<sup>3</sup>i.e.  $1/\lambda$  é a frequência de corte

## 3.2 Lógica fuzzy

Em se tratando do projeto de um sistema de controle, a primeira etapa do projeto normalmente consiste em ganhar um entendimento dos fenômenos que ocorrem na planta a ser controlada e avaliar parâmetros relevantes quantitativamente. Isto normalmente é feito durante o processo de “modelagem” do sistema, que tanto pode ser experimental (utilizando algum método de identificação) quanto analítico (usando matemática e princípios físicos). A maior parte dos sistemas controlados relaciona dados experimentais à sistemas ou modelos. Uma vez obtido o modelo do sistema, é possível executar uma série de procedimentos (como análises matemáticas e simulações) que levam a um melhor entendimento do sistema e a um melhoramento do próprio modelo, e assim iterativamente até se obter um modelo válido.

Ocorrem no entanto situações onde a natureza dos fenômenos envolvidos é muito complexa, e/ou não muito bem compreendida, e para as quais modelos baseados em princípios físicos básicos não são adequados, às vezes nem mesmo possíveis. No entanto, vários destes sistemas são controlados (inclusive estabilizados) satisfatoriamente por operadores humanos (Mandani 1994). Na maior parte dos casos, estes operadores são capazes de transmitir o conhecimento, ou explicar qual o tipo de raciocínio que os leva a tomar as decisões de controle. Portanto, intui-se que, a princípio, se for possível encontrar um modo matematicamente válido de expressar este raciocínio, será possível programar um controlador para resolver esta classe de sistemas.

Neste contexto percebe-se um dos grandes méritos da lógica fuzzy. Ela fornece um ferramental que permite a expressão de enunciados em linguagem comum de uma forma matemática. Com isto é possível enfrentar alguns dos problemas explicitados no parágrafo anterior. Ao se tentar resolver um problema de automação muito complicado, o projetista pode se valer de conhecimentos obtidos de operários experientes no manuseio daquela planta, ou até mesmo do próprio conhecimento qualitativo que este possui sobre o problema.



Em vista desses fatores, a lógica fuzzy se mostra uma alternativa bastante interessante na resolução de problemas não-lineares. No entanto, da mesma forma que qualquer enunciado em qualquer língua é extremamente vago e dependente do contexto, existem uma série de parâmetros em um enunciado fuzzy que devem ser ajustados para que se obtenha um resultado aceitável. Estes parâmetros dizem respeito sobretudo à quantidade de imprecisão<sup>4</sup> em cada termo.

Por se tratar de um dos temas mais importantes deste trabalho, a formalização da lógica fuzzy, e modelos de controladores fuzzy, serão tratados no próximo capítulo.

### 3.3 Redes Neurais

Do mesmo modo que a lógica fuzzy foi inspirada como um modo de modelar um raciocínio utilizando conceitos definidos de forma imprecisa, as redes neurais surgiram como uma tentativa de modelar a capacidade de aprender observada nos sistemas nervosos dos animais superiores. O primeiro modelo de rede neural que obteve algum sucesso no âmbito da engenharia foi a *perceptron*, de Rosenblatt, que modela a retina humana (citado em (Tsoukalas & Uhrig 1996)).

O campo das redes neurais é bastante amplo, existindo uma infinidade de topologias de redes distintas. As de interesse neste trabalho são as chamadas “redes de memória associativa”, (*associative memory networks* ou AMNs), segundo (Brown & Harris 1994). Estas redes são redes neurais artificiais *feedforward*<sup>5</sup>, de aprendizado supervisionado, e são aproximadores universais, i.e., podem aproximar qualquer função com uma precisão arbitrária.

A vantagem das AMNs sobre outros tipos de redes neurais são a relativa facilidade com que se obtém resultados analíticos rigorosos para diversas características destas redes, como convergência dos parâmetros, poder de representação, estabilidade (no caso de aprendizado *on line*). Algumas redes desse tipo também são duais da representação de um sistema de inferência fuzzy, propriedade que torna

---

<sup>4</sup>Este termo será melhor definido na seção 4.7

<sup>5</sup>Ou seja, o fluxo do sinal é apenas em uma direção, sem a presença de realimentação dentro da própria rede.

a fusão dos dois conceitos uma idéia muito interessante, já que permite utilizar os princípios da lógica fuzzy para sintetizar uma rede, que será posteriormente refinada utilizando algum algoritmo de treinamento. Esta característica será explorada durante a síntese do controlador, no capítulo 6.

## Capítulo 4

### Lógica fuzzy

O conceito básico por trás da lógica fuzzy é a representação da vagueza inerente a qualquer linguagem natural (i.e. linguagens usadas habitualmente por seres humanos normais, como o português ou o mandarim). Esta vagueza aparece sobretudo devido a imprecisão na definição e no uso de símbolos e devido a natureza tautológica e intuitiva da linguagem (Hofstadter 1979, Tsoukalas & Uhrig 1996).

O objetivo deste capítulo é fornecer informação suficiente para que seja possível compreender as idéias por trás de um controlador fuzzy. Inicialmente será dada uma idéia intuitiva do significado de conjuntos fuzzy. Depois serão expostos uma série de definições e conceitos relacionados à conjuntos fuzzy. Após isso, será apresentada a idéia de inferência fuzzy, o que levará à lógica fuzzy propriamente dita. Finalmente, algumas arquiteturas de controladores fuzzy serão tratadas.

#### 4.1 Introdução

A teoria clássica de conjuntos é baseada no conceito de que se pode sempre definir de forma clara se algum determinado objeto ou elemento pertence ou não a um determinado conjunto. Por exemplo, se determinado número pertence ao conjunto dos números primos ou não. Esta colocação é de uma inegável utilidade dentro do campo da matemática, porém, alguém que tente utilizar esta técnica para modelar alguns aspectos da realidade cotidiana normalmente esbarra na inexistência de definições precisas para certos conceitos.

Um conjunto clássico é definido sempre sobre um *universo de discurso* (normalmente representado por  $X$ ), ou seja, um determinado grupo de objetos ou elementos que podem ou não pertencer ao conjunto. Por exemplo, seja o universo de discurso o conjunto de pessoas em uma determinada região, e seja  $A$  o conjunto dos clientes de uma certa agência bancária. Pode-se representar o conjunto dos clientes do seguinte modo:

$$\begin{aligned} X &= \{x | \text{pessoas que moram em São Paulo}\} \\ x &\mapsto \chi_A(x) = \begin{cases} 1 & \text{se } x \text{ é cliente da agência} \\ 0 & \text{se } x \text{ não é cliente da agência} \end{cases} \end{aligned} \quad (4.1)$$

A função  $\chi(x)$  é chamada de função de pertinência do conjunto  $A$  sobre o universo de discurso  $X$ . Esta abordagem é útil para modelar uma série de aspectos da realidade, e, de uma forma muito mais sofisticada, é a base da teoria de bancos de dados relacionais (Silberschatz, korth & Sudarshan 1999). No entanto, resiste a representar alguns conceitos extremamente simples e cotidianos.

Imagine-se uma tentativa de representar o conceito de temperatura ambiente *confortável*. Não é possível traçar uma linha divisória rigorosa dizendo, por exemplo: se  $20 \leq T \leq 25$ , então a temperatura é confortável, caso contrário, a temperatura será desconfortável. A artificialidade deste conceito é patente, visto que, em um grupo de pessoas, muitas diriam que  $19^\circ\text{C}$  ainda é bastante razoável, outras diriam que  $22^\circ\text{C}$  ainda é frio<sup>1</sup>. Utilizando a notação de funções características, e chamando de  $T_c$  o conjunto das temperaturas confortáveis, obtém-se:

$$\begin{aligned} X &= \{x \in \mathbb{R} | -273 \leq x\} \\ \chi_{T_c} : X &\mapsto \{0; 1\} \\ x &\mapsto \chi_{T_c}(x) = \begin{cases} 1 & 20 \leq x \leq 25 \\ 0 & \text{qualquer outro valor} \end{cases} \end{aligned} \quad (4.2)$$

<sup>1</sup>Nota-se que aqui não se está fazendo uma medida estatística, mas sim tenta-se representar a idéia intuitiva, ou heurística, que um grupo de pessoas possui de um conceito

Uma outra representação mais realista deste conceito pode ser obtida deixando-se de lado por um momento as limitações impostas pela teoria clássica de conjuntos e considerando a *plausibilidade*, ou a *veracidade* da classificação, pode-se representar a “função de pertinência” deste modo:

$$\chi_{T_c}(x) = \begin{cases} 1 & 22 < x < 24 \\ \frac{x-18}{4} & 18 \leq x \leq 22 \\ \frac{24-x}{4} + 1 & 24 \leq x \leq 28 \\ 0 & \text{para qualquer outro valor} \end{cases} \quad (4.3)$$

As figuras 4.1 e 4.2 mostram uma representação gráfica das funções características para as duas equações acima.

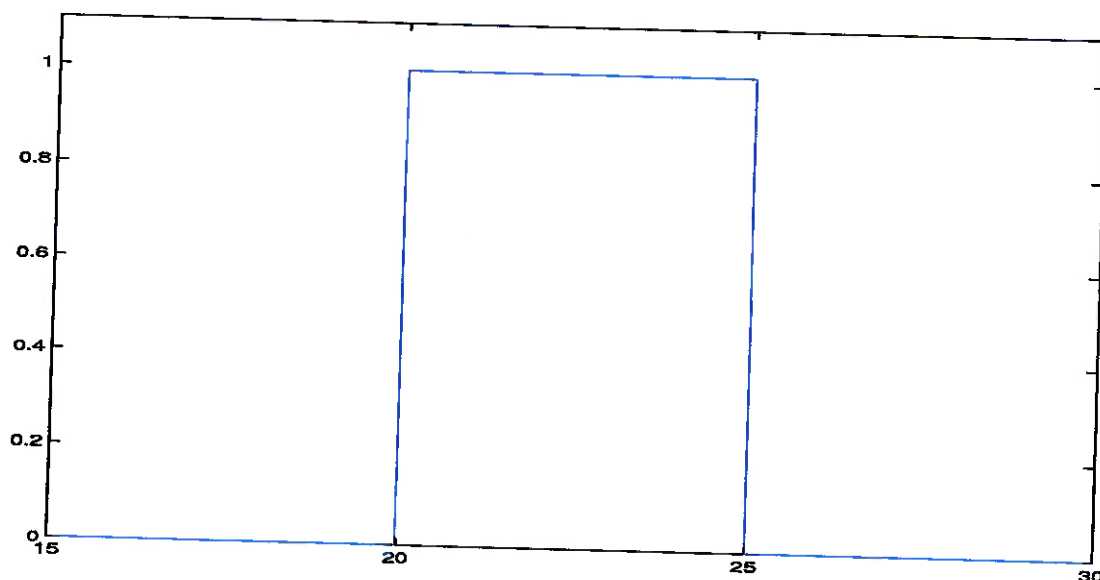


Figura 4.1: Representação clássica da função de pertinência do conjunto “temperatura confortável”

A representação e os conceitos mostrados na equação 4.3 e na figura 4.2 representam a intuição básica por traz da teoria de conjuntos fuzzy, e portanto por traz da lógica fuzzy. Nas próximas seções, será dado um desenvolvimento um pouco mais formal desta teoria.

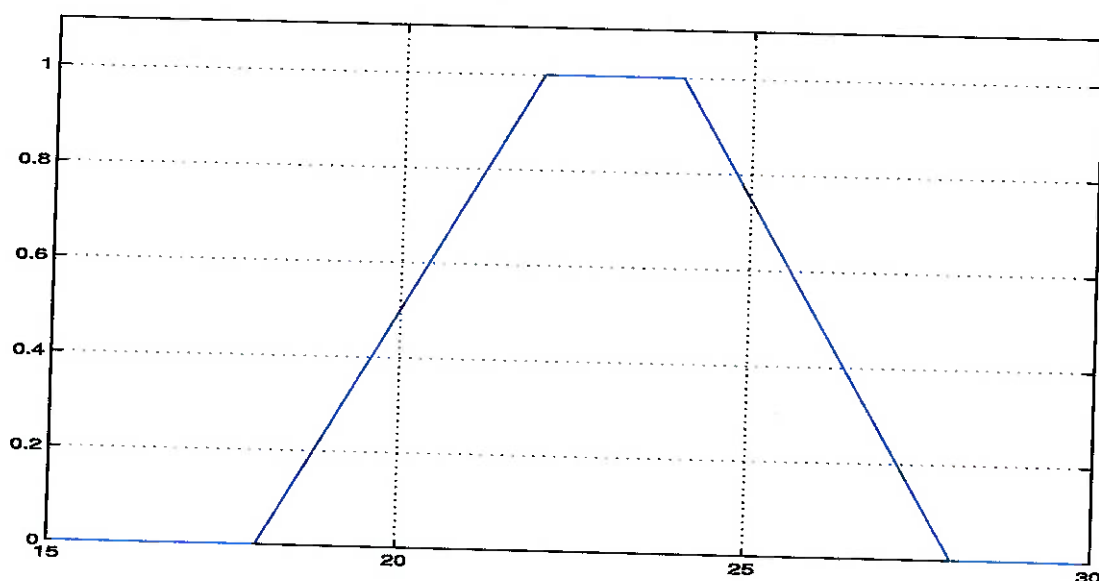


Figura 4.2: Representação fuzzy da função de pertinência do conjunto “temperatura confortável”

## 4.2 Conjuntos fuzzy

Como mostrado na seção anterior, um conjunto fuzzy é uma extensão do conceito clássico de conjunto, onde, ao invés da função de pertinência estar definida apenas sobre o conjunto  $\{0; 1\}$ , ela pode assumir uma gama muito maior de valores, sobre todo o intervalo  $[0; 1]$ . A função de pertinência de um conjunto fuzzy costuma ser representado pela letra  $\mu$ , para diferenciá-la da função para conjuntos clássicos. Assim, pode-se dizer que uma função de pertinência  $\mu_A$  mapeia todo elemento do universo de discurso  $X$  no intervalo  $[0; 1]$ , e é formalmente escrita como

$$\mu_A(x) : X \mapsto [0; 1] \quad (4.4)$$

O valor da função de pertinência para cada elemento de  $X$  pode ser definido livremente, sendo bastante dependente da aplicação. Normalmente estes valores são atribuídos com base na intuição, simplicidade de implementação, propriedades físicas do sistema, dados estatísticos<sup>2</sup>, ou mesmo uma identificação baseada em

<sup>2</sup>O que não significa que se está realizando uma análise estatística. Apesar de haverem vários pontos de contato entre as duas teorias (ver cap. 2 de (Tsoukalas & Uhrig 1996)), são dois enfoques bastante diferentes.

redes neurais ou algoritmos genéticos. Mas basta dizer que os valores das funções de pertinência, apesar de subjetivos, não devem ser arbitrários.

Um outro modo de representar um conjunto fuzzy é definir a coleção de pares ordenados

$$A = \{(x; \mu_A(x))\}, \quad x \in X \quad (4.5)$$

onde cada par  $(x; \mu_A(x))$  é chamado de *singleton* e é composto por  $x$ , seguido de seu valor de pertinência em  $A$ ,  $\mu_A(x)$ . Um *singleton* também pode ser representado como  $\mu_A(x)/x$ .<sup>3</sup> Define-se o *suporte* de um conjunto fuzzy  $A$  o conjunto dos *singletons* com um valor de pertinência diferente de zero.

Uma notação alternativa, também bastante utilizada, explicitamente representa um conjunto fuzzy como a *união* de todos os *singletons*  $\mu_A(x)/x$ , ou seja

$$A = \sum_{x_i \in X} \mu_A(x_i)/x_i \quad (4.6)$$

O sinal de somatório em 4.6 indica a união de todos os *singletons*. Esta equação pressupõe um universo de discurso discreto. Para um universo de discurso contínuo podemos reescrever 4.6 como

$$A = \int_X \mu_A(x_i)/x_i \quad (4.7)$$

Onde o símbolo de integral, ao contrário do que é normalmente utilizado em cálculo, representa a união de todos os *singletons*. Ela substitui o somatório apenas para indicar que  $X$  é contínuo, e não discreto.

### 4.3 Operações básicas

A maior parte das operações em conjuntos fuzzy, como união ou intersecção são definidas através de *t-normas* e *t-conormas* (ou *s-normas*), nomenclatura foi emprestada da teoria de probabilidades. Uma definição formal destas duas classes de funções pode ser encontrada em (Pedrycz 1993) e (Terano, Asai & Sugeno 1994). Grosso modo, as *t-normas* têm as mesmas propriedades da multiplicação e as *s-normas* as mesmas propriedades da adição. Nas definições abaixo, as *t-normas*

<sup>3</sup>Note que aqui "/" não significa divisão. É apenas um marcador.

serão representadas pelo símbolo  $\wedge$  e as s-normas serão representadas pelo símbolo  $\vee$ . Como exemplo, pode-se considerar que  $\wedge$  é a função min e  $\vee$  é a função max.

