

BC

FD-1297

Newton Maruyama  
Eng. Eletricista, Escola Politécnica da USP, 1987

## **Projeto e Análise de Controladores Fuzzy**

Dissertação apresentada à Escola Politécnica para a obtenção do título de Mestre em Engenharia

Orientador: Prof. Dr. Paulo Eigi Miyagi

São Paulo, 1991

## Agradecimentos

Ao prof. Paulo Eigi Miyagi pela paciente orientação e pelo incentivo durante a realização desse trabalho.

Ao prof. Lucas Antonio Moscato pela oportunidade de trabalhar junto à equipe do *Laboratório de Automação e Sistemas*.

Ao *Conselho Nacional de Energia Nuclear* pela bolsa de estudos concedida durante o período 1988-1989.

A todos os membros da equipe do *Laboratório de Automação e Sistemas (ME-CATRÔNICA)* pelo incentivo e pelo companheirismo decisivos à realização desse trabalho.

Aos colegas do Departamento de Engenharia Mecânica e a todos aqueles que direta ou indiretamente colaboraram com a realização desse trabalho.

**Aos meus pais  
por tudo**

## Resumo

Esse trabalho reúne alguns dos resultados mais relevantes relacionados ao projeto e análise de controladores fuzzy.

A construção de um sistema de controle fuzzy é baseada na idéia de se incorporar *experiência* ou *conhecimento especialista* para se obter a melhor estratégia de controle. O fato dessa técnica estar baseada em conhecimento especialista representa ao mesmo tempo o seu ponto forte e o seu ponto fraco. O grande desafio é o estabelecimento de uma teoria de *sistemas dinâmicos fuzzy* que permita o projeto e análise sistemáticos de controladores fuzzy.

Os resultados apresentados nesse trabalho permitem determinar algumas diretrizes gerais para o projeto de controladores fuzzy.

Inicialmente, apresenta-se os conceitos básicos da teoria de conjuntos e lógica fuzzy relevantes para o desenvolvimento de controladores fuzzy.

A arquitetura de controladores fuzzy é apresentada com detalhes e uma discussão à respeito dos parâmetros de projeto é realizada.

Uma relação entre controladores fuzzy e controladores lineares do tipo PID (P,PI,PD) é estabelecida. Determina-se quais são os tipos de regras que devem ser escolhidas para se obter controladores fuzzy do tipo P, PI, PD e PID. Aborda-se a construção de controladores fuzzy aproximadamente lineares e que no limite, número de regras tendendo a infinito, podem ser lineares.

O controlador fuzzy como elemento não linear de controle é explorado através de dois exemplos: o controle de um sistema de primeira-ordem com tempo morto e o controle de um pêndulo invertido.

O trabalho é finalizado com a apresentação de algumas técnicas propostas para a análise de controladores fuzzy.



## Abstract

This work is a survey about some important results related with design and analysis of fuzzy logic controllers.

The design of a fuzzy control system is based on incorporating *experience* or *expert knowledge* to achieve the control goals. The dependence on expert knowledge represents at the same time the power as well as the weakness of this methodology.

The greatest challenge is the establishment of a dynamic fuzzy systems theory that achieves systematic design and analysis.

The results presented in this work determine some general guidelines for the design of fuzzy logic controllers

First, the basic concepts about fuzzy sets and fuzzy logic related with the design of fuzzy logic controllers are presented. A detailed discussion about the basic configuration of fuzzy logic controllers and its structural parameters is presented.

A relation between fuzzy logic controllers and classical linear PID (P, PI, PD) controllers is established. The proper choice of type rule to obtain P, PI, PD and PID fuzzy controllers is demonstrated. It is showed that it is possible to construct aproximate linear fuzzy controllers and in a limit case, infinity number of rules, these controllers are linear.

Two case studies demonstrate the potential of fuzzy controllers as a non linear element of control: a first-order system with dead time and an inverted pendulum.

Finally, some methods proposed for analysis of fuzzy control systems are presented.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>10</b>
1.1	Lógica Fuzzy . . . . .	10
1.2	Controle Fuzzy . . . . .	12
1.3	Motivações do Trabalho . . . . .	16
<b>2</b>	<b>Conjuntos e Lógica Fuzzy</b>	<b>18</b>
2.1	Introdução . . . . .	18
2.2	Definições e Terminologia . . . . .	18
2.3	Operações sobre Conjuntos Fuzzy . . . . .	19
2.4	Variáveis Linguísticas e Conjuntos Fuzzy . . . . .	21
2.5	Lógica Fuzzy e Raciocínio Aproximado . . . . .	22
2.6	Funções de Implicação Fuzzy . . . . .	22
2.7	Interpretação dos conectivos “e” e “também” . . . . .	25
2.8	Operadores Composicionais . . . . .	26
2.9	Mecanismos de Inferência . . . . .	26
2.10	Estratégias de Transformação Fuzzy-Escalar . . . . .	33
<b>3</b>	<b>Controle Fuzzy</b>	<b>35</b>
3.1	Introdução . . . . .	35
3.2	Declarações Condicionais Fuzzy e Regras de Controle Fuzzy . . . . .	36
3.3	Operador de Transformação Escalar-Fuzzy . . . . .	37
3.4	Operadores de Conectivos de Sentença . . . . .	37
3.5	Operador Composicional . . . . .	38
3.6	Operador de Transformação Fuzzy-Escalar . . . . .	38
3.7	Parâmetros de Projeto para um Controlador Fuzzy . . . . .	38
3.7.1	Estratégias de Transformação Escalar-Fuzzy . . . . .	39
3.7.2	Base de Dados . . . . .	40
3.7.3	Base de Regras . . . . .	44
3.7.4	Lógica de Tomada de Decisões . . . . .	46
3.7.5	Estratégias de Transformação Fuzzy-Escalar . . . . .	46
<b>4</b>	<b>Controladores Fuzzy × Controladores Lineares</b>	<b>47</b>
4.1	Introdução . . . . .	47
4.2	Escolha de Variáveis . . . . .	48
4.2.1	Controlador Fuzzy PID . . . . .	48

4.2.2	Controlador Fuzzy Proporcional . . . . .	49
4.2.3	Controlador Fuzzy Proporcional Derivativo . . . . .	49
4.2.4	Controlador Fuzzy Proporcional Integral . . . . .	49
4.3	Comportamento Limite em Controladores Fuzzy . . . . .	49
4.3.1	Interpretação Geométrica . . . . .	50
4.3.2	Teoremas Limites . . . . .	53
4.4	Controladores Fuzzy Lineares . . . . .	57
4.4.1	Números Fuzzy . . . . .	57
4.4.2	Regras . . . . .	58
4.4.3	Transformação Escalar-Fuzzy . . . . .	58
4.5	Estudo de Caso: Sistema Linear de Primeira-Ordem . . . . .	58
4.6	Conclusão . . . . .	69
<b>5</b>	<b>Não Linearidades em Controladores Fuzzy</b>	<b>71</b>
5.1	Introdução . . . . .	71
5.2	Fontes de Não-Linearidades em Controladores Fuzzy . . . . .	71
5.3	Estudo de Caso: Sistema de Primeira-Ordem com Tempo Morto . . . . .	72
5.4	Estudo de Caso: Pêndulo Invertido . . . . .	85
5.5	Conclusão . . . . .	99
<b>6</b>	<b>Análise de Controladores Fuzzy</b>	<b>100</b>
6.1	Introdução . . . . .	100
6.2	Funções Descritivas . . . . .	100
6.2.1	Funções Descritivas de Elementos Multilevel-Relay . . . . .	101
6.3	Critério de Estabilidade do Círculo . . . . .	104
6.4	Função de Lyapunov . . . . .	105
6.4.1	Critério de Estabilidade . . . . .	105
6.4.2	Método de Projeto . . . . .	107
6.5	Equações Relacionais . . . . .	107
6.6	Uma Abordagem Baseada em Energia . . . . .	110
6.7	Método de Mapeamento Célula-a-Célula . . . . .	113
6.7.1	Espaço de Estados Representados por Células . . . . .	113
6.7.2	Mapeamento Célula-a-Célula . . . . .	114
6.7.3	Algoritmo de Agrupamento . . . . .	116
6.7.4	Análise Global de Sistemas Dinâmicos Fuzzy . . . . .	117
6.8	Conclusão . . . . .	121
<b>7</b>	<b>Observações Finais</b>	<b>123</b>
<b>A</b>	<b>Programas de Simulação</b>	<b>125</b>
A.1	Introdução . . . . .	125
A.2	Simulador Básico . . . . .	125
A.2.1	Características Gerais . . . . .	125
A.2.2	Módulo Principal . . . . .	128
A.2.3	Módulo de Simulação . . . . .	133
A.2.4	Máquina de Inferência Fuzzy . . . . .	137

A.2.5 Algoritmos de Integração Numérica . . . . . 141

A.3 Simulador para o Método Simplificado . . . . . 141

A.4 Simulador para o Método de Buckley e Ying . . . . . 141

A.5 Simulador para o Pêndulo Invertido . . . . . 141

BIBLIOTECA DE CIÊNCIAS EXATAS  
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
CAMPUS MARACÁ



# Lista de Figuras

1.1	função de pertinência para <i>jovem</i> e <i>usualmente</i> . . . . .	11
1.2	definição clássica para o predicado <i>jovem</i> . . . . .	11
1.3	sistema de controle fuzzy . . . . .	12
1.4	algoritmo de raciocínio fuzzy numa base de conhecimento com duas regras. . . . .	15
1.5	Representação do mapeamento $W : E \times CE \rightarrow U$ através de um algoritmo de raciocínio fuzzy. . . . .	16
2.1	representação de $A \cup B$ e $A \cap B$ . . . . .	19
2.2	representação de um conjunto fuzzy $A$ e seu complemento $\bar{A}$ . . . . .	20
2.3	representação fuzzy de speed. . . . .	21
2.4	Interpretação gráfica com $\hat{a}$ e $R_c$ . . . . .	29
2.5	Interpretação gráfica com $\hat{a}$ e $R_p$ . . . . .	30
2.6	interpretação gráfica do raciocínio fuzzy utilizando a operação de mínimo como função de implicação. . . . .	31
2.7	interpretação gráfica do raciocínio fuzzy utilizando o produto algébrico como função de implicação. . . . .	31
2.8	representação gráfica do método de Tsukamoto. . . . .	32
2.9	método simplificado. . . . .	33
3.1	Controlador Fuzzy . . . . .	36
3.2	representação gráfica de partições fuzzy. . . . .	42
3.3	definição funcional . . . . .	44
4.1	Tabela de decisões para o controlador fuzzy PI de MacVicar-Whelan. . . . .	51
4.2	Tabela de decisões idealizada para um controlador PI linear, onde a linha $L_1$ (inclinação $-C$ ) representa a fronteira de controle e a linha $L_2$ (inclinação $1/C$ ) representa a faixa das ações de controle . . . . .	51
4.3	sistema de controle fuzzy. . . . .	53
4.4	Números fuzzy triangulares para $2N + 1 = 7$ . . . . .	54
4.5	estrutura de controle para sistema de primeira-ordem. . . . .	60
4.6	números fuzzy para as variáveis $e$ e $ce$ . . . . .	60
4.7	números fuzzy para a variável $cu$ . . . . .	60
4.8	resposta a degrau. . . . .	63
4.9	superfície de controle e curvas de nível para o algoritmo de Mandani. O intervalo $[0, 20]$ indicado nos eixos correspondentes às variáveis $e$ e $ce$ se refere ao intervalo $[-4.0, +4.0]$ . A escala referente ao eixo $cu$ corresponde a valores reais. . . . .	64

4.10	superfície de controle e curvas de nível para o método de Buckley e Ying. O intervalo $[0, 20]$ indicado nos eixos correspondentes às variáveis $e$ e $ce$ se refere ao intervalo $[-4, +4]$ . A escala referente ao eixo $cu$ corresponde a valores reais. . . . .	65
4.11	superfície de controle e curvas de nível para o método simplificado. O intervalo $[0, 20]$ indicado nos eixos correspondentes às variáveis $e$ e $ce$ se refere ao intervalo $[-4.0, +4.0]$ . A escala referente ao eixo $cu$ corresponde a valores reais. . . . .	66
4.12	resposta a degrau (Algoritmo de Mandani). . . . .	68
4.13	superfície de controle e curvas de nível para o Algoritmo de Mandani (225 regras). O intervalo $[0, 20]$ indicado nos eixos correspondentes às variáveis $e$ e $ce$ se refere ao intervalo $[-4.0, +4.0]$ . A escala referente ao eixo $cu$ corresponde a valores reais . . . . .	70
5.1	estrutura de controle para sistema de primeira ordem com atraso. . . . .	73
5.2	números fuzzy para as variáveis $e$ , $ce$ e $cu$ . . . . .	73
5.3	resposta dinâmica para degraus de 30%, 60% e 100%. . . . .	76
5.4	resposta a degrau $y$ e esforço de controle $u$ . . . . .	78
5.5	superfície de controle - $A_1$ . O intervalo $[0, 20]$ indicado nos eixos $e$ e $ce$ corresponde ao intervalo real $[-2.0, +2.0]$ . A escala referente ao eixo $cu$ corresponde aos valores reais. . . . .	79
5.6	superfície de controle - $A_2$ . O intervalo $[0, 20]$ indicado nos eixos $e$ e $ce$ corresponde ao intervalo real $[-2.0, +2.0]$ . A escala referente ao eixo $cu$ corresponde aos valores reais. . . . .	80
5.7	superfície de controle - $A_3$ . O intervalo $[0, 20]$ indicado nos eixos $e$ e $ce$ corresponde ao intervalo real $[-2.0, +2.0]$ . A escala referente ao eixo $cu$ corresponde aos valores reais. . . . .	81
5.8	superfície de controle - $A_4$ . O intervalo $[0, 20]$ indicado nos eixos $e$ e $ce$ corresponde ao intervalo real $[-2.0, +2.0]$ . A escala referente ao eixo $cu$ corresponde aos valores reais. . . . .	82
5.9	superfície de controle - $A_5$ . O intervalo $[0, 20]$ indicado nos eixos $e$ e $ce$ corresponde ao intervalo real $[-2.0, +2.0]$ . A escala referente ao eixo $cu$ corresponde aos valores reais. . . . .	83
5.10	superfície de controle - $A_6$ . O intervalo $[0, 20]$ indicado nos eixos $e$ e $ce$ corresponde ao intervalo real $[-2.0, +2.0]$ . A escala referente ao eixo $cu$ corresponde aos valores reais. . . . .	84
5.11	pêndulo invertido. . . . .	85
5.12	Superfície de restrição associada a $U_c$ . O intervalo $[0, 20]$ indicado no eixo $\theta$ corresponde ao intervalo real $[-2.0, +2.0]$ e no eixo $\dot{\theta}$ corresponde ao intervalo $[-15.0, +15.0]$ . A escala referente ao eixo $u$ corresponde a valores reais. . . . .	88
5.13	números fuzzy para as variáveis de estado. . . . .	88
5.14	números fuzzy para $f$ . . . . .	89



5.15	superfície de controle $U_f$ . O intervalo $[0, 20]$ indicado no eixo $\theta$ corresponde ao intervalo real $[-2.0, +2.0]$ e no eixo $\dot{\theta}$ corresponde ao intervalo $[-15.0, +15.0]$ . A escala referente ao eixo $u$ corresponde a valores reais. . . . .	91
5.16	estrutura para o controle da haste. . . . .	91
5.17	posição angular - $\theta$ . . . . .	92
5.18	velocidade angular - $\dot{\theta}$ . . . . .	92
5.19	força aplicada - $f$ . . . . .	93
5.20	VS para a variável $\theta$ . . . . .	94
5.21	VS para a variável $\dot{\theta}$ . . . . .	94
5.22	NE, PO para a variável $x$ . . . . .	94
5.23	NE, NVS, PVS, PO para a variável $\dot{x}$ . . . . .	95
5.24	estrutura de controle para o pêndulo + carro. . . . .	95
5.25	posição angular - $\theta$ . . . . .	96
5.26	velocidade angular - $\dot{\theta}$ . . . . .	97
5.27	posição do carro - $x$ . . . . .	97
5.28	velocidade do carro - $\dot{x}$ . . . . .	98
5.29	força aplicada - $f$ . . . . .	98
6.1	característica entrada-saída do controlador fuzzy. . . . .	102
6.2	elemento não linear. . . . .	103
6.3	sistema de controle não linear. . . . .	104
6.4	sistema com realimentação. . . . .	105
6.5	representação gráfica para o critério de estabilidade do círculo. . . . .	105
6.6	sistema dinâmico fuzzy. . . . .	108
6.7	estados estáveis e instáveis. . . . .	111
6.8	dois conjuntos fuzzy denominados <i>medium</i> . . . . .	112
6.9	trajetórias para o pêndulo invertido e o seu mapeamento célula-a-célula com estados iniciais $x_1 = 0.6$ e $x_2 = 3.0$ (quadrados para o mapeamento célula-a-célula e pontos para a trajetória real). A trajetória real converge para a origem enquanto que o mapeamento célula-a-célula converge para um movimento p-8 em torno da origem [23,24]. . . . .	118
6.10	domínios de atração de passo-20 para o movimento p-8. As áreas mais escuras se referem aos domínios de atração da célula externa [23,24]. . . . .	119
A.1	sistema de controle (controlador fuzzy + planta) . . . . .	126
A.2	módulos do Simulador Básico. . . . .	127