Abaixo estão uma série de definições úteis para operações com conjuntos fuzzy.

### 4.3.1 Conjunto vazio

Um conjunto fuzzy  $A$  é dito *vazio* (escrito como  $A = \emptyset$ ) quando sua função de pertinência vale zero em todo o universo de discurso, ou seja

$$A \equiv \emptyset \quad \text{se } \mu_A(x) = 0, \quad \forall x \in X \quad (4.8)$$

### 4.3.2 Conjunto fuzzy normal

Um conjunto fuzzy  $A$  é dito *normal* quando existe pelo menos um elemento  $x_0$  no universo de discurso para o qual a sua função de pertinência vale 1, ou seja

$$\exists x_0 \in X \mid \mu_A(x_0) = 1 \quad (4.9)$$

### 4.3.3 Igualdade de conjuntos fuzzy

Dois conjuntos fuzzy são ditos *iguais* se suas funções de pertinência são iguais para todos os elementos do universo de discurso, ou seja

$$A \equiv B \quad \text{se } \mu_A(x) = \mu_B(x) \quad \forall x \in X \quad (4.10)$$

### 4.3.4 União de conjuntos fuzzy

A união de dois conjuntos fuzzy  $A$  e  $B$  definidos sobre o mesmo universo de discurso  $X$  é um novo conjunto fuzzy  $A \cup B$  também em  $X$ , definido como

$$A \cup B \equiv \mu_A(x) \vee \mu_B(x) \quad (4.11)$$

### 4.3.5 Intersecção de conjuntos fuzzy

A intersecção de dois conjuntos fuzzy  $A$  e  $B$  definidos sobre o mesmo universo de discurso  $X$  é um novo conjunto fuzzy  $A \cap B$  em  $X$ , definido como

$$A \cap B \equiv \mu_A(x) \wedge \mu_B(x) \quad (4.12)$$



### 4.3.6 Complemento de um conjunto fuzzy

O complemento de um conjunto fuzzy  $A$  é um novo conjunto fuzzy  $\bar{A}$ , com a seguinte função de pertinência

$$\mu_{\bar{A}}(x) \equiv 1 - \mu_A(x) \quad (4.13)$$

Vale a pena lembrar que, diferentemente da teoria clássica de conjuntos, em geral

$$\bar{A} \cup A \neq X \quad (4.14)$$

### 4.3.7 Produto de dois conjuntos fuzzy

O *produto* de dois conjuntos fuzzy  $A$  e  $B$  definidos no mesmo universo de discurso  $X$  é um novo conjunto fuzzy  $A \cdot B$ , definido por

$$\mu_{A \cdot B} \equiv \mu_A(x) \cdot \mu_B(x) \quad (4.15)$$

Note que, na teoria clássica de conjuntos, esta definição equivale a intersecção entre dois conjuntos clássicos.

### 4.3.8 Potência de um conjunto fuzzy

É possível elevar um conjunto fuzzy  $A$  à potência  $\alpha$  (um número real positivo) elevando a sua função de pertinência à  $\alpha$ . Assim, a potência de  $\alpha$  de um conjunto fuzzy  $A$  é um conjunto fuzzy  $A^\alpha$  com a função de pertinência

$$\mu_{A^\alpha}(X) \equiv [\mu_A(x)]^\alpha \quad (4.16)$$

### 4.3.9 alfa-Cortes

A qualquer conjunto fuzzy é possível associar uma coleção de conjuntos clássicos conhecidos como  $\alpha$ -cortes de  $A$ . Um  $\alpha$ -corte é um conjunto consistindo de todos os elementos de  $X$  que pertencem ao conjunto  $A$  com um grau de pertinência de pelo menos  $\alpha$ . Os  $\alpha$ -cortes são um modo de resolver um conjunto fuzzy em uma série de conjuntos clássicos constituintes. Formalmente, o  $\alpha$ -corte de  $A$  é definido como

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\} \quad (4.17)$$

Onde  $\alpha$  é um parâmetro que varia de 0 a 1.

## 4.4 Relações fuzzy

Em relações fuzzy consideram-se pares de elementos (ou, de um modo mais geral,  $n$ -tuplas) que têm um *grau* de relacionamento entre si. Do mesmo modo que a questão da pertinência de um elemento a um conjunto pode ser considerada um problema de grau, também a associação de dois conjuntos pode ser considerada um problema de grau.

Formalmente, uma relação fuzzy é um conjunto fuzzy sobre um produto cartesiano. Ou seja, as relações fuzzy não são nada mais que conjuntos fuzzy definidos sobre universos de discurso multi-dimensionais. Supondo a relação fuzzy binária  $R$  definida sobre  $X \times Y$ , podemos representá-la utilizando a notação mostrada na equação 4.5, assim

$$R = \left\{ ((x; y); \mu_R(x, y)) \right\} \quad (x; y) \in X \times Y \quad (4.18)$$

## 4.5 Números fuzzy

Um caso especial interessante de conjuntos fuzzy são os chamados números fuzzy. Estes números são uma forma muito sintética e útil de representar a ambigüidade e incerteza encontradas em alguns resultados numéricos. Intuitivamente, eles são um método de representar asserções do tipo “aproximadamente três”, ou “quase cinco”. Além de todas as operações com conjuntos fuzzy que foram discutidas anteriormente, existe um conjunto de operações similares as da aritmética comum que pode ser definido para números fuzzy.

O universo de discurso sobre o qual estão definidos os números fuzzy é o conjunto dos números reais e seus subconjuntos, e suas funções de pertinência devem ser *normais* e *convexas*. A definição de conjunto normal foi dada em 4.3.2. Intuitivamente, um conjunto é convexo quando sua função de pertinência não tem nenhum “buraco”, como representa a figura 4.3. Vale a pena notar que, como é típico em lógica fuzzy, estas são as duas únicas limitações quanto a forma da

função de pertinência, sendo que qualquer definição mais específica dependerá da aplicação.

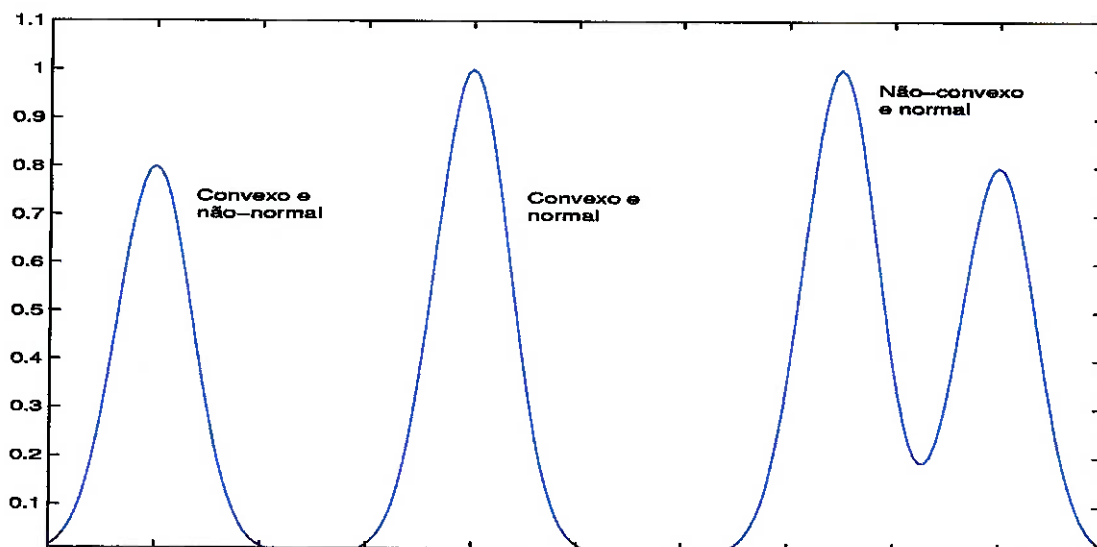


Figura 4.3: Comparação entre uma função de pertinência normal, uma não-normal e uma não-convexa

De um modo formal, um conjunto fuzzy convexo  $A$  é definido tal que: (Pedrycz 1993)

$$\forall \lambda \in [0;1] \forall x \in \mathbb{R} A[\lambda x + (1 - \lambda)y] \geq \min(A(x), A(y)) \quad (4.19)$$

um dos modos mais úteis de se definir números fuzzy é através da união de seus  $\alpha$ -cortes

$$A = \bigvee_{0 \leq \alpha \leq 1} \alpha A_\alpha = \bigvee_{0 \leq \alpha \leq 1} \alpha [a_1^{(\alpha)}; a_2^{(\alpha)}] \quad (4.20)$$

Para simplificar a notação, pode-se indicar a equação 4.20 resumidamente por

$$A = [a_1^{(\alpha)}; a_2^{(\alpha)}] \quad (4.21)$$

sendo feita a ressalva de que esta equação indica a união de todos os  $\alpha$ -cortes de 0 a 1. Finalmente, é importante notar que uma boa parte das regras válidas para números comuns, como  $(7 \div 3) \times 3 = 7$ , não vale para números fuzzy.

### 4.5.1 Adição

Define-se a adição como

$$A + B \equiv [a_1^{(\alpha)}; a_2^{(\alpha)}] + [b_1^{(\alpha)}; b_2^{(\alpha)}] \quad (4.22)$$

Os intervalos são calculados da seguinte maneira

$$A + B = [a_1^{(\alpha)} + b_1^{(\alpha)}; a_2^{(\alpha)} + b_2^{(\alpha)}] \quad (4.23)$$

### 4.5.2 Subtração

De modo análogo a adição, define-se a subtração como

$$A - B \equiv [a_1^{(\alpha)}; a_2^{(\alpha)}] - [b_1^{(\alpha)}; b_2^{(\alpha)}] \quad (4.24)$$

### 4.5.3 Multiplicação

Define-se a multiplicação de dois números fuzzy como

$$A \cdot B \equiv [a_1^{(\alpha)}; a_2^{(\alpha)}] \cdot [b_1^{(\alpha)}; b_2^{(\alpha)}] \quad (4.25)$$

Os intervalos são calculados da seguinte maneira

$$A \cdot B = [a_1^{(\alpha)} \cdot b_1^{(\alpha)}; a_2^{(\alpha)} \cdot b_2^{(\alpha)}] \quad (4.26)$$

### 4.5.4 Divisão

Define-se o quociente de dois números fuzzy como

$$A \div B \equiv [a_1^{(\alpha)}; a_2^{(\alpha)}] \div [b_1^{(\alpha)}; b_2^{(\alpha)}] \quad (4.27)$$

Os intervalos são calculados da seguinte maneira

$$A \div B = \left[ \frac{a_1^{(\alpha)}}{b_2^{(\alpha)}}, \frac{a_2^{(\alpha)}}{b_1^{(\alpha)}} \right] \quad (4.28)$$

## 4.6 Inferência fuzzy

Pode-se dizer que esta é a operação principal da lógica fuzzy. Na verdade, é apenas a partir da definição de regras de inferência que se pode falar em *lógica fuzzy*. As primeiras aplicações de controladores fuzzy (Mandani 1994), e uma grande parte dos controles atuais (Terano et al. 1994), (Reyero & Nicolás 1995)) são totalmente baseados em uma série de regras, ou *descrições lingüísticas*, escritas em linguagem natural, com construções do tipo “Se/Para”, e se utilizam regras de inferência para obter uma ação de controle a partir desta base de conhecimento.

Em se tratando da lógica clássica, existem duas operações básicas de inferência, conhecidas como *modus ponens* e *modus tollens* (respectivamente, modo que afirma e modo que nega). O *modus ponens* é representado da seguinte maneira

$$\frac{P \rightarrow Q \quad P}{\therefore Q} \quad (4.29)$$

Ele é composto de duas *premissas* e um *conseqüente*. A primeira premissa é a regra “Se/Para”. A segunda premissa é a veracidade de P. Partindo destas duas premissas é possível avaliar a veracidade de Q.

Do mesmo modo, o *modus tollens* costuma ser representado por:

$$\frac{P \rightarrow Q \quad \neg Q}{\therefore \neg P} \quad (4.30)$$

Aqui, do mesmo modo, temos duas premissas, ou *antecedentes*, e um *conseqüente*. Aqui raciocina-se que se Q é falsa, então P é falsa.

Em lógica fuzzy, estas duas relações são generalizadas em *modus ponens generalizado* (MPG) e em *modus tollens generalizado* (MTG). No MPG, a primeira asserção é definida através de uma relação fuzzy, ou seja, a parte “Se/Para” da inferência é representada por uma relação na forma da equação 4.18. Então, dado um certo valor fuzzy que se supõe semelhante a P, obtém-se um *conseqüente* com

um determinado grau de relação com  $Q$ . Graficamente:

$$\frac{\begin{array}{l} \text{Se } x \text{ é } P \text{ então } y \text{ é } Q \\ x \text{ é } P' \end{array}}{\therefore y \text{ é } Q'} \quad (4.31)$$

Analiticamente, esta expressão é avaliada como

$$Q' = P' \circ R(x, y) \quad (4.32)$$

Analogamente, para o MTG temos:

$$\frac{\begin{array}{l} \text{Se } x \text{ é } P \text{ então } y \text{ é } Q \\ y \text{ é } Q' \end{array}}{\therefore X \text{ é } P'} \quad (4.33)$$

Analiticamente, esta expressão é avaliada como

$$P' = R(x, y) \circ Q' \quad (4.34)$$

Este tipo de cálculo (a utilização de composição de relações para inferir conseqüências) é conhecido como regra composicional de inferência. Agora que já foi definido o modo de operação para uma regra, podemos definir *algoritmos fuzzy*, que são uma coleção de regras do tipo mostrado em 4.31 conectadas entre si através de um “Senão”. Analiticamente, este “senão” é interpretado ou como uma t-norma ou como uma s-norma, dependendo do tipo de relação que será usado. Assim, um algoritmo fuzzy é uma expressão analítica resultante da conexão de uma série de relações fuzzy que, ao receber como antecedente um determinado conjunto fuzzy, produz como conseqüente um conjunto fuzzy, resultado da aplicação de todas as regras que formam o algoritmo.

## 4.7 Quantidade de imprecisão

Em (Pedrycz 1995) se define o conceito da medida de imprecisão (*Fuzzyness*) de um conjunto fuzzy. Esta medida visa dar uma idéia de quão amplo, ou pouco

definido, é determinado conceito representado através de um conjunto fuzzy. Existem dois tipos de medidas: uma medida de entropia e uma medida de energia. A definição da medida de entropia  $H(A)$  de um conjunto fuzzy  $A$  é

$$H(A) = \int_X f(A(x)) dv \quad (4.35)$$

onde  $v$  é qualquer medida definida em  $X$  e  $f$  é qualquer função  $f : [0; 1] \rightarrow [0; 1]$ , de modo que seja monotonicamente crescente de 0 até 0,5 e monotonicamente decrescente de 0,5 até 1.

Evidentemente, esta definição é extremamente geral. O que se costuma utilizar é  $f = \log_2$ , o que faz com que  $H$  seja equivalente a definição de entropia na termodinâmica clássica.

Uma outra medida possível é a energia de um conjunto fuzzy, que é definida como

$$E(A) = \int_X h(A(x)) dv \quad (4.36)$$

onde  $v$  é qualquer medida definida em  $X$  e  $h$  é qualquer função  $f : [0; 1] \rightarrow [0; 1]$ , de modo que seja monotonicamente crescente de 0 até 1.

Uma escolha comum para  $h$  é  $h(v) = v$ , e esta medida pode ser interpretada como a cardinalidade de  $A$ .

## 4.8 Controladores fuzzy

A estrutura básica de um controlador fuzzy está representada na figura 4.4. Percebe-se a presença de três unidades distintas: a etapa de fuzificação<sup>4</sup>, o motor de inferência e a etapa de defuzificação. O núcleo dos controladores fuzzy é o motor de inferências. Aí estão todas as regras de inferência e é neste local que os antecedentes são propriamente avaliados e os conseqüentes produzidos.

O bloco de fuzificação é necessário porque os sensores não costumam apresentar dados na forma de conjuntos fuzzy, mas sim como números comuns. O algoritmo de defuzificação recebe esses dados e os mapeia em conjuntos fuzzy, que são então passados ao motor de inferências.