# Lista de Tabelas

3.1	Quantização . . . . .	41
3.2	Normalização e Conjuntos Fuzzy Primários Utilizando Definição Funcional	41
4.1	desempenho dinâmico do controlador PI. . . . .	59
4.2	normalização e funções de pertinência para as variáveis $e$ e $ce$ . Veja nota de rodapé na página 59. . . . .	61
4.3	normalização e funções de pertinência da variável $cu$ . Veja nota de rodapé na página 59. . . . .	61
4.4	tabela de decisões ( $e, ce$ ) $\rightarrow cu$ . . . . .	61
4.5	normalização e funções de pertinência para as variáveis $e$ e $ce$ . Veja nota de rodapé na página 59. . . . .	62
4.6	normalização e funções de pertinência para variável $cu$ . Veja nota de rodapé na página 59. . . . .	67
4.7	tabela de decisões. . . . .	67
5.1	normalização e funções de pertinência para as variáveis $e$ , $ce$ e $cu$ . Veja nota de rodapé na página 59. . . . .	73
5.2	tabela de decisões ( $e, ce \rightarrow cu$ ). . . . .	74
5.3	desempenho dinâmico com a variação de $K_{cu}$ . . . . .	75
5.4	desempenho dinâmico com a variação de $K_e$ . . . . .	75
5.5	desempenho dinâmico com a variação de $K_{ce}$ . . . . .	75
5.6	desempenho dinâmico com a variação de $K_{ce}$ . . . . .	75
5.7	Algoritmos de Raciocínio Fuzzy. . . . .	77
5.8	normalização e funções de pertinência para a variável $\theta$ . Veja nota de rodapé na página 59. . . . .	89
5.9	normalização e funções de pertinência para a variável $\hat{\theta}$ . Veja nota de rodapé na página 59. . . . .	89
5.10	normalização e funções de pertinência para a variável $f$ . Veja nota de rodapé na página 59. . . . .	90
6.1	número de células dos grupos em diferentes conjuntos de células mínimo. .	121
6.2	Proporções de células dos grupos em diferentes conjuntos de células mínimo.	121

# Tabela de Símbolos

$\mu_F$  função de pertinência do conjunto fuzzy  $F$

$F = \int_U \mu_F(u)/u$  conjunto fuzzy  $F$  definido sobre o universo de discurso contínuo  $U$

$F = \sum_{i=1}^n \mu_F(u)/u_i$  conjunto fuzzy  $F$  definido sobre o universo de discurso discreto  $U$

$\circ$  operador composicional sup-star

max máximo

min mínimo

\* norma triangular

$\wedge$  intersecção

$\cdot$  produto algébrico

$\odot$  produto limitado

$\hat{\cap}$  produto drástico

$\ddagger$  co-norma triangular

$\vee$  união

$\hat{+}$  soma algébrica

$\oplus$  soma limitada

$\dot{\cup}$  soma drástica

$\triangle$  soma disjunta

$\rightarrow$  função de implicação fuzzy

not $A$  complemento de  $A$

sup supremo

inf infimo

$R_c$  regra do mínimo para função de implicação fuzzy

$R_p$  regra do produto para função de implicação fuzzy

- $R_{lp}$  regra do produto limitado para função de implicação fuzzy
- $R_{dp}$  regra do produto drástico para função de implicação fuzzy
- $R_a$  regra aritmética para função de implicação fuzzy
- $R_m$  regra Max-Min para função de implicação fuzzy
- $R_s$  seqüência padrão
- $R_b$  implicação fuzzy booleana
- $R_{\Delta}$  implicação fuzzy de Goguen
- $\cup$  união
- $\cap$  intersecção
- MISO sistema de múltiplas entradas e única saída
- MIMO sistema de múltiplas entradas e múltiplas saídas



# Capítulo 1

## Introdução

### 1.1 Lógica Fuzzy

A palavra *fuzzy*, da língua inglesa, possui o significado de: algo não perfeitamente delimitado; não claramente expresso; confuso. Tradicionalmente utiliza-se na língua inglesa a palavra *crisp* como antônimo de *fuzzy*. Dentro da literatura técnica nacional o termo *fuzzy* tem sido traduzido como *difuso* ou *nebuloso*. Nesse texto, optou-se por utilizar o termo original na língua inglesa (lógica fuzzy, conjunto fuzzy, etc.)<sup>1</sup>. O termo *fuzzy* foi inicialmente utilizado por Lotfi A. Zadeh em 1965 no seu artigo "Fuzzy Sets" [117] onde introduz as propriedades básicas de conjuntos fuzzy.

Lógica fuzzy, como o nome sugere, está relacionada aos princípios formais de raciocínio aproximado. A importância da lógica fuzzy vem do fato que a maioria dos modos de raciocínio humano está relacionado a inferências de respostas aproximadas para questões baseadas em bases de conhecimento que são inexatas, incompletas ou não totalmente confiáveis.

Por exemplo, através de lógica fuzzy pode-se manipular proposições do seguinte tipo:

1. *Usualmente* leva-se *em torno de uma hora* para se ir de carro de São Paulo a Campinas,
2. Os estudantes são *muito jovens*,
3. Se o preço é *alto* então o lucro será *grande*,
4. A *maioria* dos especialistas acredita que a probabilidade de um terremoto é *muito pequena*.

Nas proposições acima *usualmente* e *maioria* são quantificadores fuzzy; *em torno de uma hora*, *jovens*, *alto*, *grande* são predicados fuzzy; *muito* é um modificador de predicado e *muito pequena* é uma probabilidade fuzzy. Quantificadores, predicados, modificadores de predicado e probabilidades fuzzy podem ser representados através de *conjuntos fuzzy*.

---

<sup>1</sup>Alguns termos como *fuzzification* e *defuzzification* foram adaptados para transformação escalar-fuzzy e transformação fuzzy-escalar.

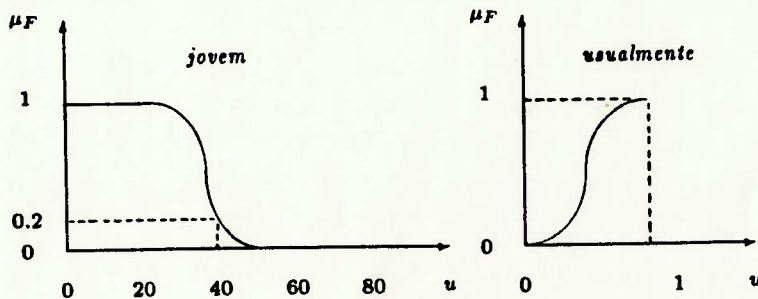


Figura 1.1: função de pertinência para *jovem* e *usualmente*

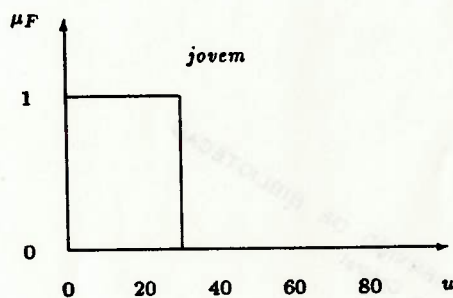


Figura 1.2: definição clássica para o predicado *jovem*

Um conjunto fuzzy  $F$  num *universo de discurso*  $U$  é caracterizado por uma função de pertinência  $\mu_F : U \rightarrow [0, 1]$ , que associa a cada elemento  $u$  de  $U$  um número  $\mu_F(u)$  que representa o grau de pertinência de  $u$  no conjunto  $F$ . A figura 1.1 mostra uma possível definição para o predicado fuzzy *jovem* (onde  $u$  representa a idade de uma pessoa) e para o quantificador fuzzy *usualmente* (onde  $u$  representa a frequência de ocorrência de um evento). Em contradição, um conjunto ordinário é definido através de uma função de pertinência  $\mu_F : U \rightarrow \{0, 1\}$ . A figura 1.2 ilustra uma possível definição para o predicado *jovem* através de um conjunto ordinário (*crisp set*).

Sob o ponto de vista clássico, a idade de uma pessoa  $u$  pode pertencer ao conjunto *jovem* ( $\mu_F(u) = 1$ ) ou não pertencer ao conjunto *jovem* ( $\mu_F(u) = 0$ ). Por exemplo, na figura 1.2 temos  $\mu_F(40) = 0$ . Dentro do domínio fuzzy, a idade de uma pessoa pertence ao conjunto fuzzy *jovem* de acordo com um grau de pertinência que varia de 0 a 1. Por exemplo, na figura 1.1 temos  $\mu_F(40) = 0.2$ , ou seja, a idade 40 pertence ao conjunto *jovem* com um grau de pertinência 0.2.

O grande poder de expressão da lógica fuzzy deriva do fato de que ela contém como casos especiais não apenas sistemas lógicos clássicos e multivalores, mas também teoria



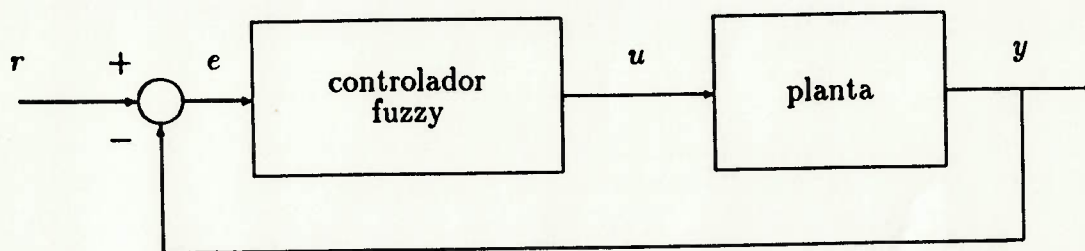


Figura 1.3: sistema de controle fuzzy

da probabilidade e lógica probabilística. No transcorrer desse texto será apresentado apenas conceitos de conjuntos fuzzy e lógica fuzzy relevantes para o entendimento de controladores fuzzy. Uma análise mais detalhada sobre lógica fuzzy e seus mecanismos de inferência pode ser encontrada em [127,128,126,125].

## 1.2 Controle Fuzzy

A construção de um sistema de controle fuzzy é baseado na idéia de se incorporar “experiência” ou “conhecimento especialista” de um operador humano para se obter a melhor estratégia de controle. Um sistema de controle fuzzy é similar a um sistema especialista no sentido de que ambos modelam o conhecimento e o comportamento humano de tomada de decisões. Entretanto, sistemas especialistas são em geral apenas ferramentas de consulta e não mecanismos de tempo real. Os sistemas de controle fuzzy se apóiam na teoria de conjuntos fuzzy e lógica fuzzy para a construção de bases de conhecimento e para a realização de inferências.

A idéia original de controle fuzzy foi inicialmente proposta por Zadeh em [118, 119,120] como uma das possíveis aplicações da teoria de conjuntos fuzzy. O primeiro controlador fuzzy foi implementado por Mandani em 1974 [62]. A primeira aplicação industrial de controle fuzzy foi realizado por Ostergaard em 1977 [72] ao projetar um controlador para um trocador de calor. O sistema de operação automática de trens projetado por Yasunobu e Myamoto em 1985 [113] foi a primeira aplicação de grande impacto de controladores fuzzy. Esse projeto foi o primeiro a utilizar um controlador fuzzy preditivo que permite a escolha do esforço de controle otimizando-se vários parâmetros simultaneamente.

Controladores fuzzy têm sido aplicados a uma grande variedade de problemas desde as primeiras experiências realizadas por Mandani. Como exemplo temos: fornos de cimento [102], plantas nucleares [15], veículos autônomos [88,61,91,93], manipuladores [73,96,63,40], guindastes [114], máquina diesel [69], pêndulo invertido [112,23], refinarias [11], processos biológicos [29] e tratamento de água [99].

A configuração básica de um sistema de controle fuzzy é mostrado na figura 1.3.

Uma planta e um controlador fuzzy constituem a malha de controle.

O controlador fuzzy pode ser representado por uma equação do seguinte tipo:

$$u = W(e), \quad (1.1)$$

onde  $e$  representa o erro do processo e  $u$  representa o esforço de controle.

A planta pode ser representada através da seguinte equação:

$$y = G(u), \quad (1.2)$$

onde  $y$  representa a saída do processo.

O problema fundamental de todo sistema de controle é determinar  $W(e)$  de tal forma que o comportamento global do sistema em malha fechada dado por,

$$y = H(r), \quad (1.3)$$

onde  $r$  representa o sinal de referência, corresponda a um conjunto de especificações de projeto. As especificações relacionadas ao projeto de controladores genéricos podem ser basicamente divididas em três categorias [16]:

1. *especificações de desempenho*: descrevem como deve ser o desempenho do sistema em malha fechada. Por exemplo: o sistema em malha fechada deve ser estável, os efeitos de distúrbios ou ruídos no sistema devem ser pequenos, o sistema deve responder de uma maneira particular aos sinais de referência, etc.
2. *especificações de desempenho robusto*: descrevem como deve ser o desempenho do sistema em malha fechada se alguma das partes do sistema mudar ou for perturbada. Por exemplo, a sensibilidade a alguma variação de parâmetro deve ser pequena, o sistema em malha fechada deve permanecer estável frente a perturbações específicas, etc.
3. *especificações da lei de controle*: descrevem propriedades da própria lei de controle. Por exemplo: a lei de controle deve ser descrita por uma equação diferencial de baixa ordem, o controlador deve ser estável em malha aberta, etc.

Para realizar o projeto do controlador utilizando alguma técnica, é conveniente representar o sistema global através de alguma ferramenta matemática. Os sistemas lineares invariantes no tempo, por exemplo, podem ser representados através de funções de transferência ou equações diferenciais ordinárias. É possível dessa forma utilizar-se de métodos analíticos (como LQG,  $H_\infty$ , LTR, etc.) de forma a atender às especificações de projeto.

Para um controlador fuzzy a função  $u = W(e)$  não pode ser representada através de equações diferenciais lineares ou não lineares. O operador  $W$  está relacionado à execução de um algoritmo que realiza um mapeamento do espaço das variáveis de estado<sup>2</sup> no espaço das variáveis de controle a partir de uma base de conhecimento que representa uma determinada estratégia de controle.

<sup>2</sup>em geral nos controladores fuzzy as variáveis dos termos antecedentes são denominadas variáveis de estado

Vamos supor que um determinado controlador fuzzy possua como variáveis de estado o erro ( $e$ ) do processo e a variação do erro ( $ce$ ), e como variável de saída o esforço de controle ( $u$ ). Na figura 1.4 temos a representação gráfica de um processo de raciocínio fuzzy que envolve uma base de conhecimento com apenas duas regras. Vamos a seguir tentar interpretar tal algoritmo sob o ponto de vista da arquitetura do controlador.

Um controlador fuzzy possui quatro componentes principais. O comportamento de  $W$  é determinado por cada um desses componentes:

1. *interface escalar-fuzzy*: realiza a transformação de um valor escalar em um conjunto fuzzy. No caso da figura 1.4 os valores escalares de entrada ( $e_0, ce_0$ ) são utilizados diretamente. Um valor escalar equivale a um conjunto fuzzy unitário denominado "singleton fuzzy" como poderá ser visto no capítulo 2.

2. *base de conhecimento*:

(a) *base de regras*: no caso da figura 1.4 é composta por duas regras,

$$R_1: \text{ se } e = A_1 \text{ e } ce = B_1 \text{ então } u = C_1$$

$$R_2: \text{ se } e = A_2 \text{ e } ce = B_2 \text{ então } u = C_2$$

A escolha das variáveis de estado e das variáveis de controle corresponde ao que denominamos na teoria de identificação de sistemas de identificação estrutural.

(b) *base de dados*: corresponde às definições quantitativas (definição das funções de pertinência) dos conjuntos fuzzy, isto é, no caso da figura 1.4 as definições de  $A_1, B_1, C_1, A_2, B_2$  e  $C_2$ . A definição quantitativa dos conjuntos fuzzy corresponde à identificação de parâmetros do controlador.

3. *lógica de tomada de decisões*: é o núcleo de um controlador fuzzy, infere as ações de controle através do emprego de implicações fuzzy e regras de inferência fuzzy. O mecanismo de inferência apresentado na figura 1.4 corresponde ao algoritmo de raciocínio fuzzy de Mandani e corresponde a seguinte decisão de controle:

$$\alpha_1 = \mu_{A_1}(e_0) \wedge \mu_{B_1}(ce_0), \quad (1.4)$$

$$\alpha_2 = \mu_{A_2}(e_0) \wedge \mu_{B_2}(ce_0), \quad (1.5)$$

$$\mu_C(u) = \mu_{C_1}(u) \vee \mu_{C_2}(u), \quad (1.6)$$

$$= [\alpha_1 \wedge \mu_{C_1}(u)] \vee [\alpha_2 \wedge \mu_{C_2}(u)], \quad (1.7)$$

onde  $\vee = \max$  e  $\wedge = \min$ .