---

<sup>4</sup>Na falta de um termo adequado em português, se usará este aportuguesamento

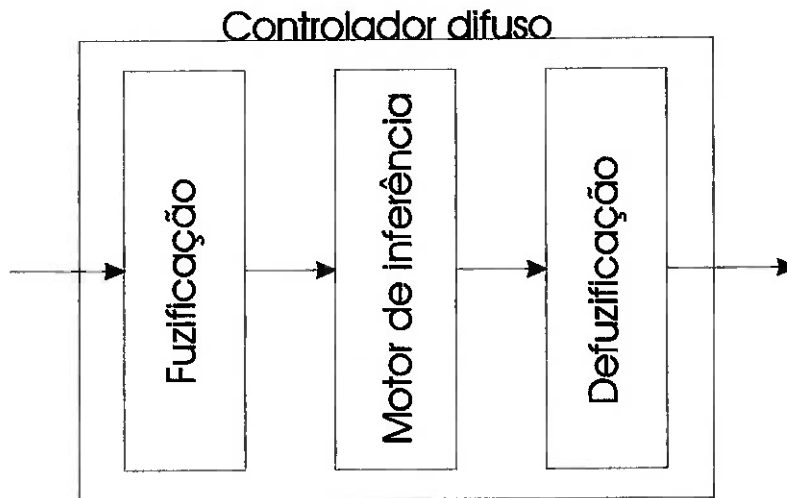


Figura 4.4: Estrutura básica de um controlador fuzzy

Do mesmo modo, salvo raras exceções, os atuadores também só aceitam sinais de controle na forma de números comuns. Então o algoritmo defuzificador recebe um conjunto fuzzy como conseqüente do motor de inferências e transforma, de acordo com alguma regra, este conjunto em um número normal. A exceção à necessidade de um fuzificador e defuzificador se dá quando a comunicação é feita diretamente com seres humanos. Para estes casos, é possível obter entradas já fuzificadas, e as saídas, apesar de em alguns casos ainda necessitarem de um tratamento, não precisam mais ser convertidas em números comuns.

Existem diversas técnicas tanto de fuzificação quanto de defuzificação, a maior parte delas não relevantes ao problema que se pretende resolver, portanto não serão abordadas aqui. Com relação à arquitetura do controlador proposto, ela está descrita no capítulo 6.

O tipo de controlador que será utilizado neste trabalho é uma variação de um controlador conhecido com *Takagi e Sugeno*. De acordo com (Wang 1994), estes controladores são da forma:

$$R_l : \text{ Se } x_1 \text{ é } X_{S1} \text{ e } x_2 \text{ é } X_{S2} \dots \text{ e } x_n \text{ é } X_{SN} \quad (4.37)$$

$$\text{então } y_l = c_0^l + c_1^l x_1 + \dots + c_N^l x_N$$

onde  $X_{S_i}$  são conjuntos fuzzy,  $c_i^l$  são valores reais,  $y_l$  é a saída da regra  $l$ , e



$x_n$  são os valores de entrada. Ou seja, as regras tem um antecedente difuso e um consequente real, que é uma combinação linear das entradas. A saída  $y$  de um sistema de *Takagi e Sugeno* é uma média ponderada das  $y_l$ :

$$y(x) = \frac{\sum_{l=1}^M w_l y_l}{\sum_{l=1}^M w_l} \quad (4.38)$$

Onde o peso  $w_l$  é o valor de verdade de todo o antecedente, ou seja:

$$w_l = \prod_{i=1}^N \mu_i^l(x_i) \quad (4.39)$$

# Capítulo 5

## Redes Neurais

O objetivo deste capítulo é dar uma breve introdução ao tipo de rede neural que será utilizada neste trabalho, e mostrar o método de treinamento, junto com alguns teoremas sobre convergência e estabilidade do aprendizado.

### 5.1 Memória associativas

Uma rede do tipo memória associativa baseada em retículo (AMN<sup>1</sup>) pode ser mapeada na estrutura da figura 5.1. A primeira camada desta rede é apenas uma camada de *buffer*, mas que pode ser alterada para realizar a normalização das variáveis de entrada. A camada intermediária é composta de uma série de funções não lineares, chamadas funções de ativação, de escopo mais ou menos local, ou seja, que só tem valor maior que zero para uma região limitada do espaço formado pelas variáveis de entrada. A camada de saída é um combinador linear adaptativo (ALC<sup>2</sup>), que é onde se encontra a capacidade de aprendizado desta rede. Mostra-se que, devido a ativação apenas local da camada intermediária, esta rede armazena informações localmente, isto é, pontos próximos no espaço de entrada ativam funções que se superpõe na camada intermediária.

A capacidade de modelagem desta rede depende do tamanho, forma e distribuição das funções de ativação. Esta dependência com relação ao tamanho ficará muito evidente para o projeto atual. De um modo geral, os requisitos de

---

<sup>1</sup>*Lattice based Associative Memory Network*

<sup>2</sup>*Adaptive Linear Combiner*

memória para as AMNs dependem exponencialmente do tamanho do espaço de entrada. No entanto, caso as funções de ativação possuam um suporte compacto<sup>3</sup>, apenas um pequeno número delas estará ativa a cada momento. Neste caso, apesar da necessidade de memória continuar a mesma, o tempo de execução (o tempo para calcular a resposta da rede) não aumenta na mesma velocidade (em alguns tipos de AMNs, o número de funções ativas em cada momento é um parâmetro de projeto).

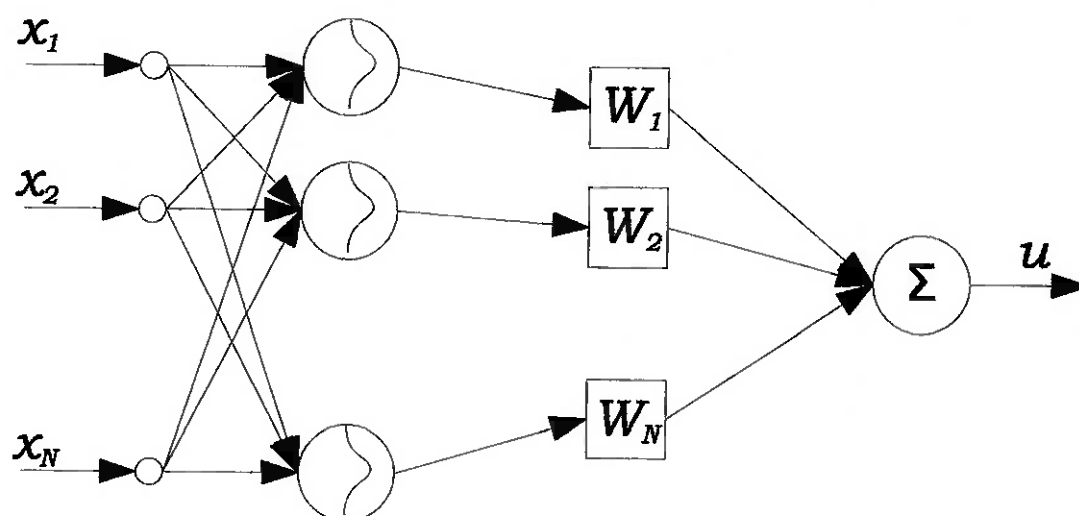


Figura 5.1: Modelo geral de uma rede neural associativa de múltiplas camadas

Uma outra vantagem das AMNs (e existe uma ligação direta com a lógica fuzzy neste ponto) é que, como pela estrutura da rede se **obriga** a generalização local, o conhecimento é armazenado de modo transparente (ou menos obscuro) neste tipo de rede. Caso a rede tenha funções de ativação com um suporte não compacto, a alteração de um peso altera o valor da saída da rede para todo o espaço de entrada. Isto permite uma maior liberdade de aprendizado, mas também exige algoritmos mais complexos, e faz com que o conhecimento seja armazenado na rede de um modo incompreensível para um ser humano. De modo análogo, a alteração de um

<sup>3</sup>De acordo com Brown & Harris (1994), apenas nesse caso pode-se dizer que as redes são baseadas em retículo, mas que, caso as funções tenham valores significativamente maiores que zero para apenas uma pequena região do espaço de entrada (que é o caso da função de ativação gaussiana, utilizada neste projeto), valem as mesmas propriedades.

peso em uma AMNs altera apenas as saídas relacionadas com pontos próximos daquele ponto no espaço de estados. A extensão da alteração na saída depende do grau de superposição das funções de ativação.

## 5.2 Retropropagação

Como foi dito na seção anterior, a saída das AMNs é um ALC. Este elemento é funciona como um modelo linear adaptativo aplicado sobre o espaço de entrada distorcido pelas funções de ativação. Sendo  $\mathbf{a}$  um vetor com as saídas de todas as funções de ativação, e  $\mathbf{w}$  o vetor com os pesos associados a cada uma das funções de ativação, a saída da AMN no tempo  $t$  é da forma:

$$u = \mathbf{a}^T \mathbf{w} \quad (5.1)$$

Para adaptar o vetor de pesos, é necessária uma medida de performance do modelo. Para o caso de aprendizado *offline*, está sempre disponível o valor desejado ( $\hat{u}$ ) para a saída da rede. Deste modo, uma medida da performance do modelo é:

$$J(\epsilon) = \frac{1}{2} \epsilon^2 \quad (5.2)$$

onde  $\epsilon$  é o erro do modelo, e é definido como

$$\epsilon = \hat{u} - u = \hat{u} - \mathbf{a}^T \mathbf{w} \quad (5.3)$$

O vetor de pesos deve ser então ajustado de modo a melhorar a performance do modelo. O algoritmo de retropropagação simples consiste em atualizar os pesos baseado no valor da variação do erro com a variação do peso, ou seja:

$$\Delta \mathbf{w} = -\alpha \frac{\partial \epsilon}{\partial \mathbf{w}} \quad (5.4)$$

onde  $\alpha$  é uma constante chamada constante de aprendizado, que está relacionada com a velocidade de aprendizado. Substituindo 5.2 em 5.4, obtemos:

$$\Delta \mathbf{w} = \alpha \epsilon \mathbf{a} \quad (5.5)$$

Sobre essa equação, é interessante notar que a direção de busca do peso correto é sempre paralela a  $\mathbf{a}$ , e que, reforçando o que foi dito anteriormente com relação ao aprendizado local, os pesos só serão atualizados para os elementos não 0 de  $\mathbf{a}$ .

Após a correção do erro, a saída *a posteriori*  $\underline{u}$  da rede é:

$$\begin{aligned}\underline{u} &= \mathbf{a}^T \underline{\mathbf{w}} \\ &= \mathbf{a}^T \mathbf{w} + \mathbf{a}^T \alpha \epsilon \mathbf{a} \\ &= \alpha \|\mathbf{a}\|_2^2 \hat{u} + (1 - \alpha \|\mathbf{a}\|_2^2) u\end{aligned}\quad (5.6)$$

onde  $\|\mathbf{a}\|_2^2 = \mathbf{a}^T \mathbf{a}$  e o erro de saída *a posteriori* é dado por:

$$\begin{aligned}\epsilon &= \hat{u} - \underline{u} \\ &= (1 - \alpha \|\mathbf{a}\|_2^2) \epsilon\end{aligned}\quad (5.7)$$

Para um valor não zero de  $\epsilon$ , as seguintes relações entre o valor do erro de saída *a priori* e *a posteriori* podem ser estabelecidas para diferentes valores de  $\alpha$ :

$ \epsilon  >  \epsilon $	Se $\alpha \notin \left[0; \frac{2}{\ \mathbf{a}\ _2^2}\right]$
$ \epsilon  =  \epsilon $	Se $\alpha = 0$ ou $\alpha = \frac{2}{\ \mathbf{a}\ _2^2}$
$ \epsilon  <  \epsilon $	Se $\alpha \in \left(0; \frac{2}{\ \mathbf{a}\ _2^2}\right)$
$ \epsilon  = 0$	Se $\alpha = \frac{1}{\ \mathbf{a}\ _2^2}$

Tabela 5.1: Influência do valor de  $\alpha$  no aprendizado.

Quando  $\alpha = 1/\|\mathbf{a}\|_2^2$ , o erro *a posteriori* é sempre 0. No entanto, não é um valor interessante para ser utilizado a não ser que, para todos os  $\hat{u}$ ,  $\mathbf{a}$  sejam ortogonais entre si. Caso esse não seja o caso, o aprendizado de um valor implicará na “esquecimento” de outro.

Caso a variação na magnitude de  $\mathbf{a}$  variar muito durante o treinamento,  $\alpha$  deve ser escolhido cuidadosamente, de modo que o treinamento é estável mesmo quando  $\|\mathbf{a}\|_2^2$  é grande. Isto pode acarretar uma adaptação lenta em certas áreas do espaço de entrada. De um modo geral, um grande valor de  $\alpha$  no início proporciona um aprendizado inicial rápido, enquanto um valor pequeno produz maior rejeição de ruídos.

## Capítulo 6

# Arquitetura do controlador

O objetivo deste projeto é, utilizando técnicas de chaveamento, lógica fuzzy e redes neurais artificiais, e construindo sobre uma série de controladores LQ e modelos lineares obtidos em outros projetos, construir um controlador capaz de manter o valor da relação estequiométrica normalizada dos gases de escape de um modelo de motor a gás natural ao redor de um, tentando obter um erro máximo de  $\pm 1\%$ . O desempenho deste novo controlador será comparado com o de um controlador (já implementado) baseado em múltiplos modelos e chaveamento, que será descrito com mais detalhes na próxima seção. , e que foi obtido de (Freitas Junior 2003).

A “filosofia” por trás do projeto consiste em utilizar princípios de lógica fuzzy para “interpolare” os vários controladores existentes. Então, utilizando-se de analogias entre a estrutura de interpoladores fuzzy e alguns tipos de redes neurais, utilizar um algoritmo de treinamento para ajustar a interpolação de modo a mostrar resultados próximos dos exatos para pontos conhecidos. Esta arquitetura está detalhada na seção 6.2.

### 6.1 Controlador LQ existente

Como já foi dito no capítulo 2, pode-se considerar a variação no ângulo de abertura da borboleta como uma perturbação sobre o motor. No artigo (Narendra et al. 1995), propõe-se que uma estratégia de chaveamento entre vários controladores lineares e/ou adaptativos pode ser utilizada para reduzir os transientes devido à variações rápidas das condições ambientais. Baseado nessa idéia, imaginou-se que

o uso desta técnica traria bons resultados, pois pode-se considerar o ângulo de abertura da borboleta como uma variação ambiental.

A partir daí, dividiu-se o curso de interesse da válvula borboleta (de 35 a 90 graus) em 25 pontos mais ou menos igualmente espaçados (i.e. 35, 38, 40, 43, 45 etc), e para cada uma dessas condições encontrou-se um ponto de equilíbrio. Então, foi feita a linearização em torno de cada um desses pontos de equilíbrio, e para cada um desses modelos lineares foi sintetizado um controlador do tipo LQ<sup>1</sup>. Utilizando-se estes controladores, realizou-se um sistema simples de chaveamento. Neste sistema, cada controlador é identificado pelo ângulo de abertura da borboleta que foi utilizado na sua síntese<sup>2</sup>, e seleciona-se qual o controlador utilizado baseando-se apenas no ângulo de abertura da borboleta. Ou seja:

Seja  $S = \{S_\alpha\}$  o conjunto dos controladores, onde  $\alpha = 30, 33, 35, 38, 40 \dots 90$

Seja  $\bar{\alpha}$  o ângulo de abertura da borboleta medido.

Seja  $S_{\alpha_C}$  o controlador utilizado atualmente na planta.

$\alpha_C$  é o valor de  $\alpha$  que minimiza a relação  $|\alpha - \bar{\alpha}|$

Deste modo, pode-se dizer que o controlador escolhido é aquele que melhor aproxima a condição de operação naquele momento. Este primeiro controlador é o proposto no capítulo 5 de (Freitas Junior 2003). No entanto um dos problemas que se percebeu rapidamente é que as linearizações (e portanto os controladores LQ baseados nelas) são válidas apenas para uma região muito próxima ao ponto onde o sistema foi linearizado. Deste modo, o controlador apresentou um desempenho bastante aceitável para posições próximas de um dos pontos de linearização (e.g.: o controlador de 35 graus funciona muito bem com a borboleta a 35,5 graus), mas quando enfrentava uma excursão maior (da ordem de 1 grau, por exemplo), o desempenho já caía bastante.

<sup>1</sup>Estes controladores, além de produzirem os valores de  $\beta$  e  $vc_b$ , também geram o valor de  $\alpha$ . Este valor foi usado durante as simulações do controlador fuzzy-neural para se obter uma avaliação aproximada da validade da interpolação.

<sup>2</sup>Estes apelidos serão sempre utilizados de aqui para adiante. Assim, quando se disser “controlador de 50°”, na verdade se está falando do controlador LQ projetado sobre a linearização do modelo do motor ao redor de um ponto de equilíbrio quando o ângulo de abertura da borboleta é de 50 graus.