4. *interface fuzzy-escalar*: transforma uma ação de controle fuzzy numa ação de controle escalar. O conjunto fuzzy  $C$  deve ser transformado num valor escalar  $u_0$  para

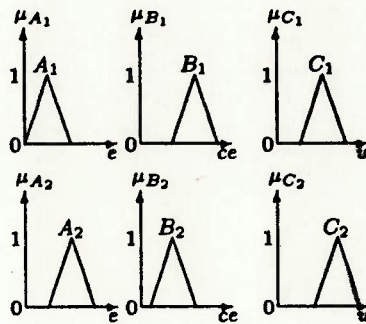


valores de entrada:  $\begin{cases} e_0 \\ ce_0 \end{cases}$

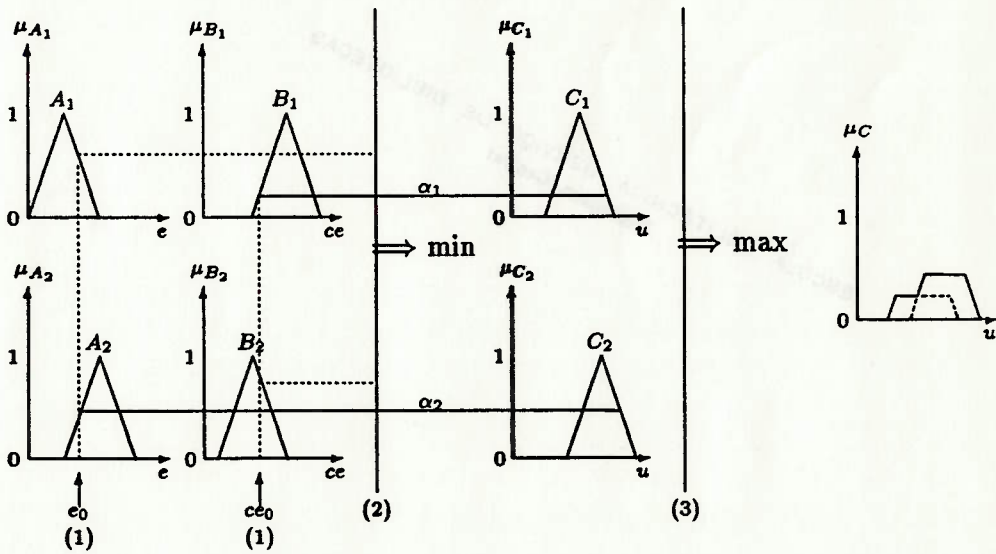
(a) interface escalar-fuzzy

$\begin{cases} R_1 : \text{se } e = A_1 \text{ e } ce = B_1 \text{ então } u = C_1 \\ R_2 : \text{se } e = A_2 \text{ e } ce = B_2 \text{ então } u = C_2 \end{cases}$

(b) base de regras

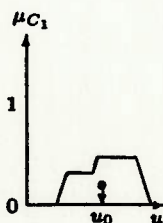


(c) base de dados



$\begin{cases} (1) \text{ entrada dos valores } e_0, ce_0 \\ (2) \text{ cálculo de } \alpha_1 \text{ e } \alpha_2 \\ (3) \text{ cálculo de } \mu_C(u) \end{cases}$

(d) lógica de tomada de decisões



(e) interface fuzzy-escalar

Figura 1.4: algoritmo de raciocínio fuzzy numa base de conhecimento com duas regras.

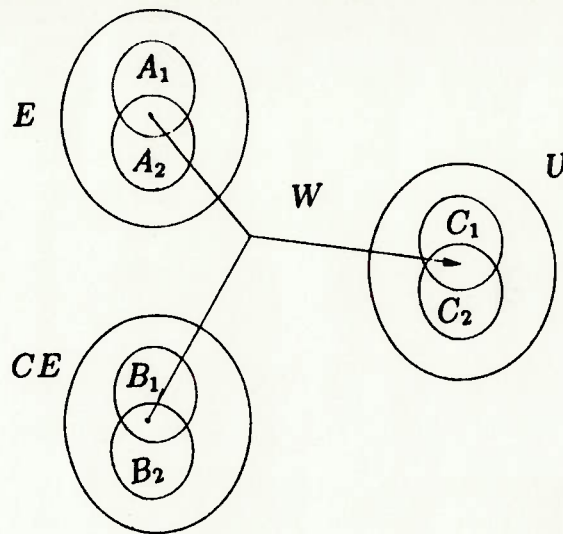


Figura 1.5: Representação do mapeamento  $W : E \times CE \rightarrow U$  através de um algoritmo de raciocínio fuzzy.

ser utilizado como ação de controle, em geral, adota-se o centro de gravidade do conjunto  $C$  como ação de controle,

$$u_0 = \frac{\int u \mu_C(u) du}{\int \mu_C(u) du}. \quad (1.8)$$

Como pôde ser observado, o algoritmo acima representa um mapeamento do tipo  $W : E \times CE \rightarrow U$ , onde  $E$ ,  $CE$  e  $U$  representam respectivamente os domínios das variáveis  $e$ ,  $ce$  e  $u$ . Esse mapeamento pode ser ilustrado através da figura 1.5.

Podemos afirmar com certeza que o aspecto dominante do estado atual da técnica de projeto de controladores fuzzy é a construção da base de conhecimento. Dessa forma a satisfação das especificações de projeto acima citadas deve ser obtida através da experiência ou conhecimento especialista de um operador humano. Os outros componentes podem ser considerados de menor importância desde que escolhidos dentro de um determinado conjunto de opções.

### 1.3 Motivações do Trabalho

Nos últimos anos, pudemos assistir a saída da tecnologia de sistemas fuzzy do ambiente acadêmico para as aplicações industriais e mais recentemente a incorporação de tal tecnologia em produtos de consumo. Notadamente, o Japão tem realizado os maiores avanços nessa área, onde sistemas fuzzy controlam o metrô; focalizam e estabilizam câmeras de vídeo; ajustam ar condicionados, máquinas de lavar roupa e aspiradores de pó; e controlam suspensões ativas e caixas de câmbio automáticas em automóveis.

Muito embora o número de aplicações de lógica fuzzy esteja aumentando a cada ano,

nota-se ainda que avanços teóricos devem ser realizados no sentido de se obter métodos sistemáticos de projeto e de análise dos controladores fuzzy.

O objetivo desse trabalho é organizar alguns dos resultados relevantes de controle fuzzy já obtidos com relação ao projeto e análise de controladores.

Como já foi dito, controladores fuzzy representam *conhecimento especialista* ou *conhecimento estruturado* [54]. O fato dessa técnica estar baseada em conhecimento especialista representa ao mesmo tempo o seu ponto forte e o seu ponto fraco. Muitas vezes a ausência de conhecimento especialista inviabiliza a sua utilização.

O grande desafio é o estabelecimento de uma teoria de *sistemas dinâmicos fuzzy* que permita o projeto de controladores e a análise global do comportamento do sistema semelhante à teoria de sistemas lineares.

Os resultados apresentados nesse trabalho permitem determinar algumas diretrizes gerais para o projeto de controladores fuzzy relacionados principalmente a construção da base de conhecimento e a adoção de mecanismos de inferência adequados. Algumas técnicas de análise voltadas à determinação de estabilidade e robustez de sistemas de controle fuzzy também são apresentadas.

O capítulo 2 trata sobre os conceitos básicos da teoria de conjuntos e lógica fuzzy relevantes para o desenvolvimento de controladores fuzzy. Mecanismos de inferência fuzzy são apresentados. Em particular, realiza-se a dedução formal dos algoritmos de Mandani e de Larsen.

O capítulo 3 apresenta com detalhes a arquitetura de um controlador fuzzy. Uma discussão a respeito dos parâmetros de projeto para controladores fuzzy é realizada.

O capítulo 4 estabelece uma relação entre controladores fuzzy e controladores lineares clássicos do tipo PID (P,PI,PD).

Controladores fuzzy possuem em geral um comportamento não linear. Mas, se uma base de regras adequada for construída, pode ser provado que a medida que o número de regras cresce, o comportamento do controlador fuzzy torna-se cada vez mais linear. São apresentadas algumas simulações com controladores fuzzy construídos de forma a obter um comportamento próximo de um controlador PI linear.

O capítulo 5 explora o controlador fuzzy como elemento não linear. São enunciadas as partes principais que compõem um controlador fuzzy e sua influência sobre as não linearidades globais obtidas. O projeto de um controlador fuzzy para um sistema de primeira-ordem com tempo morto e o projeto de um controlador para um pêndulo invertido utilizando uma função de Lyapunov são apresentados.

O capítulo 6 trata sobre as técnicas de análise de controladores fuzzy. Muito embora a técnica de projeto de controladores fuzzy seja bastante simples de ser aplicada, a análise de tais controladores sob o ponto de vista de conceitos como estabilidade e robustez é bastante difícil, em grande parte devido a sua característica linguística e não formal. São apresentadas as seguintes técnicas: *Funções Descritivas*, *Critério de Estabilidade do Círculo*, *funções de Lyapunov*, *Equações Relacionais Fuzzy* e *Método de Mapeamento Célula-a-Célula*.

O capítulo 7 apresenta os comentários finais e as sugestões para trabalhos futuros.

O apêndice A descreve os programas de simulação implementados.



## Capítulo 2

# Conjuntos e Lógica Fuzzy

### 2.1 Introdução

Esse capítulo trata sobre os conceitos básicos da teoria de conjuntos e lógica fuzzy relevantes para o desenvolvimento de controladores fuzzy. Em particular, são apresentados os mecanismos de inferência empregados em controladores fuzzy. Tais mecanismos de inferência são em geral mais simples do que os utilizados em sistemas especialistas pelo fato da base de regras de controladores fuzzy não apresentarem encadeamento de regras. Uma análise mais detalhada sobre lógica fuzzy e sua utilização em sistemas especialistas pode ser encontrada em [124,125,126,127,128].

### 2.2 Definições e Terminologia

Seja  $U$  uma coleção de objetos denotada genericamente por  $\{u\}$ , a qual pode ser discreta (número finito de elementos) ou contínua.  $U$  é denominado de *Universo de Discurso* e  $u$  representa um elemento genérico de  $U$ .

**Definição 2.1 (CONJUNTO FUZZY)** *Um conjunto fuzzy  $F$  num universo de discurso  $U$  é caracterizado por uma função de pertinência  $\mu_F : U \rightarrow [0, 1]$  que associa a cada elemento  $u$  de  $U$  um número  $\mu_F(u)$  no intervalo  $[0, 1]$  que representa o grau de pertinência de  $u$  em  $F$ . Um conjunto fuzzy pode ser visto como uma generalização do conceito de um conjunto ordinário cuja função de pertinência assume apenas dois valores  $\{0, 1\}$ . Portanto um conjunto fuzzy  $F$  em  $U$  pode ser representado por um conjunto de pares ordenados de um elemento genérico  $u$  e o seu grau de pertinência:  $F = \{(u, \mu_F(u)) \mid u \in U\}$ . Quando  $U$  é contínuo, um conjunto fuzzy pode ser escrito como,*

$$F = \int_U \mu_F(u)/u, \quad (2.1)$$

onde o símbolo de integral,  $\int_U$ , representa a união dos pares  $(u, \mu_F(u))$  representados por  $\mu_F(u)/u$ . Quando  $U$  é discreto, um conjunto fuzzy  $F$  é representado por,

$$F = \sum_{i=1}^n \mu_F(u_i)/u_i, \quad (2.2)$$

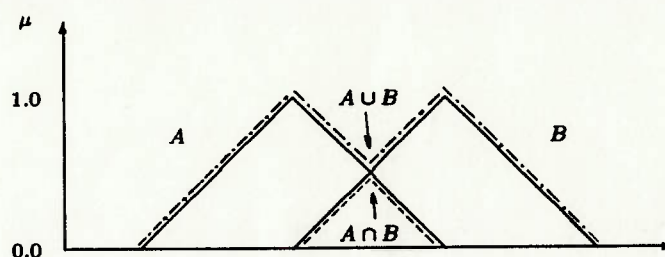


Figura 2.1: representação de  $A \cup B$  e  $A \cap B$

onde o símbolo de somatória,  $\Sigma$ , também representa a operação de união.

**Definição 2.2 (SUPORTE, PONTO DE “CROSSOVER” e “FUZZY SINGLETON”)** O suporte de um conjunto fuzzy  $F$  é o conjunto de todos os pontos  $u$  em  $U$  tal que  $\mu_F(u) > 0$ . Em particular, o elemento  $u$  em  $U$  tal que  $\mu_F = 0.5$ , é chamado de Ponto de “Crossover” e um conjunto fuzzy cujo suporte é um único ponto em  $U$  com  $\mu_F = 1.0$  é denominado “Fuzzy Singleton”.

### 2.3 Operações sobre Conjuntos Fuzzy

Seja  $A$  e  $B$  dois conjuntos fuzzy no universo de discurso  $U$  com funções de pertinência  $\mu_A$  e  $\mu_B$ , respectivamente. As operações de união, intersecção, e complemento são definidas através das funções de pertinência dos conjuntos, como mostrado a seguir.

**Definição 2.3 (UNIÃO)** A função de pertinência  $\mu_{A \cup B}$  da união  $A \cup B$  é definida para todo  $u \in U$  como (figura 2.1),

$$\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u)). \quad (2.3)$$

**Definição 2.4 (INTERSECÇÃO)** A função de pertinência  $\mu_{A \cap B}$  da intersecção  $A \cap B$  é definida para todo  $u \in U$  como (figura 2.1),

$$\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u)). \quad (2.4)$$

**Definição 2.5 (COMPLEMENTO)** A função de pertinência  $\mu_{\bar{A}}$  do complemento de um conjunto fuzzy  $A$  é definido para todo  $u \in U$  como (figura 2.2),

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u). \quad (2.5)$$

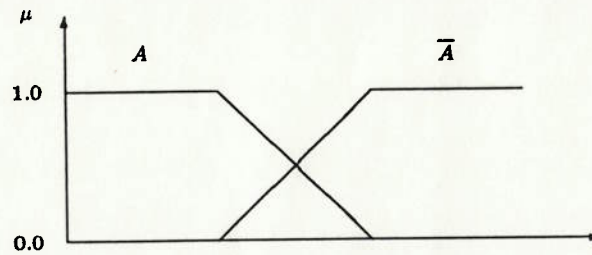


Figura 2.2: representação de um conjunto fuzzy  $A$  e seu complemento  $\bar{A}$ .

**Definição 2.6 (PRODUTO CARTESIANO)** Se  $A_1, \dots, A_n$  são conjuntos fuzzy em  $U_1, \dots, U_n$ , respectivamente, o produto cartesiano de  $A_1, \dots, A_n$  é um conjunto fuzzy no espaço  $U_1 \times \dots \times U_n$  com uma função de pertinência igual a,

$$\mu_{A_1 \times \dots \times A_n}(u_1, \dots, u_n) = \min\{\mu_{A_1}(u_1), \dots, \mu_{A_n}(u_n)\}, \quad (2.6)$$

ou

$$\mu_{A_1 \times \dots \times A_n}(u_1, \dots, u_n) = \mu_{A_1}(u_1) \cdot \dots \cdot \mu_{A_n}(u_n), \quad (2.7)$$

onde  $(\cdot)$  representa a operação de produto algébrico.

**Definição 2.7 (RELAÇÃO FUZZY)** Uma relação fuzzy de aridade  $n$  é um conjunto fuzzy em  $U_1 \times \dots \times U_n$  e é expresso como,

$$R_{U_1 \times \dots \times U_n} = \{(u_1, \dots, u_n), \mu_r(u_1, \dots, u_n) \mid (u_1, \dots, u_n) \in U_1 \times \dots \times U_n\}. \quad (2.8)$$

**Definição 2.8 (COMPOSIÇÃO SUP-STAR)** Se  $R$  e  $S$  são relações fuzzy  $U \times V$  e  $V \times W$ , respectivamente, a composição de  $R$  e  $S$  é uma relação fuzzy denotada por,  $R \circ S$  e é definida através de,

$$R \circ S = \left\{ \left[ (u, w), \sup_v (\mu_R(u, v) * \mu_S(v, w)) \right], u \in U, v \in V, w \in W \right\}, \quad (2.9)$$

onde  $*$  pode ser qualquer operador da classe de normas triangulares (veja definição 2.13).

**Definição 2.9 (nível- $\alpha$ )** Define-se o conjunto nível- $\alpha$  ou corte- $\alpha$  de  $A$  como,

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\}. \quad (2.10)$$

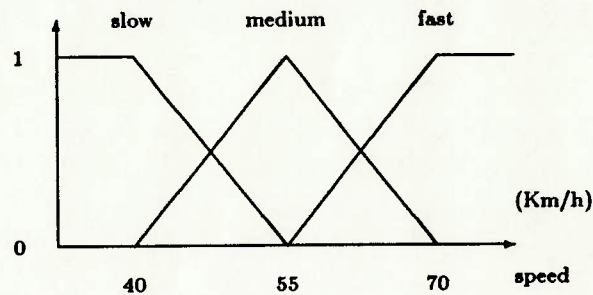


Figura 2.3: representação fuzzy de speed.