A figura 6.1 exemplifica a situação. Suponha que os círculos completos representem a região de validade da linearização, e suponha que a linha tracejada indique a fronteira de quando um controlador é chaveado para o seguinte. Percebe-se que existem várias regiões não modeladas no espaço, para as quais nenhum dos controladores existentes é válido.

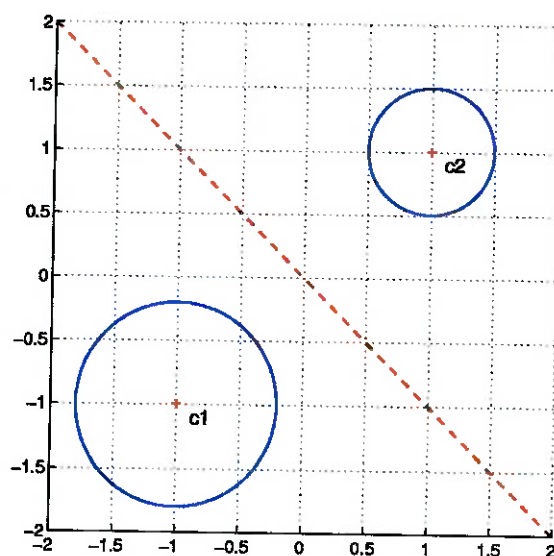


Figura 6.1: Representação das regiões de validade dos controladores e da linha de chaveamento

Este controlador servirá de comparação para o desempenho do novo controlador a ser projetado. Este controlador é baseado em uma estratégia com múltiplos modelos e chaveamento “brusco”, ou seja, ocorrem mudanças bruscas na lei de controle.

A estrutura de controle é semelhante à mostrada na figura 3.2, com exceção que o chaveamento não se dá baseado no erro de predição, mas sim na posição da borboleta. As figuras 6.3 e 6.4 mostram a resposta das saídas para uma excursão de 50 a 80 graus de abertura da borboleta (figura 6.2).

A variável que mais interessa é  $\phi_e$ , pois é um indicativo da faixa de eficiência em que irá trabalhar o catalizador. Levando isto em conta, serão utilizados dois parâmetros para a comparação:  $\phi_{eMAX}$ , a máxima excursão fora da faixa de  $\pm 1\%$ , e  $\int_C \phi_e$ , a integral corrigida do erro, ou seja, a integral da diferença entre  $\phi_e$  e



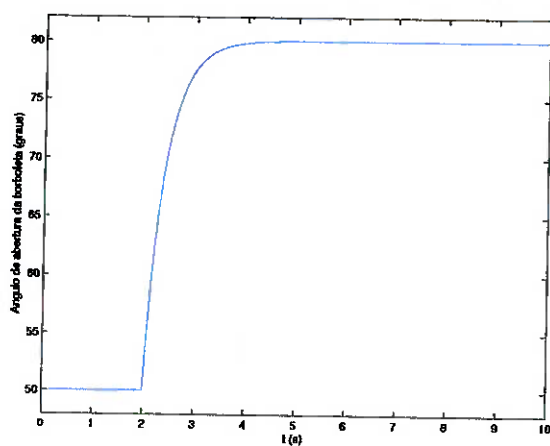


Figura 6.2: Ângulo de abertura da borboleta

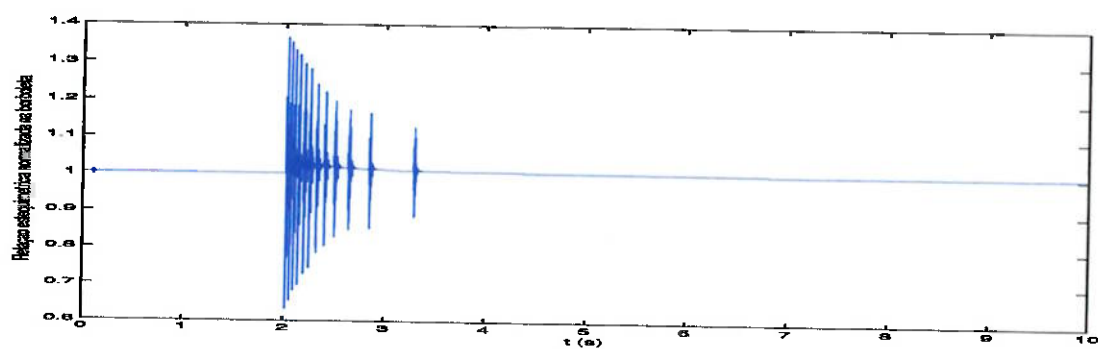


Figura 6.3: Variação da relação estequiométrica normalizada, medida na borboleta

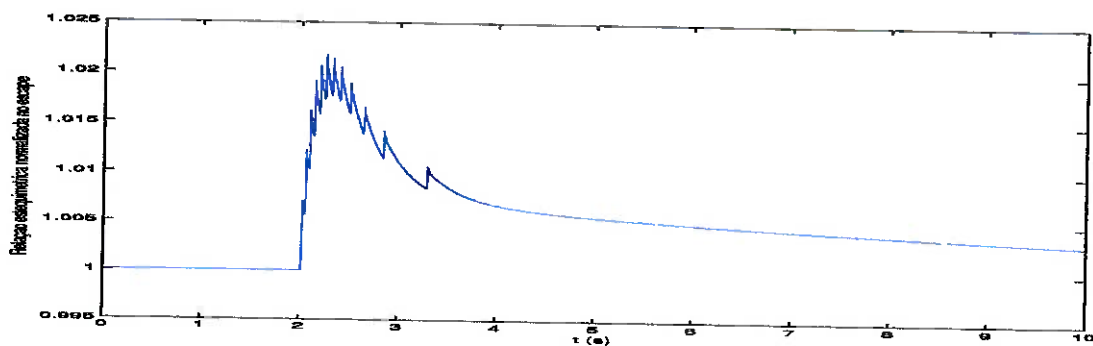


Figura 6.4: Variação da relação estequiométrica normalizada, medida no escape

$1 \pm 0,005$ . Esta zona morta de  $\pm 0,3\%$  aparece devido a curva da figura 1.1, que mostra que com o motor trabalhando em uma região próxima de 1, produz menos poluentes. Assim, apesar da produção de  $NO_x$  ser grande, existe a vantagem de um bom funcionamento do catalizador nesta região. A tabela 6.1 mostra os valores destes parâmetros para a simulação da figura 6.4.

Característica	Valor
Integral do erro	0,0220
Máxima excursão	2,17 %

Tabela 6.1: Valores dos parâmetros de desempenho para o controlador LQ com chaveamento brusco

## 6.2 Síntese do controlador

Na seção anterior foram mostrados dois problemas com relação ao controle LQ com chaveamento brusco. O primeiro é que sempre que ocorre o chaveamento, ocorre uma mudança brusca na entrada de controle (isto explica porque o gráfico da figura 6.3 parece tão “espinhoso”), e isto gera transitórios no motor que tornam mais lenta a convergência deste para o valor de referência. O segundo problema, e bem mais sério, é que uma boa parte do espaço não possui nenhum controlador válido associado. Ou seja, durante a maior parte do regime de trabalho do motor, este está recebendo entradas de controle que estão pouco relacionadas com a dinâmica local do motor.

Uma das soluções possíveis para esse problema é tentar utilizar os controladores existentes para “interpolarm” novos controladores. Utilizando-se o arcabouço de lógica fuzzy mostrado no capítulo 4, podemos imaginar uma base de regras de controle do tipo:

$$R_i: \quad \text{Se } \bar{x}_S \text{ é } x_{Si} \text{ então } \bar{u} \text{ é } u_i \quad (6.1)$$

Onde  $\bar{x}_S$  é o vetor de variáveis de estado estendido<sup>3</sup> medido do sistema e  $\bar{u}$  é a saída de controle para aquela regra.  $x_{Si}$  e  $u_i$  são respectivamente os valores de

<sup>3</sup>Ou seja, todas as variáveis de estado, mais a medida do ângulo de abertura da borboleta

“referência” e as saídas de controle para a regra  $i$ .

Esta idéia se baseia no princípio de que quanto mais próximo no espaço de estados se está de um controlador, maior deverá ser o peso deste controlador na interpolação.

Cada controlador têm a seguinte forma:

$$\mathbf{u}_i = \mathbf{u}_i(\bar{\mathbf{x}}) = [\mathbf{K}_i] \bar{\mathbf{x}} + \mathbf{u}_{0i} \quad (6.2)$$

Onde  $[\mathbf{K}_i]$  é a matriz de realimentação de estados para o controlador  $i$  e  $\mathbf{u}_{0i}$  é o vetor de controle para regime permanente<sup>4</sup>. Como o sistema controlado tem duas entradas ( $\beta$  e  $vc_b$ ), pode-se considerar o controlador formado por um conjunto de regras desse tipo na verdade como dois controladores fuzzy do tipo *Takagi e Sugeno*: um cujo conseqüente é  $\beta$  e outro cujo conseqüente é  $vc_b$ . No entanto, como o antecedente para ambos é exatamente igual, se trabalhará sempre com a forma matricial acima, por ser mais conveniente.

Uma vez definida a forma geral de cada uma das regras da base de regras, falta definir a forma dos conjuntos de pertinência e qual das regras de inferência será usada. Como se deseja uma interpolação apenas entre controladores próximos, e no entanto se deseja um operador não-linear e contínuo (para evitar ruídos devidos a descontinuidades), serão utilizadas funções gaussianas para definir os conjuntos de pertinência. Pelas mesmas razões de suavidade, se usará como regra de inferência o produto algébrico.

Dessa forma, temos definido um interpolador não-linear entre os controladores que pode ser definido da seguinte forma:

Seja

$$\mu_i(\mathbf{x}) = \prod_{j=1}^K e^{-\left(\frac{x_j - x_{ij}}{\sigma_{ij}}\right)^2} \quad (6.3)$$

a função de pertinência da regra  $i$ , onde  $j$  representa cada um dos componentes do vetor de estados estendido, e  $\sigma_{ij}$  define o raio da função gaussiana para aquela variável. A figura 6.5 mostra uma forma típica de  $\mu_i$  para um caso com duas

<sup>4</sup>Ou seja, a entrada de controle necessária para manter o sistema naquele ponto de equilíbrio

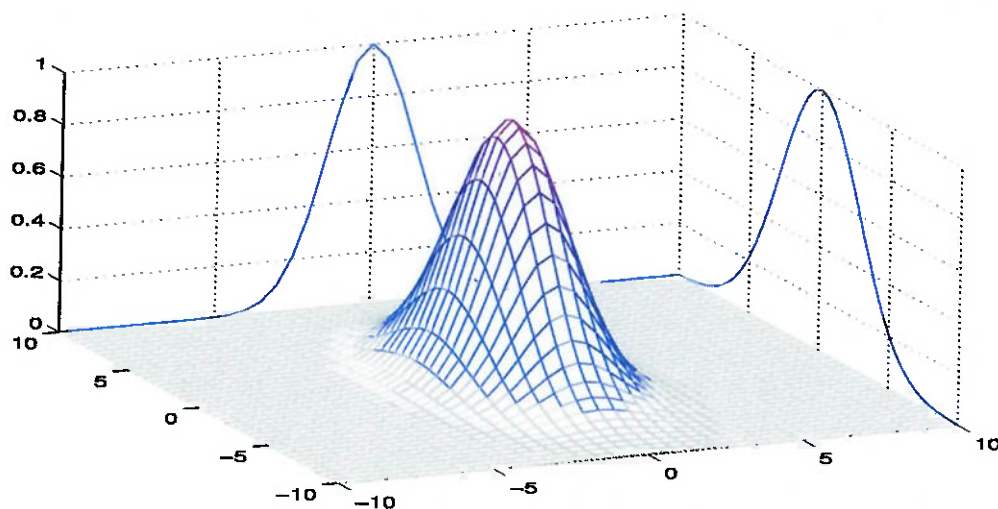


Figura 6.5: Representação de uma função de pertinência gaussiana em um espaço bidimensional

dimensões. A partir desta definição, o controlador fuzzy fica definido pela seguinte equação:

$$\bar{u} = \frac{\sum_{i=1}^N \mu_i(\bar{x}_s) \left( [K_i] \bar{x} + u_{oi} \right)}{\sum_{i=1}^N \mu_i(\bar{x}_s)} \quad (6.4)$$

O próximo passo é estabelecer um método que permita realizar algum tipo de treinamento deste controlador. Exatamente como está, a equação 6.4 pode ser representada por um modelo de rede do tipo *feedforward*, como mostrado na figura 6.6. Normalmente sobre esses tipos de rede podem ser aplicados algoritmos de aprendizado do tipo retropropagação. Em (Wang 1994) está descrito um algoritmo deste tipo para realizar o treinamento inclusive dos valores de  $\sigma_{ij}$  e de  $x_{si}$ . Com relação aos valores de  $x_{si}$ , estes já são conhecidos com bastante precisão. Experimentou-se o algoritmo para tentar ajustar os valores dos  $\sigma_{ij}$ , mas ocorreram problemas tanto de instabilidade quanto de convergência demasiadamente lenta.

De acordo com Brown & Harris (1994), e como já foi discutido no capítulo 5, realizar a adaptação para uma rede cuja saída depende linearmente dos parâmetros

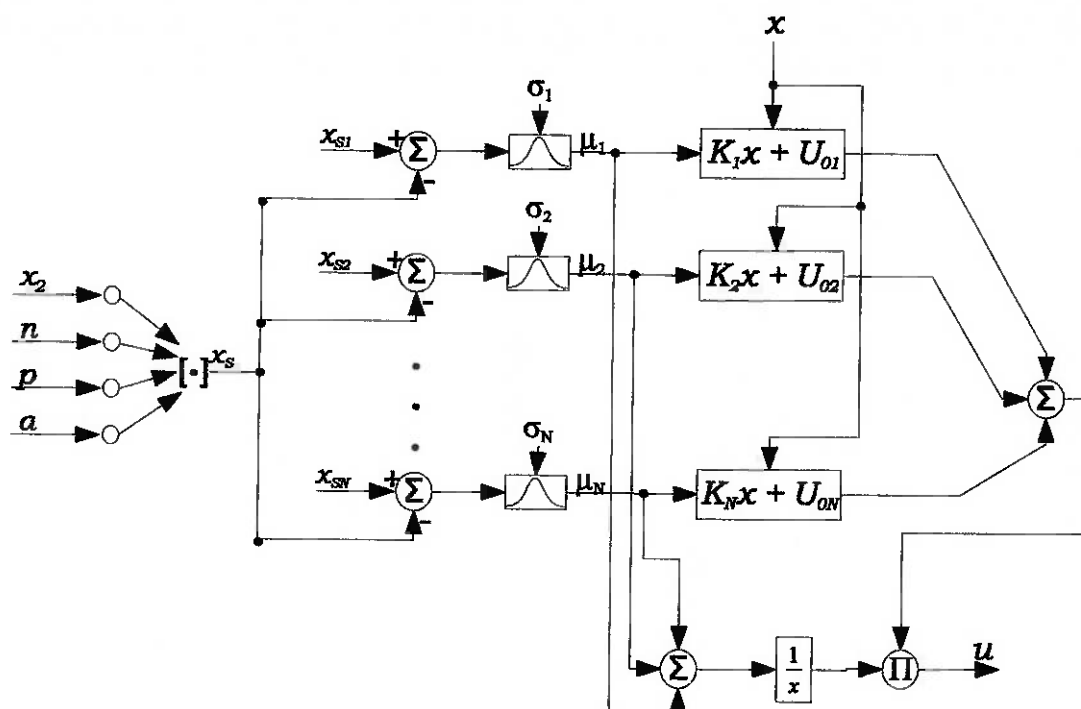


Figura 6.6: Controlador fuzzy-neural

a serem adaptados, e que armazena o conhecimento de forma local, é relativamente simples e apresenta boas características de convergência. Baseado em tudo isso, e também se inspirando no artigo de (Branco & Dente 1998), considerou-se interessante introduzir uma matriz de pesos, multiplicando o termo com o controlador, de modo que a equação 6.4 fica:

$$\bar{u} = \frac{\sum_{i=1}^N \mu_i(\bar{x}_s) [\mathbf{W}_i] \left( [\mathbf{K}_i] \bar{x} + u_{oi} \right)}{\sum_{i=1}^N \mu_i(\bar{x}_s)} \quad (6.5)$$

onde  $[\mathbf{W}_i]$  é uma matriz quadrada diagonal, do mesmo tamanho de  $[\mathbf{K}_i]$ . Deste modo, pode-se aplicar o algoritmo de retropropagação facilmente a essa rede.