## 2.4 Variáveis Linguísticas e Conjuntos Fuzzy

**Definição 2.10 (NÚMERO FUZZY)** *Um número fuzzy  $F$  num universo contínuo  $U$  é um conjunto fuzzy  $F$  em  $U$  normal e convexo, isto é,*

$$\begin{aligned} \max_{u \in U} \mu_F(u) &= 1, \\ \mu_F(\lambda u_1 + (1 - \lambda)u_2) &\geq \min(\mu_F(u_1), \mu_F(u_2)), \\ u_1, u_2 \in U, \lambda &\in [0, 1] \end{aligned}$$

**Definição 2.11 (VARIÁVEIS LINGUÍSTICAS)** *Uma variável linguística é caracterizado por uma quintupla  $(x, T(x), U, G, M)$ , onde  $x$  é o nome de uma variável;  $T(x)$  é o conjunto de termos de  $x$ , ou seja, o conjunto dos nomes dos valores linguísticos de  $x$  e onde cada valor valor é interpretado como um número fuzzy definido em  $U$ ;  $G$  é a regra sintática para geração dos nomes dos valores de  $x$ ; e  $M$  é uma regra semântica que associa a cada valor um significado. Por exemplo, se speed é interpretado como variável linguística, então o seu conjunto de termos  $T(\text{speed})$  poderia ser:*

$$T(\text{speed}) = \{\text{slow, medium, fast, very slow, ...}\}$$

onde cada termo de  $T(\text{speed})$  é caracterizado por um conjunto fuzzy no universo de discurso  $U = [0, 100]$ . Pode-se interpretar slow como uma velocidade abaixo de 40 km/h, medium como uma velocidade em torno de 55 km/h, e fast como uma velocidade acima de 70 km/h. Esses termos podem ser caracterizados como conjuntos fuzzy com funções de pertinência como mostrado na figura 2.3. Variáveis linguísticas são extensivamente discutidas por Zadeh em [121, 122, 123].



**Definição 2.16 (DISJUNÇÃO FUZZY)** *uma disjunção fuzzy é definida para todo  $u \in U$  e  $v \in V$  através de*

$$A \rightarrow B = A \times B = \int_{U \times V} \mu_A(u) \dot{+} \mu_B(v) / (u, v), \quad (2.13)$$

onde  $\dot{+}$  é um operador que representa uma co-norma triangular.

**Definição 2.17 (IMPLICAÇÃO FUZZY)** *A implicação fuzzy está associada com cinco famílias de funções de implicação fuzzy em uso:*

1. *Implicação Material:*  $A \rightarrow B = (\text{not} A) \dot{+} B$
2. *Cálculo Proposicional:*  $A \rightarrow B = (\text{not} A) \dot{+} (A * B)$
3. *Cálculo Proposicional Extendido:*  $A \rightarrow B = (\text{not} A \times \text{not} B) \dot{+} B$
4. *Generalização de Modus Ponens:*  $A \rightarrow B = \sup\{c \in [0, 1], A * c \leq B\}$
5. *Generalização de Modus Tollens:*  $A \rightarrow B = \inf\{t \in [0, 1], B \dot{+} t \leq A\}$

Baseado nessas definições muitas funções de implicação fuzzy podem ser geradas a partir das normas triangulares e co-normas triangulares. Por exemplo, utilizando-se a definição de conjunção fuzzy, a implicação de Mandani,  $R_c$  é obtida se o operador de intersecção é utilizado. Se o produto algébrico é utilizado, obtém-se a implicação fuzzy proposta por Larsen,  $R_p$ . As implicações fuzzy mais correntes estão definidas abaixo:

1. Regra do mínimo para implicação fuzzy (Mandani):

$$R_c = A \times B = \int_{U \times V} \mu_A(u) \wedge \mu_B(v) / (u, v). \quad (2.14)$$

2. Regra do produto para implicação fuzzy (Larsen):

$$R_p = A \times B = \int_{U \times V} \mu_A(u) \mu_B(v) / (u, v). \quad (2.15)$$

3. Produto Limitado:

$$R_{bp} = A \times B = \int_{U \times V} \mu_A(u) \odot \mu_B(v) / (u, v). \quad (2.16)$$

4. Produto Drástico:

$$R_{dp} = A \times B = \int_{U \times V} \mu_A(u) \dot{\wedge} \mu_B(v) / (u, v). \quad (2.17)$$

5. Regra aritmética para implicação fuzzy (Zadeh):

$$R_a = (\text{not} A \times V) \oplus (U \times B) = \int_{U \times V} 1 \wedge (1 - \mu_A(u) + \mu_B(v)) / (u, v). \quad (2.18)$$

## 2.5 Lógica Fuzzy e Raciocínio Aproximado

Em lógica fuzzy e em raciocínio aproximado, existem duas importantes regras de inferência denominadas Modus Ponens Generalizado (GMP),

$$\begin{array}{l} \text{premissa 1: } x = A' \\ \text{premissa 2: } \text{se } x = A \text{ então } y = B \\ \hline \text{conseqüência: } y = B' \end{array}$$

e Modus Tollens Generalizado (GMT):

$$\begin{array}{l} \text{premissa 1: } y = B' \\ \text{premissa 2: } \text{se } x = A \text{ então } y = B \\ \hline \text{conseqüência: } x = A' \end{array}$$

A inferência fuzzy está baseada na regra de inferência composicional para raciocínio aproximado sugerida por Zadeh em 1973 [120]. Os conjuntos fuzzy  $A, A', B, B'$  são utilizados como valores linguísticos para as variáveis  $x, y$  ao invés de conjuntos precisos como em lógica clássica.

A regra de inferência Modus Ponens Generalizado, que se reduz a Modus Ponens quando  $A' = A$  e  $B' = B$ , é um processo de inferência direto (*forward data driven inference*) que é particularmente útil para controladores fuzzy. A regra de inferência Modus Tollens Generalizado é um processo de inferência reverso (*backward goal driven inference*) é comumente utilizado em sistemas especialistas, por exemplo para diagnose médica.

**Definição 2.12 (REGRA DE INFERÊNCIA COMPOSICIONAL Sup-Star)** Se  $R$  é uma relação fuzzy em  $U \times V$ , e  $x$  é um conjunto fuzzy em  $U$ , então a regra de inferência sup-star define que o conjunto fuzzy  $y$  em  $V$  induzido por  $x$  é dado por,

$$y = x \circ R, \quad (2.11)$$

onde  $x \circ R$  é a composição sup-star de  $x$  e  $R$ . Se star representa o operador mínimo então essa definição se reduz a regra de inferência composicional de Zadeh [120].

## 2.6 Funções de Implicação Fuzzy

Em geral, uma regra de controle fuzzy é uma relação fuzzy que é expressa como uma implicação fuzzy. A definição de uma implicação fuzzy pode ser expressa através de uma função de implicação fuzzy.

A escolha de uma função de implicação fuzzy envolvem critérios [57] que não serão discutidos aqui.

Seguindo a idéia de Zadeh, vários pesquisadores propuseram funções de implicação fuzzy no qual os antecedentes e consequentes contêm variáveis fuzzy. Na literatura podem ser encontradas atualmente cerca de 40 funções de implicação fuzzy. Em geral essas implicações podem ser classificadas em três categorias básicas: *conjunção fuzzy*, *disjunção*



fuzzy e implicação fuzzy. A seguir serão discutidos alguns tipos de funções de implicação fuzzy.

**Definição 2.13 (NORMAS TRIANGULARES)** *Uma norma triangular, representada genericamente por  $*$ , é uma função  $*$  :  $[0, 1] \times [0, 1] \rightarrow [0, 1]$ , inclui as operações de intersecção, produto algébrico, produto limitado e produto drástico. As operações associadas são definidas para todo  $x, y \in [0, 1]$ :*

intersecção	$x \wedge y = \min\{x, y\}$
produto algébrico	$x \cdot y = xy$
produto limitado	$x \odot y = \max\{0, x + y - 1\}$
produto drástico	$x \dot{\wedge} y = \begin{cases} x & y = 1 \\ y & x = 1 \\ 0 & x, y < 1. \end{cases}$

**Definição 2.14 (CO-NORMAS TRIANGULARES)** *Uma co-norma triangular, representada genericamente por  $\dot{+}$ , é uma função  $\dot{+}$  :  $[0, 1] \times [0, 1] \rightarrow [0, 1]$ , inclui as operações de união, soma algébrica, soma limitada, soma drástica, e soma disjunta. As operações associadas são definidas para todo  $x, y \in [0, 1]$ :*

união	$x \vee y = \max\{x, y\}$
soma algébrica	$x \dot{+} y = x + y - xy$
soma limitada	$x \oplus y = \min\{1, x + y\}$
soma drástica	$x \dot{\cup} y = \begin{cases} x & y = 0 \\ y & x = 0 \\ 1 & x, y > 0. \end{cases}$
soma disjunta	$x \Delta y = \max\{\min(x, 1 - y), \min(1 - x, y)\}$ .

As normas triangulares são empregadas para definir conjunções em raciocínio aproximado, enquanto que co-normas triangulares desempenham o mesmo papel para disjunções.

Uma regra de controle fuzzy, "se  $x = A$  então  $y = B$ " é representada por uma função de implicação fuzzy e é denotada por  $A \rightarrow B$ , onde  $A$  e  $B$  são conjuntos fuzzy definidos nos universos de discurso  $U$  e  $V$  com funções de pertinência  $\mu_A$  e  $\mu_B$ , respectivamente.