A equação 6.5 é praticamente a forma final do controlador, faltando apenas definir os valores para os três conjuntos de parâmetros  $x_{si}$ ,  $\sigma_{ij}$ ,  $[\mathbf{W}_i]$ . Durante este processo se verá que serão necessárias duas modificações: a introdução de uma normalização para os valores de entrada e uma normalização dos valores de saída, para facilitar respectivamente, a definição de  $\sigma_{ij}$  e o treinamento de  $[\mathbf{W}_i]$ .

### 6.2.1 Definição dos parâmetros do controlador

Os parâmetros que precisam ser definidos para este controlador são, para cada regra:

- $\mathbf{x}_{Si}$ , o centro da função gaussiana;
- $\sigma_{ij}$ , um vetor com os raios da função gaussiana para cada uma das dimensões do ítem anterior;
- $[\mathbf{W}_i]$ , a matriz de pesos do controlador.

O primeiro passo é definir o valor de  $\mathbf{x}_{Si}$ . A tabela 6.2 mostra os valores de alguns dos pontos fixos para os controladores. Um fato importante que se percebe desta tabela é que as variáveis  $x_1$ ,  $x_3$  e  $x_4$  têm o mesmo valor de estabilização para todos os controladores, e logo não podem se prestar para diferenciar um do outro. Portanto, fica claro que não é necessário utilizar todas as variáveis de estado estendidas. Pode-se redefinir o vetor  $\mathbf{x}_S$  como:

$$\mathbf{x}_S = \begin{bmatrix} x_2 \\ p \\ n \\ \alpha \end{bmatrix} \quad (6.6)$$

Assim, ficam definidos os  $\mathbf{x}_{Si}$  para cada regra.

Com relação a escolha dos  $\sigma_{ij}$ , existem milhares de métodos que podem ser utilizados. Wang (1994) sugere como um “chute” inicial que se encontre a amplitude da excursão de cada uma das variáveis de estado (que chamaremos de  $x_{jmax}$  e  $x_{jmin}$ ), e que se faça:

$$\sigma_{ij} = \sigma_j = \frac{x_{jmax} - x_{jmin}}{N} \quad (6.7)$$

Este chute seria então refinado através de um aprendizado por retropropagação. Como já foi dito anteriormente, este método apresentou um problema de convergência muito lenta, e em alguns casos instabilidade. O método que foi escolhido, por sugestão de Pedro Sérgio<sup>5</sup>, e que é o mais condizente com a “filosofia”

---

<sup>5</sup>Ver a seção de agradecimentos

$\alpha$	$x_1$	$x_2$	$x_3$	$x_4$	$p$	$n$
30°	0	$1,6367 \cdot 10^{-7}$	0,01	0,005	36022,0000	133,7641
35°	0	$2,1825 \cdot 10^{-7}$	0,01	0,005	43113,0000	149,0364
40°	0	$2,7842 \cdot 10^{-7}$	0,01	0,005	47634,0000	172,0766
45°	0	$3,4302 \cdot 10^{-7}$	0,01	0,005	53891,0000	187,3875
50°	0	$4,0755 \cdot 10^{-7}$	0,01	0,005	59150,0352	202,8447
55°	0	$4,6875 \cdot 10^{-7}$	0,01	0,005	63710,0435	216,6058
60°	0	$5,2513 \cdot 10^{-7}$	0,01	0,005	67710,0553	228,3233
65°	0	$5,7566 \cdot 10^{-7}$	0,01	0,005	71210,0603	237,9922
70°	0	$6,2058 \cdot 10^{-7}$	0,01	0,005	74150,0697	246,3933
75°	0	$6,6017 \cdot 10^{-7}$	0,01	0,005	76600,0926	253,7272
80°	0	$6,9404 \cdot 10^{-7}$	0,01	0,005	78700,0719	259,6262
85°	0	$7,4205 \cdot 10^{-7}$	0,01	0,005	79000,1310	276,5330
90°	0	$7,8036 \cdot 10^{-7}$	0,01	0,005	79703,9510	288,2412

Tabela 6.2: Valores de estabilização

fuzzy, é definir  $\sigma_{ij}$  a partir da proximidade dos controladores no espaço de estado. A figura 6.7 mostra uma série de pontos em um espaço de estados bidimensional. Para calcular o valor de  $\sigma$  para  $P_3$ , por exemplo, deve-se procurar o ponto mais próximo (que no caso é  $P_1$ ), e definir  $\sigma$  a partir da distância  $d_{31}$  entre estes pontos (e.g.  $\sigma_3 = 2d_{31}$ ). É importante notar que, mesmo que  $\sigma_3 = kd_{31}$ ,  $\sigma_1$  não precisa ser obrigatoriamente  $\sigma_1 = kd_{13}$ , uma vez que pode haver um ponto mais próximo de  $P_1$  que  $P_3$ . No caso da figura, o ponto mais próximo de  $P_1$  é  $P_5$ .

A dificuldade com esta abordagem é a diferença de escala entre as várias variáveis de estado. Pela tabela 6.2, percebe-se que existe uma diferença de 11 ordens de grandeza entre  $x_2$  e  $p$ . Sempre que existe alguma diferença de escala, se torna inválido o uso de qualquer medida de distância da forma  $\|\mathbf{x} - \mathbf{y}\|_n$ .

Um modo de resolver este problema é aplicar uma normalização às variáveis de estado, fazendo com que todas variem dentro de uma mesma faixa, por exemplo, entre 0 e 1. Deste modo faz sentido pensar em termos de distância no espaço de estados.

Para realizar a normalização, basta determinar os valores mínimo e máximo para cada variável de estado e mapear esses valores para 0 e 1. Depois os valores intermediários são mapeados linearmente entre 0 e 1.

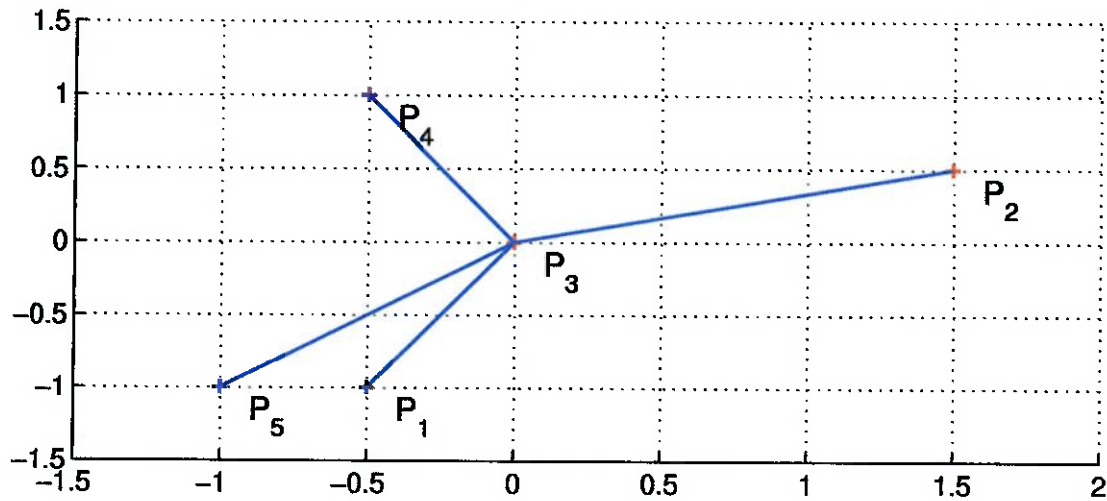


Figura 6.7: Escolha do controlador mais próximo no espaço de estados

Utilizando este método, ao invés de se definir vários  $\sigma_{ij}$  para cada regra, é preciso definir apenas um  $\sigma_i$ . Também, ao invés de se calcular várias gaussianas para cada regra, e depois multiplicá-las entre si, é preciso calcular apenas uma, e uma norma euclidiana, o que torna o processamento mais rápido, uma vez que o cálculo da função gaussiana é relativamente lento<sup>6</sup>. Deste modo, a equação 6.3 fica simplesmente:

$$\mu_i(\mathbf{x}) = e^{-\left(\frac{\|\mathbf{x}_s - \mathbf{x}_{si}\|_2}{\sigma_i}\right)^2} \quad (6.8)$$

Finalmente, pode-se passar a determinação dos  $\sigma_i$ . Foi definida uma constante  $\xi_i$ , tal que, sendo  $k$  o ponto mais próximo de  $i$ ,

$$d_{ik} = \xi_i \sigma_i \quad (6.9)$$

Na implementação atual, ao invés de se definir um valor diferente de  $\xi_i$  para cada regra, decidiu-se fazer  $\xi_i = \xi$ . Isto porque não existe nenhuma regra ou heurística que auxilie na definição deste valor individualmente para cada regra<sup>7</sup>. Os resultados das simulações mostraram que o comportamento do controlador

<sup>6</sup>Estes problemas de velocidade de processamento serão tratados mais a fundo na seção 6.3

<sup>7</sup>Uma técnica que merece ser investigada é a utilização de uma otimização por algoritmos genéticos. Mas isto foge ao escopo deste trabalho



muda brutalmente com o valor de  $\xi$ . Os gráficos de 6.8 a 6.11 mostram a variação da resposta do controlador para diferentes valores de  $\xi$ .

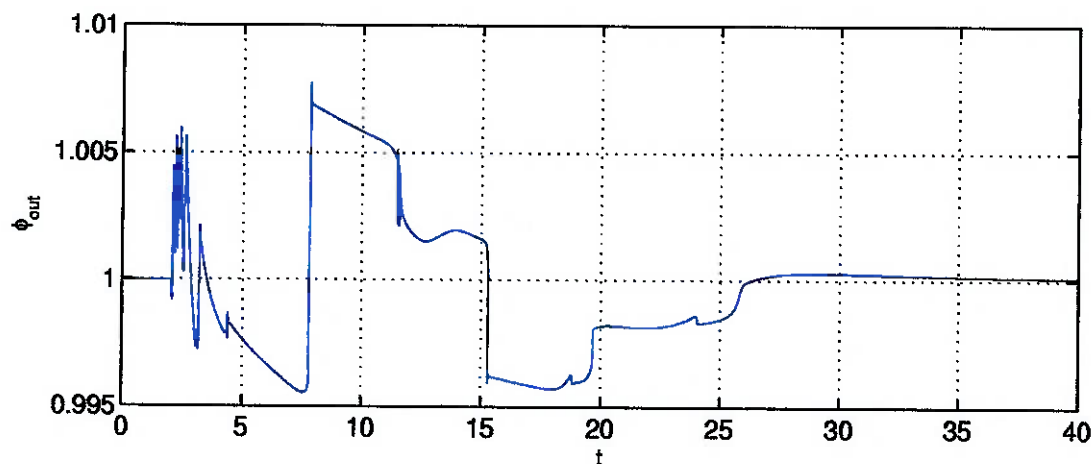


Figura 6.8: Simulação para uma variação do ângulo do pedal de 50 a 80 graus, com  $\xi = 4$ .

Duas características interessantes com relação ao valor de  $\xi$  podem ser extraídas dos gráficos. A primeira é que, quanto menor o valor de  $\xi$ , mais “suave” se torna o controlador. A segunda é que, em contrapartida, os erros de *offset* se tornam maiores (nas figuras 6.8 e 6.9 praticamente não existe erro de *offset*).

Uma terceira característica, que aparece para  $\xi$  muito grande ou muito pequeno está relacionada com erros numéricos durante o cálculo. Caso  $\xi$  seja muito pequeno, para a maior parte do espaço de estados os graus de pertinência de todos os controladores estarão muito próximos de 0. Devido aos problemas de precisão finita dos microprocessadores, caso o motor passe por uma região desse tipo, não será possível interpolar um controlador, ou porque todos os graus de pertinência serão identicamente 0, ou porque estarão tão próximos de 0 que estarão próximos da resolução mínima do processador<sup>8</sup>. Neste caso, ou ocorrerá um erro de divisão por zero, ou o controlador interpolado não será válido. Do mesmo modo, se  $\xi$  for muito pequeno, ocorrerá um problema semelhante, mas com os valores de

<sup>8</sup>A resolução mínima de um processador é a menor diferença possível entre dois números. Para os processadores da família x86 esta resolução é da ordem de  $10^{-17}$ .

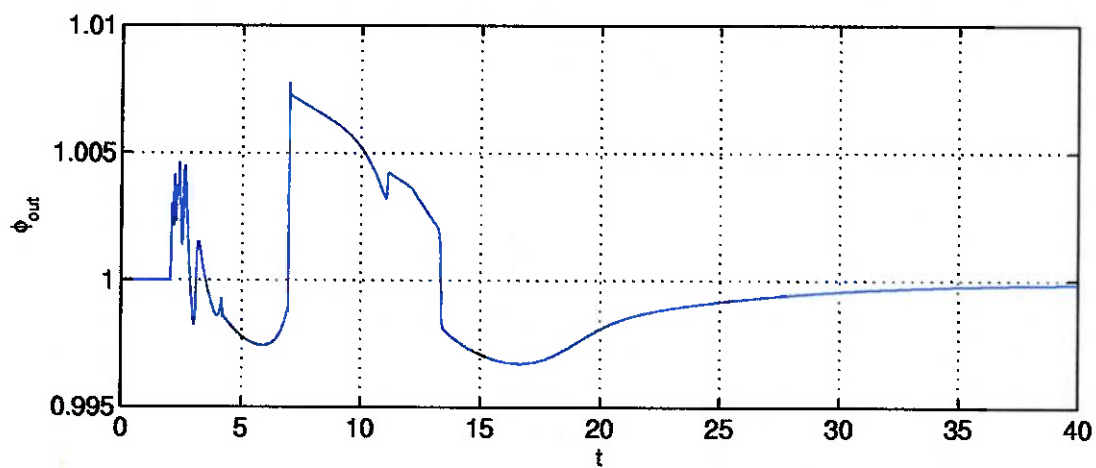


Figura 6.9: Simulação para uma variação do ângulo do pedal de 50 a 80 graus, com  $\xi = 2$ .

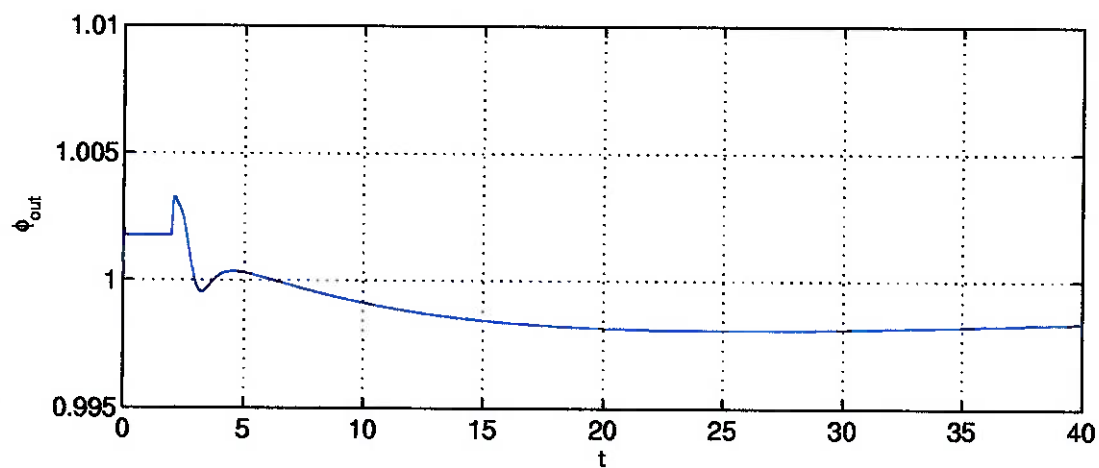


Figura 6.10: Simulação para uma variação do ângulo do pedal de 50 a 80 graus, com  $\xi = 0,5$ .

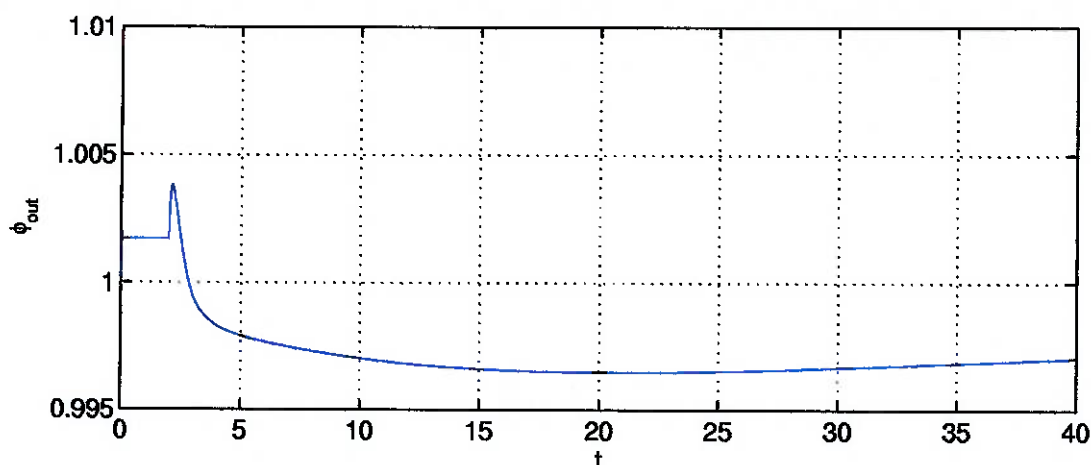


Figura 6.11: Simulação para uma variação do ângulo do pedal de 50 a 80 graus, com  $\xi = 0,3$ .

pertinência sendo todos muito próximos de 1.

Apesar dos erros de *offset*, os controladores com  $\xi = 0,5$  e  $\xi = 0,3$  tiveram uma resposta melhor (no sentido de se desviarem bem menos do valor objetivo), e portanto pode-se supor que, após aplicado o treinamento por retropropagação, será possível eliminar em grande parte os erros de *offset*, e se obter um bom controlador.

Como foi mostrado no capítulo 5, o algoritmo de retropropagação consiste em um ajuste de um vetor de pesos  $w$  a partir da derivada do erro de saída da rede. No entanto, a equação 5.5 é válida apenas para um sistema com uma única saída. Novamente, uma rede com múltiplas saídas pode ser considerada como múltiplas redes, cada uma com apenas uma saída. A pequena dificuldade com esta abordagem é ter que definir um valor de  $\alpha$  para cada uma das saídas<sup>9</sup>, pois, devido a diferença de valores, cada uma terá um valor diferente, e um limite de convergência diferente<sup>10</sup>. Dependendo dos valores escolhidos para  $\alpha$ , a velocidade de convergência também pode ser bastante diferente para cada uma das saídas.

<sup>9</sup> $\alpha$  é a constante de aprendizado.

<sup>10</sup>Conforme estabelecido na tabela 5.1.

Uma solução para este problema é novamente fazer a normalização das saídas. Deste modo, as saídas produzidas por cada uma das regras devem estar entre 0 e 1. Isto consiste em encontrar uma matriz  $[A]$  e um vetor  $B$  de modo que:

$$[A]u_i + B = u_{Ni} \quad (6.10)$$

Onde  $u_{Ni}$  é o vetor de saídas normalizadas. Um outro modo de obter  $u_{Ni}$  é converter a matriz  $[K_i]$  e o vetor  $u_{oi}$  de modo a criar um novo controlador

$$u_{Ni} = u_{Ni}(\bar{x}) = [K_{Ni}]\bar{x} + u_{Noi} \quad (6.11)$$

onde

$$\begin{aligned} [K_{Ni}] &= [A][K_i] \\ u_{Noi} &= [A]u_{oi} + B \end{aligned} \quad (6.12)$$

Para simplificar a notação, não será mais utilizado o subscrito  $N$ , ficando implícito que se está tratando sempre da saída normalizada. Com esta operação, chegamos a forma final do controlador, que está representado no diagrama 6.12.

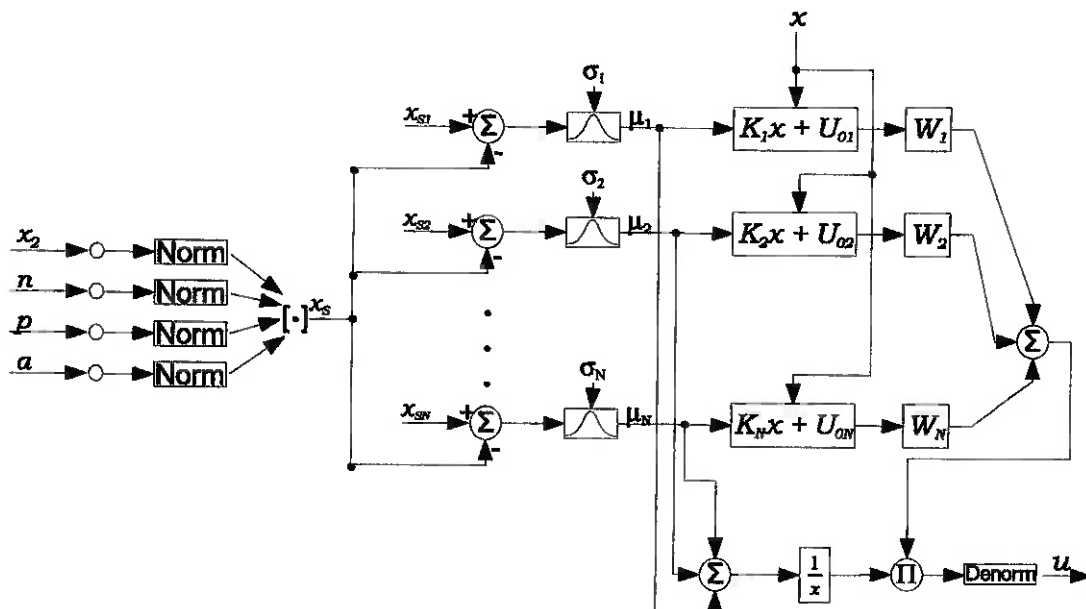


Figura 6.12: Diagrama do controlador

Com as saídas normalizadas, pode-se proceder com o treinamento da matriz  $[\mathbf{W}_i]$  de cada regra. Aplicando a equação 5.5 ao controlador, chega-se a seguinte matriz de atualização de pesos:

$$[\Delta \mathbf{W}_i] = C \left( -\alpha \frac{(\mathbf{u} - \mathbf{u}_l) \mu_i y_i}{\sum_{i=1}^N \mu_i} \right) \quad (6.13)$$

Onde a função  $C(\cdot)$  é definida como:

$$\begin{bmatrix} a_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_N \end{bmatrix} = C \left( \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} \right) \quad (6.14)$$

Esta função é necessária apenas para transformar um vetor em uma matriz diagonal (isto é preciso pelo modo como foi definido  $[\mathbf{W}_i]$ ).

A partir daí, o treinamento procede da seguinte maneira:

Dado um conjunto de  $N$  pares de entrada e saída  $\{\mathbf{x}_{sl}, \mathbf{u}_l\}$ , para cada par  $l$ , adiciona-se em cada uma das matrizes de peso o valor de  $[\Delta \mathbf{W}_i]$  correspondente àquela regra. Iterar este processo até que se obtenha um valor razoável para o erro<sup>11</sup>, ou que um número máximo de iterações seja alcançado.

Os gráficos 6.13 a 6.15 mostram os erros médios de treinamento para os controladores com  $\xi = 0,5$  e  $\xi = 0,3$ . O erro médio de cada iteração é a média dos erros de cada par de treinamento. Desses gráficos se percebe dois fatos importantes. O primeiro é que a taxa de aprendizado (ou a taxa de diminuição do erro) é grande para as primeiras iterações, e depois diminui até se tornar bastante pequena. A partir daí normalmente são necessários muitas iterações para obter uma diminuição bastante pequena no erro. O segundo fato é que a rede com  $\xi = 0,5$  aprende muito mais rápido que a rede com  $\xi = 0,3$ . Um efeito que também deve ser levado em consideração é a interferência de aprendizado. Isto também está relacionado com o valor de  $\xi$ , e quanto menor este valor, maior a interferência. A interferência aparece como uma diminuição global do erro médio, mas o aumento do erro de alguns pares de aprendizado específicos. Por esta razão que os resultados mostrados no capítulo 7 farão uma comparação entre o controlador LQ chaveado, e os três

<sup>11</sup>O erro é como o definido na equação 5.2, e vale  $\epsilon = 0,5(\mathbf{u} - \mathbf{u}_l)^2$

controladores mostrados até agora: o com  $\xi = 0,005$  e 1000 passos de treinamento (controlador 1), o com  $\xi = 0,003$  e 3000 passos de treinamento (controlador 2) e o com  $\xi = 0,003$  e 10000 passos de treinamento (controlador 3). Ficará claro que, em alguns casos, que o controlador 3 tem um maior erro de *offset* que o controlador 2, apesar de ter um erro médio menor.

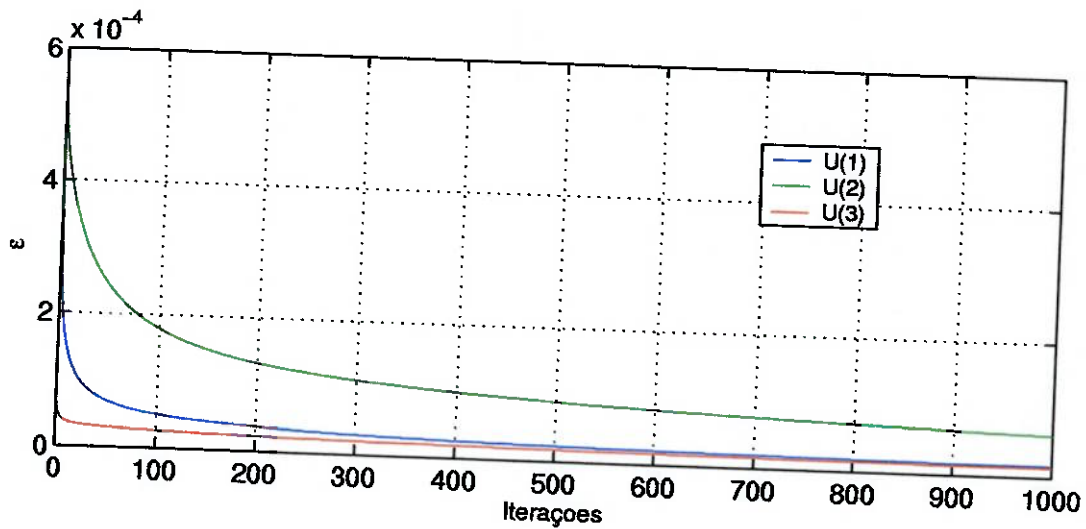


Figura 6.13: Erro médio de treinamento para a o controlador utilizando  $\xi = 0,5$

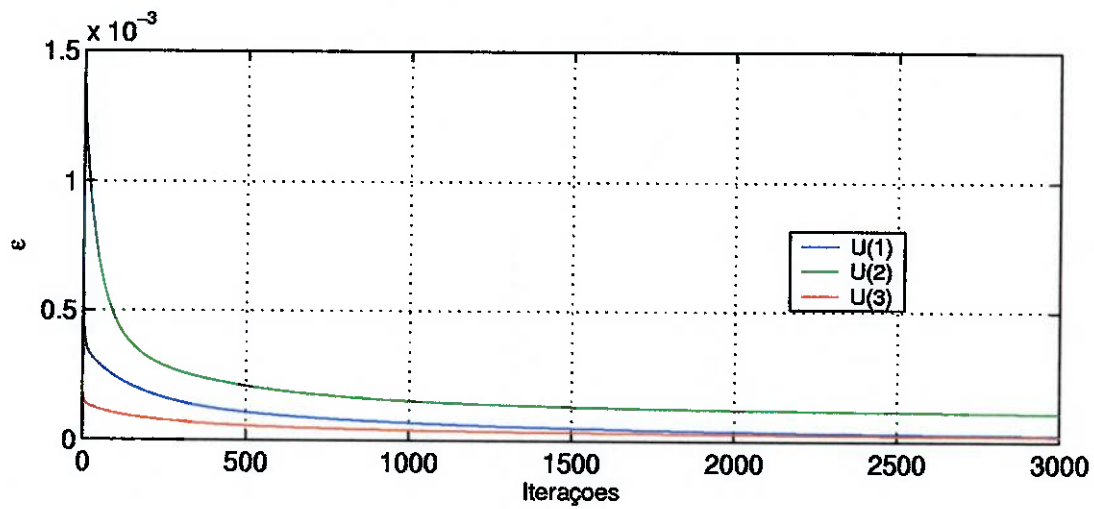


Figura 6.14: Erro médio de treinamento para a o controlador utilizando  $\xi = 0,3$

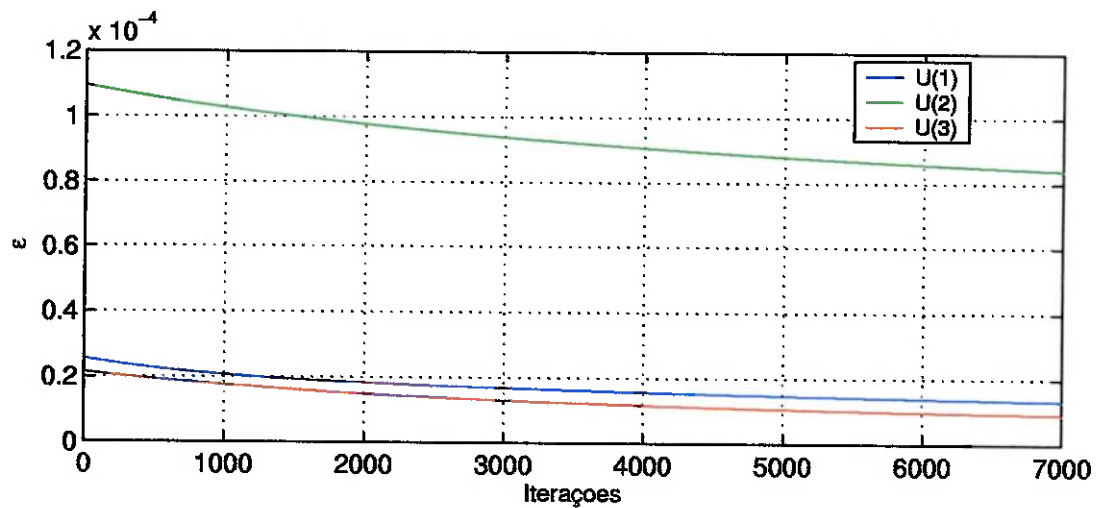


Figura 6.15: Erro médio de treinamento para a o controlador utilizando  $\xi = 0,3$ , continuando a partir do gráfico 6.14, totalizando 10000 iterações

### 6.3 Detalhes de implementação

Uma das vantagens do controlador LQ com chaveamento brusco é que este é um controlador extremamente barato em termos computacionais. A tabela faz uma comparação entre o número de operações necessárias a cada ciclo para o controlador LQ e para o controlador fuzzy-neural proposto na seção anterior.

Controlador LQ	Controlador fuzzy-neural
1 operação de busca	1 normalização de vetor
1 multiplicação de matrizes	$N$ subtrações de vetor
1 soma de vetores	$N$ normas euclidianas
	$N$ cálculos de gaussianas
	$N$ somas de vetor
	$N$ somas de escalar
	$N$ multiplicação de matriz
	$N$ multiplicação por matriz ( $[W]$ )
	1 divisão de vetor por escalar
	1 denormalização de vetor

Tabela 6.3: Comparação entre o número de operações realizadas para o controlador LQ com chaveamento brusco e para o controlador fuzzy-neural

Percebe-se claramente que o custo de implementação do controlador neuro-fuzzy é muito mais alto, apesar das vantagens de eficiência que ficarão claras no capítulo 7. O principal objetivo desta seção é mostrar que existe uma forma de diminuir significativamente o número de cálculos necessários.

A idéia básica é que, em cada instante, apenas algumas regras estão ativadas, e todas as outras têm graus de pertinência zero ou muito próximos disso. Portanto, caso seja possível determinar *a priori* quais regras estarão ativas e quais estarão inativas, é possível diminuir brutalmente o número de operações realizadas.

Existem vários métodos de realizar esta operação. Um deles é definir um  $\alpha$ -corte nas funções gaussianas, de modo que, por exemplo, sempre que a função gaussiana for menor que 0,001 ela valerá zero. Pode-se então, ainda durante a fase de projeto, qual o valor de  $\|x_s - x_{si}\|_2$  que produz 0,001 na gaussiana. Então, ao invés de se calcular a gaussiana, realiza-se um teste para ver se  $x_s$  irá ou não



ativar essa regra. Um outro método, citado em (Brown & Harris 1994), é utilizar uma variação da gaussiana que tem um suporte compacto:

$$\mu_i(x) = \begin{cases} \ln(-1)e^{-\frac{(\lambda_{h,2}-\lambda_{h,1})^2}{4(\lambda_{h,2}-x)(x-\lambda_{h,1})}} & \text{Se } x \in (\lambda_{h,1}, \lambda_{h,2}) \\ 0 & \text{Se } x \notin (\lambda_{h,1}, \lambda_{h,2}) \end{cases} \quad (6.15)$$

É claro que quanto menor o valor de  $\xi$ , maior o número de regras que estarão ativadas em um determinado instante, e portanto menos efetiva esta abordagem.

## Capítulo 7

### Resultados

Neste capítulo serão apresentados os resultados de uma série de simulações, comparando os desempenhos dos 3 controladores projetados no capítulo anterior com o controlador LQ com chaveamento brusco. Todas as simulações mostram a resposta do motor a variações em degrau no ângulo de abertura da borboleta. A tabela 7.1 mostra os valores iniciais e finais dos ângulos de abertura da borboleta, assim como os parâmetros de desempenho para cada um dos controladores. Nesta tabela, os valores dos ângulos estão em graus, o valor de  $\phi_{eMAX}$  está em porcentagem (e.g. uma excursão de 1 até 0,97 estará indicada como 3%) e o valor de  $\int_C \phi_e$  está multiplicado por 100.

Simulação			Control. LQ		Control. 1		Control. 2		Control. 3	
Nº	$\alpha_0$	$\alpha_\infty$	$\phi_{eMAX}$	$\int_C \phi_e$	$\phi_{eMAX}$	$\int_C \phi_e$	$\phi_{eMAX}$	$\int_C \phi_e$	$\phi_{eMAX}$	$\int_C \phi_e$
1	50	80	2,20	3,88	0,36	0	0,37	0	0,37	0
2	85	40	26,0	111,0	8,31	149,2	7,36	119,2	7,54	134,8
3	40	85	14,6	19,3	1,24	10,6	1,01	1,82	0,97	2,98
4	33	88	35,9	58,0	3,46	23,4	2,68	12,9	2,76	22,9
5	88	33	33,1	265,4	19,2	353,0	17,3	317,3	17,6	355,5
6	50	88	2,43	7,46	0,89	1,8	0,44	0	0,43	0
7	88	50	3,44	30,3	1,16	13,8	1,53	18,0	1,5	17,6

Tabela 7.1: Tabela com os valores utilizados nas simulações e com os parâmetros de performance obtidos.

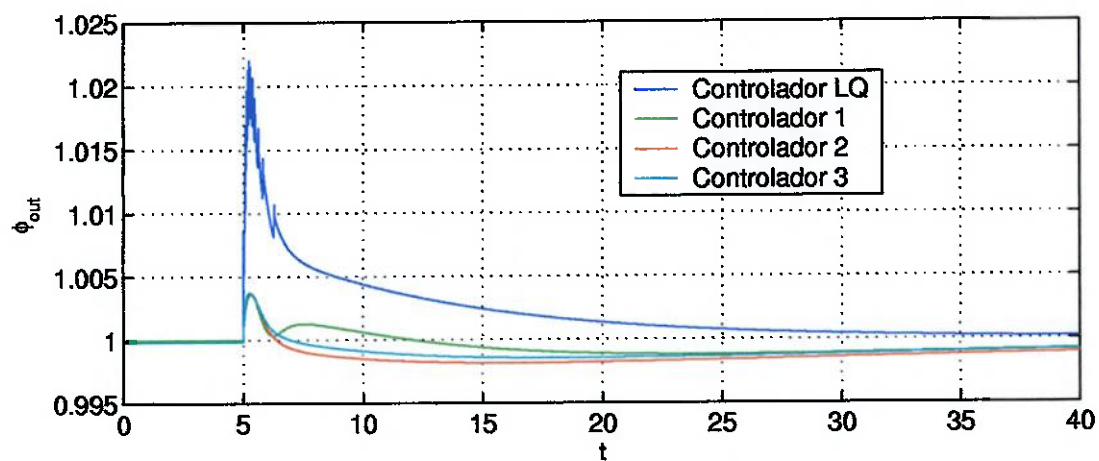


Figura 7.1:  $\phi_e$  para a simulação 1 (de 50 a 80 graus)

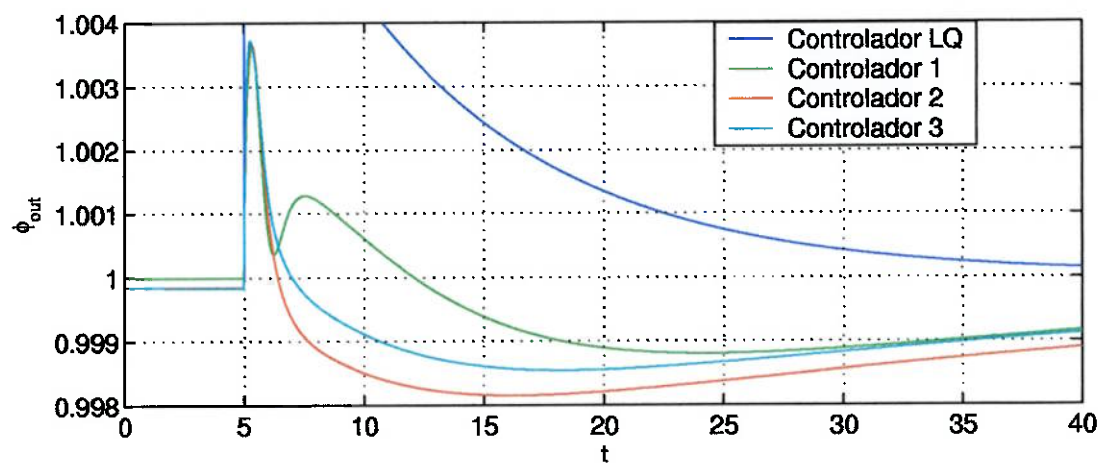


Figura 7.2:  $\phi_e$  para a simulação 1, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3

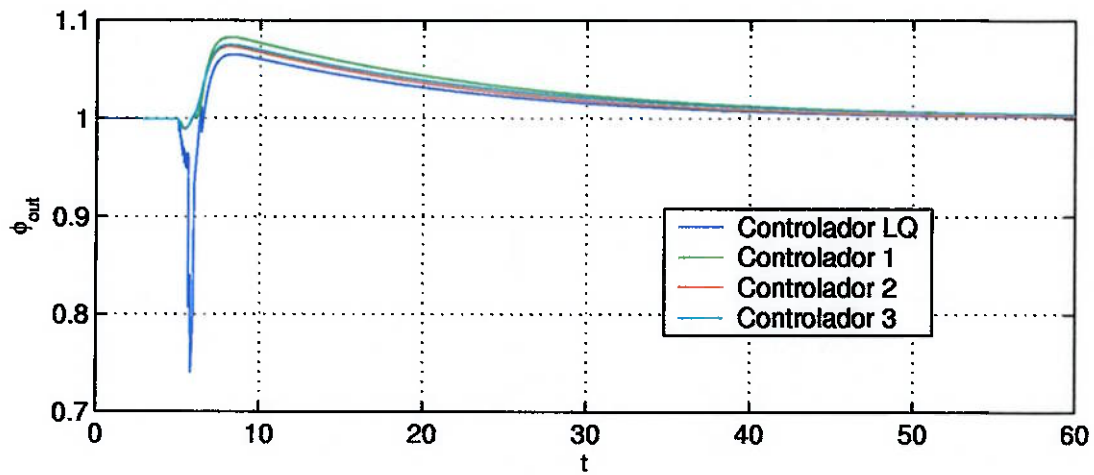


Figura 7.3:  $\phi_e$  para a simulação 2 (de 85 a 40 graus). Esta é a única simulação em que o controlador LQR apresentou um desempenho superior aos fuzzy-neurais

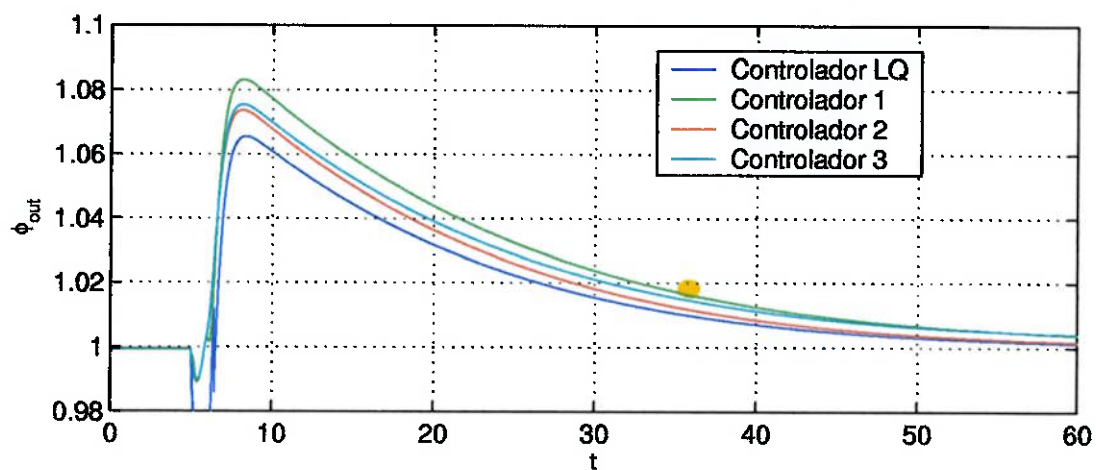


Figura 7.4:  $\phi_e$  para a simulação 2, ampliada para mostrar a resposta dos controladores de 1 a 3.

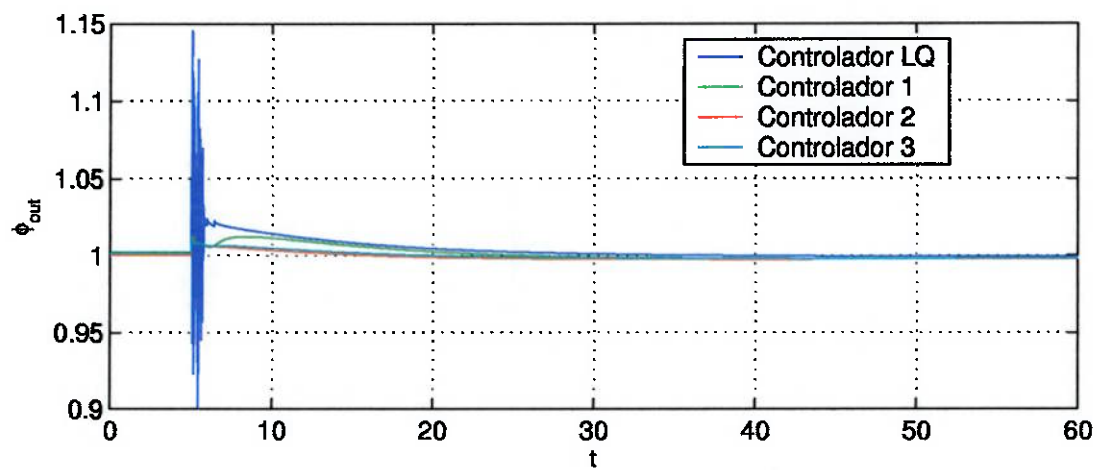


Figura 7.5:  $\phi_e$  para a simulação 3 (de 40 a 85 graus)

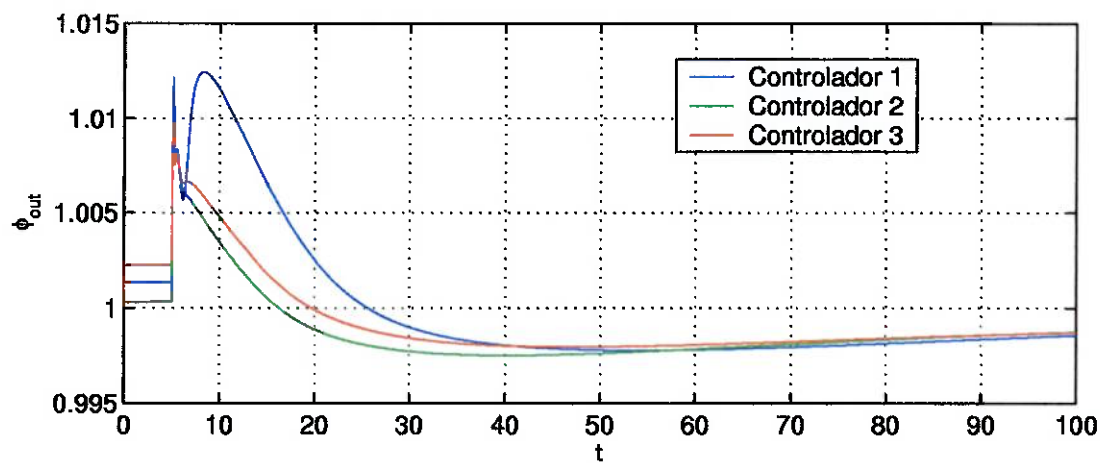


Figura 7.6:  $\phi_e$  para a simulação 3, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3

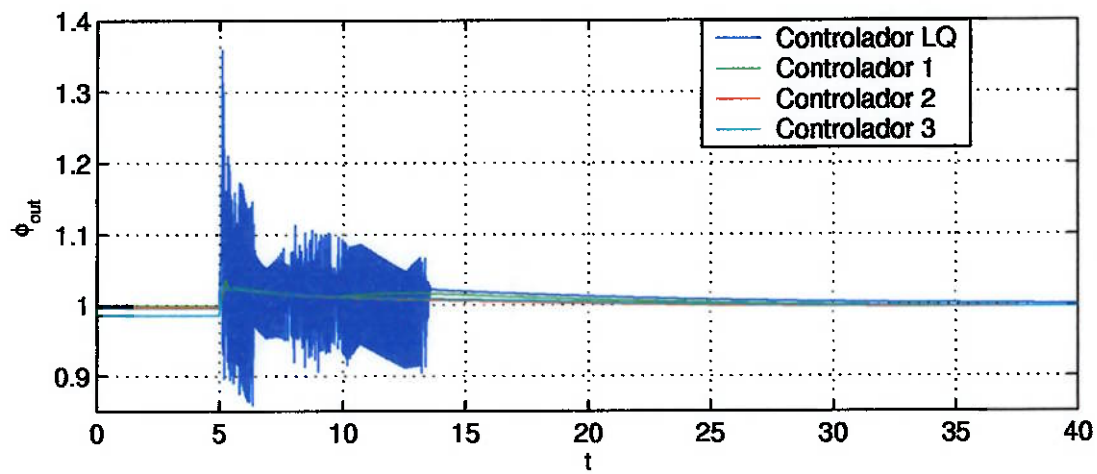


Figura 7.7:  $\phi_e$  para a simulação 4 (de 33 a 88 graus)

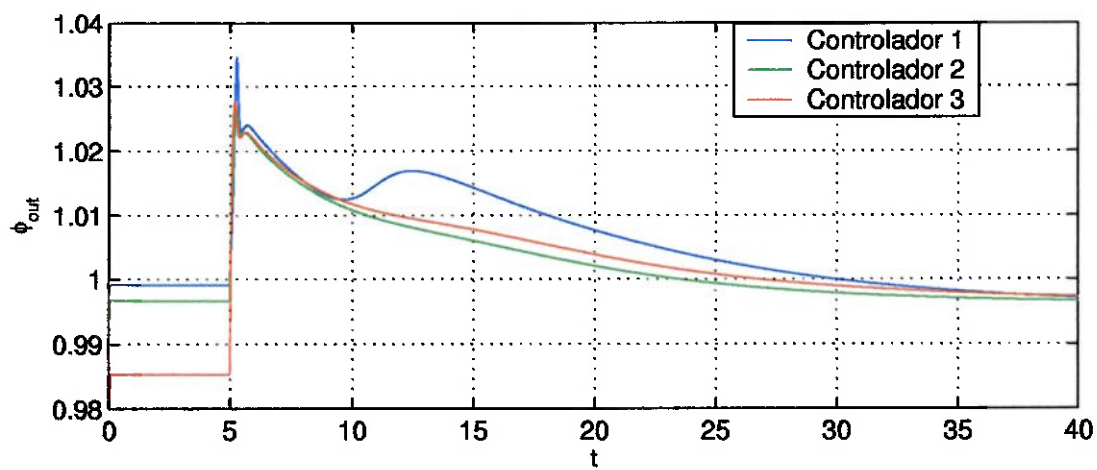


Figura 7.8:  $\phi_e$  para a simulação 4, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3. É interessante notar como o erro de *offset* é significativamente maior para o controlador 3. Isto se deve ao problema de interferência de aprendizado

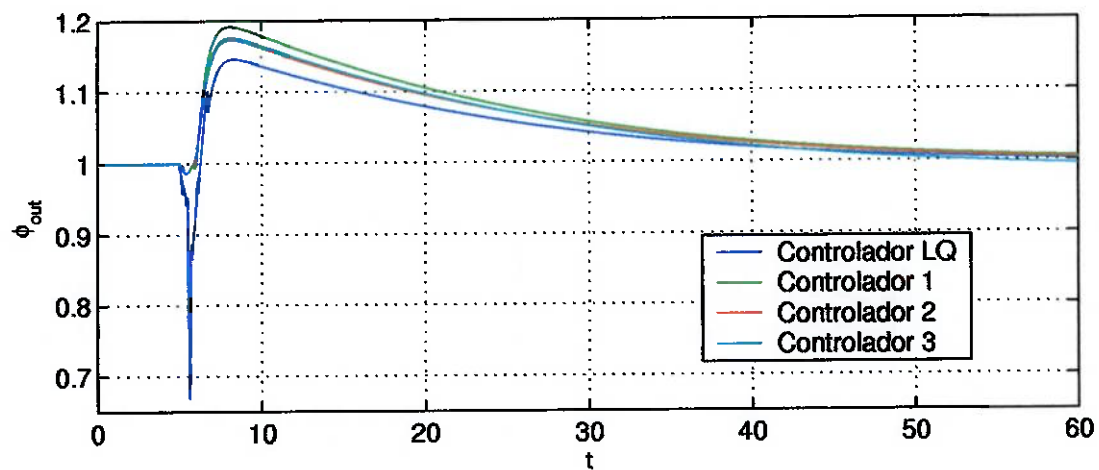


Figura 7.9:  $\phi_e$  para a simulação 5 (de 88 a 33 graus)

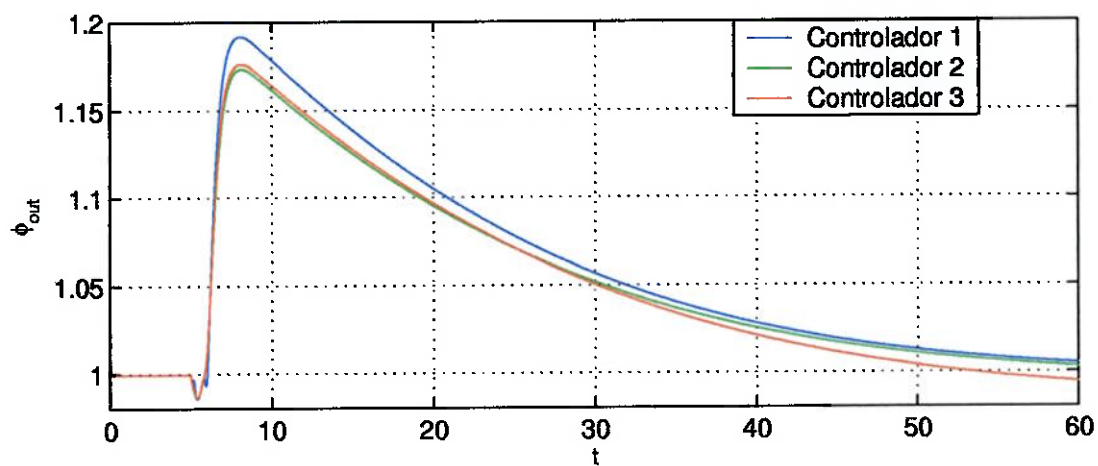


Figura 7.10:  $\phi_e$  para a simulação 5, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3.

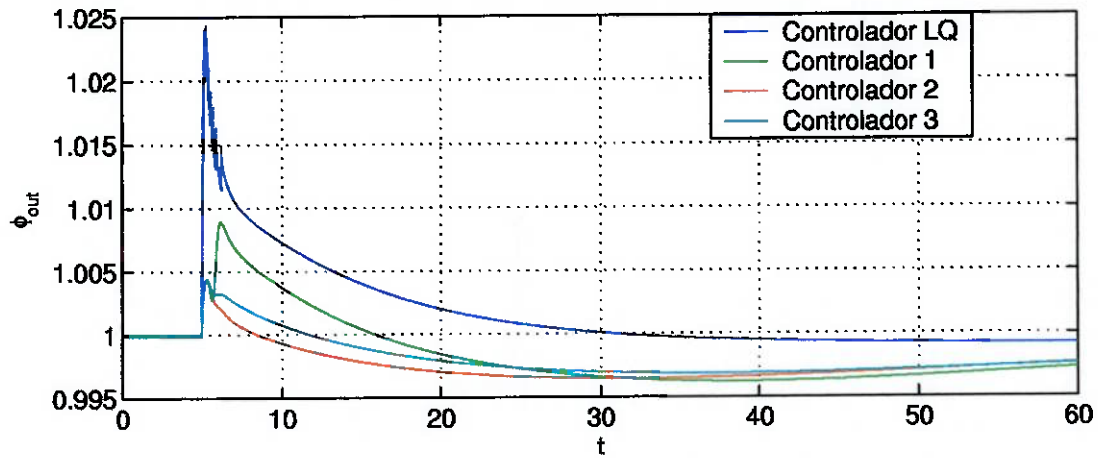


Figura 7.11:  $\phi_e$  para a simulação 6 (de 50 a 88 graus)

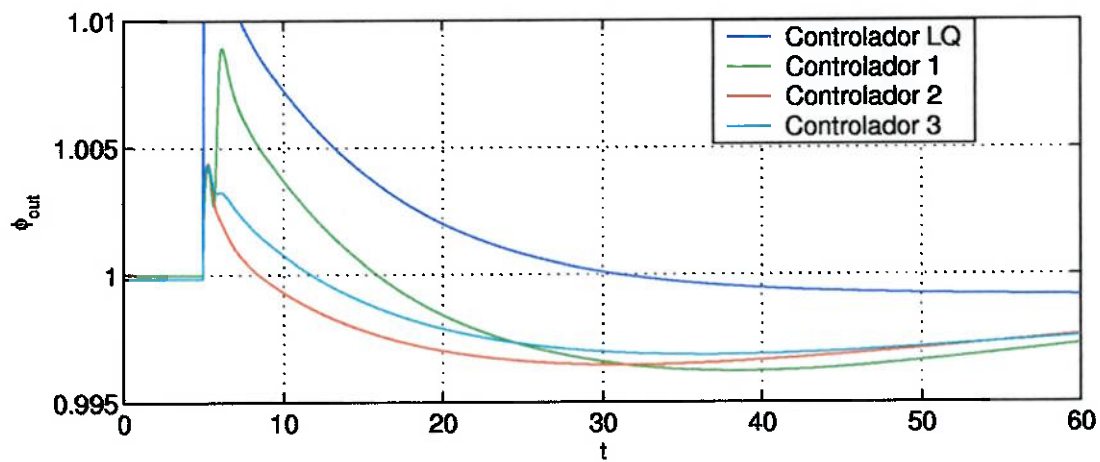


Figura 7.12:  $\phi_e$  para a simulação 6, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3.



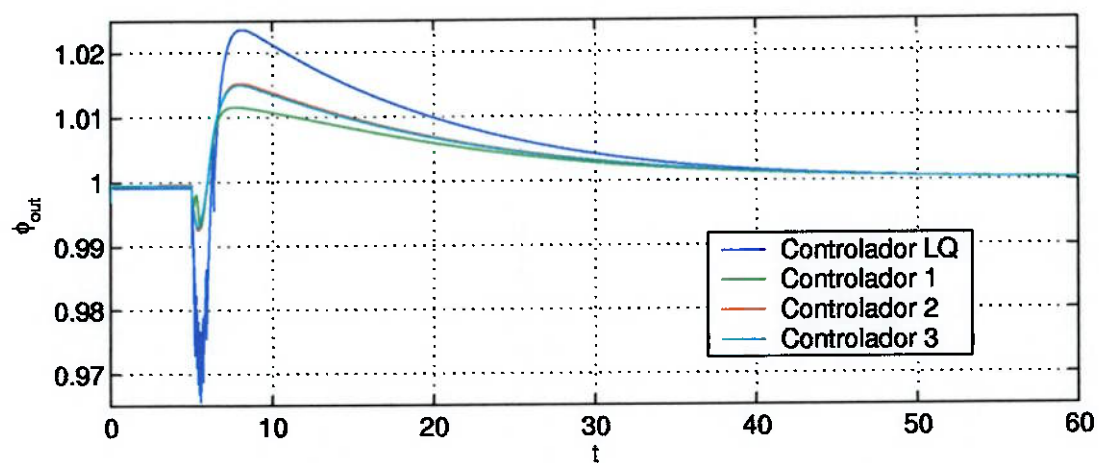


Figura 7.13:  $\phi_e$  para a simulação 7 (de 88 a 50 graus)

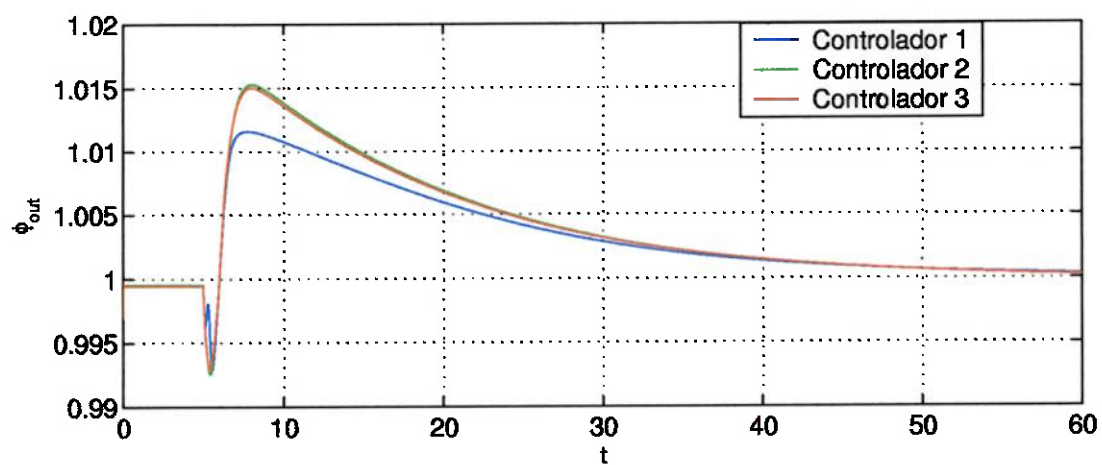


Figura 7.14:  $\phi_e$  para a simulação 7, ampliada para mostrar a resposta dinâmica dos controladores de 1 a 3.

## 7.1 Análise dos resultados

Dos valores da tabela 7.1, percebe-se que, em termos de máxima excursão, os resultados dos controladores fuzzy-neurais é muito melhor que o do controlador LQ. Também se percebe que não existe uma diferenciação clara entre o desempenho dos controladores fuzzy-neurais. Quando se passa para a análise de  $\int_C \phi_e$ , percebe-se que os controladores com  $\xi = 0,3$  são superiores que o com  $\xi = 0,5$ . Outro fato interessante que fica claro é que o desempenho desses controladores é muito melhor quando o ângulo de abertura da borboleta está aumentando do que quando este está diminuindo. Para o caso de  $\alpha$  diminuindo, em alguns casos o controlador LQ apresenta um desempenho melhor. Uma outra característica interessante é notar como as dificuldades de aprendizado interferem no erro de *offset*. No gráfico 7.8, percebe-se claramente o grande *offset* do controlador 3. Este problema pode ser resolvido com uma mudança do valor de  $\xi_i$ , especificamente para as regras mais próximas daquele ponto. No entanto, isto pode interferir com outros aspectos do desempenho do controlador próximo aquela região.

## Capítulo 8

### Conclusões

Com os dados do capítulo anterior, podemos resumir os seguintes pontos positivos:

- O desempenho do controlador fuzzy-neural com relação a máxima excursão é bastante superior ao do controlador LQ com chaveamento brusco;
- Com relação a  $\int_C \phi_e$ , que funciona como uma medida da quantidade de poluentes produzidos pelo motor, o desempenho do controlador foi superior na maior parte dos casos;

Estes pontos mostram que o controlador desenvolvido é uma alternativa vantajosa ao controlador LQ chaveado. No entanto, existem alguns pontos negativos e outros que precisam de maior estudo. Estes são:

- O custo computacional é mais elevado;
- A seleção de parâmetros é muito empírica, e difícil.

Com relação ao custo computacional, pode-se dizer que o problema não é do controlador fuzzy-neural, que é muito caro, mas sim o controlador LQ que é muito barato. Outras estratégias de controle, como o GPC e LMI, que foram utilizadas por (Freitas Junior 2003) para abordar o mesmo problema também possuem um custo computacional bastante elevado.

Quanto à dificuldade de se determinar os parâmetros para o controlador, estas dificuldades podem ser superadas com o uso de alguma técnica de otimização que lide com problemas não lineares, como algoritmos genéticos ou buscas randômicas.

O uso destas técnicas pode inclusive trazer um grande aumento de desempenho para o controlador, pois permite fugir da restrição imposta de que  $\xi_i = \xi$ . O grande problema desta restrição é que ela na verdade ignora a abrangência de cada controlador no espaço de estados. Considera-se que este é o maior problema com a implementação atual, pois influencia tanto a resposta dinâmica quanto o *offset* do controlador.

## Bibliografia

- Branco, P. C. & Dente, J. (1998), Design of an electro-hydraulic system using neuro-fuzzy techniques, *in* Jain & Martin (1998), chapter 4.
- Brown, M. & Harris, C. (1994), *Neurofuzzy Adaptive Modelling and Control*, International Series in Systems and Control, Prentice Hall.
- Cook, P. A. (1993), *Nonlinear Dynamical Systems*, Prentice Hall international series in systems and control engineering, segunda edn, Prentice Hall.
- Franklin, G. F., Powell, J. D. & Workman, M. (1997), *Digital Control of Dynamic Systems*, terceira edn, Addison Wesley.
- Freitas Junior, F. S. (2003), Técnicas de controle de emissões baseadas em múltiplos modelos para motores de combustão interna movidos a gás natural. Tese apresentado à Escola Politécnica da Universidade de São Paulo para a obtenção do título de Doutor em Engenharia.
- Goldberg, D. E. (1989), *Genetic Algorithms in search, optimization, and machine learning*, Addison-Wesley.
- Hofstadter, D. R. (1979), *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books.
- II, R. J. M., ed. (1994), *Fuzzy logic technology and applications*, IEEE Technology Update Series, IEEE technical Activities Board.
- Jain, L. C. & de Silva, C. W., eds (1999), *Intelligent Adaptive Control: Industrial Applications*, International Series on Computational Intelligence, The CRC Press.

- Jain, L. C. & de Silva, C. W., eds (1999), *Intelligent Adaptive Control: Industrial Applications*, International Series on Computational Intelligence, The CRC Press.
- Jain, L. C. & Martin, N., eds (1998), *Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications*, primeira edn, CRC Press LLC.
- Kandel, A. & Langholz, G., eds (1994), *Fuzzy control systems*, CRC Press.
- Li, H. & Gupta, M. M., eds (1995), *Fuzzy Logic and Intelligent systems*, International series in intelligent technologies, Kluwer Academic Publishers.
- Lopes, J. A. & Fleury, A. T. (1995), 'A dynamic model for fuel injection /ignition control of natural gas internal combustion engines', *5th Diname-Symposium on Dynamic Problems of Mechanics* pp. 57-60.
- Mandani, E. H. (1994), Twenty years of fuzzy control: Experiences gained and lessons learnt, in II (1994), chapter 1, pp. 19-24.
- Narendra, K. S., Balakrishnan, J. & Ciliz, M. K. (1995), 'Adaptation and learning using multiple models, switching, and tuning', *IEEE Control Systems* 15(3).
- Pedrycz, W. (1993), *Fuzzy control and fuzzy systems*, segunda edn, Research Studies Press.
- Pedrycz, W. (1995), *Fuzzy Sets Engineering*, CRC Press.
- Reyero, R. & Nicolás, C. F. (1995), *Sistemas de control basados en lógica borrosa*, OMRON Electronics S.A.
- Ruspini, E. H., Bonissone, P. P. & Pedrycz, W., eds (1998), *Handbook of fuzzy computation*, primeira edn, IoP Publishing.
- Silberschatz, A., korth, H. F. & Sudarshan, S. (1999), *Sistema de banco de dados*, terceira edn, Makron Books.

- Terano, T., Asai, K. & Sugeno, M. (1994), *Applied fuzzy systems*, AP professional.
- Tsoukalas, L. H. & Uhrig, R. E. (1996), *Fuzzy and Neural approaches in engineering*, Wiley series on adaptive and learning systems for signal processing, communications, and control, primeira edn, Wiley Interscience.
- Vincent, T. L. & Grantham, W. J. (1997), *Nonlinear and optimal control systems*, John Wiley & Sons, Inc.
- Wang, L.-X. (1994), *Adaptive Fuzzy sytems and control*, Prentice Hall.