**Definição 2.15 (CONJUNÇÃO FUZZY)** *uma conjunção fuzzy é definida para todo  $u \in U$  e  $v \in V$  através de*

$$A \rightarrow B = A \times B = \int_{U \times V} \mu_A(u) * \mu_B(v) / (u, v), \quad (2.12)$$

onde  $*$  é um operador representando uma norma triangular.

6. Regra Max-Min para implicação fuzzy (Zadeh):

$$R_m = (A \times B) \cup (\text{not}A \times V) = \int_{U \times V} (\mu_A(u) \wedge \mu_B(v)) \vee (1 - \mu_A(u)) / (u, v). \quad (2.19)$$

7. Seqüência Padrão:

$$R_s = A \times V \rightarrow U \times B = \int_{U \times V} (\mu_A(u) > \mu_B(v)) / (u, v)$$

$$\text{onde } \mu_A(u) > \mu_B(v) = \begin{cases} 1 & \mu_A(u) \leq \mu_B(v) \\ 0 & \mu_A(u) > \mu_B(v). \end{cases} \quad (2.20)$$

8. Implicação fuzzy Booleana:

$$R_b = (\text{not}A \times V) \cup (U \times B) = \int_{U \times V} (1 - \mu_A(u)) \vee (\mu_B(v)) / (u, v). \quad (2.21)$$

9. Implicação fuzzy de Goguen:

$$R_\Delta = A \times V \rightarrow U \times B = \int_{U \times V} (\mu_A(u) \gg \mu_B(v)) / (u, v)$$

$$\text{onde } \mu_A(u) \gg \mu_B(v) = \begin{cases} 1 & \mu_A(u) \leq \mu_B(v) \\ \frac{\mu_A(u)}{\mu_B(v)} & \mu_A(u) > \mu_B(v) \end{cases} \quad (2.22)$$

## 2.7 Interpretação dos conectivos “e” e “também”

Na maioria dos controladores fuzzy o conectivo de sentenças “e” é usualmente interpretado como uma conjunção fuzzy no espaço do produto cartesiano das variáveis envolvidas. Por exemplo, na regra “se  $(x = A$  e  $y = B)$  então  $z = C$ ”, o antecedente é interpretado como um conjunto fuzzy no espaço  $U \times V$ , com a função de pertinência dada por,

$$\mu_{A \times B}(u, v) = \min\{\mu_A(u), \mu_B(v)\}, \quad (2.23)$$

ou

$$\mu_{A \times B}(u, v) = \mu_A(u) \cdot \mu_B(v), \quad (2.24)$$

onde  $U$  e  $V$  são universos de discurso associados com  $A$  e  $B$ , respectivamente.

Quando um sistema fuzzy é caracterizado por um conjunto de regras de controle fuzzy, a ordem em que aparecem as regras na base de conhecimento deve ser irrelevante. Isso implica que o conectivo de sentenças “também” deve possuir as propriedades de comutatividade e associatividade. Nesse sentido, deve-se notar que os operadores de normas triangulares e co-normas triangulares possuem tais propriedades e portanto se qualificam para a interpretação do conectivo “também”

Estudos a respeito da influência das funções de implicação e do conectivo “também” podem ser encontrados em [9,33,50,51,67,68].

Sob o ponto de vista prático utiliza-se a operação de união como interpretação do conectivo “também” em associação com as funções de implicação,  $R_c$  e  $R_p$ , já que acarretam em algoritmos simples de serem implementados.

## 2.8 Operadores Composicionais

Em geral, um operador composicional pode ser expresso como uma composição sup-star, onde "sup" denota o operador *supremo* e "star" denota um operador; como por exemplo *min*, *produto*, etc.; que é escolhido para uma aplicação específica. Na literatura podem ser encontrados quatro tipos de operadores composicionais utilizados como regra de inferência composicional:

- operação sup-min
- operação sup-produto
- operação sup-produto limitado
- operação sup-produto drástico

Em aplicações de controle os operadores sup-min e sup-produto são mais utilizados.

## 2.9 Mecanismos de Inferência

Os mecanismos de inferência empregados em controladores fuzzy são em geral muito mais simples do que aqueles utilizados em sistemas especialistas, já que na base de regras de um controlador o consequente de uma regra não é aplicado ao antecedente de outra regra. Em outras palavras, em controladores fuzzy não são empregados mecanismos de inferência em cadeia, já que as ações de controle são baseadas em inferências diretas de um único nível (Modus Ponens Generalizado)

Genericamente, uma base de regras possui a forma de um sistema MIMO<sup>1</sup>

$$R = \{R_{\text{MIMO}}^1, R_{\text{MIMO}}^2, \dots, R_{\text{MIMO}}^n\},$$

onde  $R_{\text{MIMO}}^i$  representa a regra:

$$\text{se } (x = A_i \text{ e } \dots, \text{ e } y = B_i) \text{ então } (z_1 = C_i, \dots, z_q = D_i).$$

o antecedente de  $R_{\text{MIMO}}^i$  forma um conjunto fuzzy  $A_i \times \dots \times B_i$  no espaço  $U \times \dots \times V$ . O consequente é a união de  $q$  ações de controle independentes. Portanto a  $i$ -ésima regra  $R_{\text{MIMO}}^i$  pode ser representada como uma implicação fuzzy,

$$R_{\text{MIMO}}^i : (A_i \times \dots \times B_i) \rightarrow (z_1 + \dots + z_n),$$

dessa forma, a base de regras  $R$  pode ser representada como a união das várias regras,

$$\begin{aligned} R &= \left\{ \bigcup_{i=1}^n R_{\text{MIMO}}^i \right\} \\ &= \left\{ \bigcup_{k=1}^q \bigcup_{i=1}^n [(A_i \times \dots \times B_i) \rightarrow z_k] \right\} \\ &= \{RB_{\text{MISO}}^1, RB_{\text{MISO}}^2, \dots, RB_{\text{MISO}}^q\} \end{aligned} \quad (2.25)$$

<sup>1</sup>Múltiplas Entradas e Múltiplas Saídas



com efeito, uma base de regras  $R$  de um controlador fuzzy é composta por um conjunto de sub-bases de regras  $RB_{MISO}^i$ , onde cada sub-base de regras consistindo de  $n$  regras de controle fuzzy com múltiplas variáveis de estado do processo e uma única variável de controle. Um sistema fuzzy MIMO pode então ser representado como uma coleção de sistemas fuzzy MISO<sup>2</sup>,

$$R = \{R_{MISO}^1, R_{MISO}^2, \dots, R_{MISO}^q\},$$

onde  $R_{MISO}^k$  representa a regra:

$$\text{se } (x = A_i \text{ e } \dots \text{ e } y = B_i) \text{ então } (z_k = D_i) \\ i = 1, \dots, n.$$

Vamos considerar a seguinte forma geral de regras de controle fuzzy para sistemas MISO para o caso de sistemas fuzzy com duas variáveis de entrada e uma variável de saída:

$$\begin{aligned} \text{entrada: } & x = A' \text{ e } y = B' \\ & R_1 : \text{ se } x = A_1 \text{ e } y = B_1 \text{ então } z = C_1 \\ \text{também } & R_2 : \text{ se } x = A_2 \text{ e } y = B_2 \text{ então } z = C_2 \\ & \dots \\ & \dots \\ \text{também } & R_n : \text{ se } x = A_n \text{ e } y = B_n \text{ então } z = C_n \\ \hline & z = C' \end{aligned}$$

onde  $x, y$  e  $z$  são variáveis linguísticas representando as variáveis de estado do processo e a variável de controle, respectivamente,  $A_i, B_i$ , e  $C_i$  são valores linguísticos das variáveis  $x, y$ , e  $z$  nos universos  $U, V$  e  $W$ , respectivamente, e  $i = 1, 2, \dots, n$ .

A regra de controle fuzzy “se  $(x = A_i \text{ e } y = B_i)$  então  $(z = C_i)$ ” é implementada como uma implicação (relação) fuzzy  $R$  e é definido como,

$$\begin{aligned} \mu_{R_i} & \triangleq \mu_{(A_i \text{ e } B_i \rightarrow C_i)}(u, v, w) \\ & = [\mu_{A_i}(u) \text{ e } \mu_{B_i}(v)] \rightarrow \mu_{C_i}(w), \end{aligned} \tag{2.26}$$

“ $A_i$  e  $B_i$ ” é um conjunto fuzzy  $A_i \times B_i$  em  $U \times V$ ;  $R_i \triangleq A_i \text{ e } B_i \rightarrow C_i$  é uma implicação fuzzy em  $U \times V \times W$ ; e  $\rightarrow$  denota uma função de implicação fuzzy.

O conseqüente  $C'$  é deduzido através da regra de inferência composicional sup-star empregando as definições de função de implicação fuzzy e dos conectivos “e” e “também”.

No que se segue, vamos considerar algumas propriedades de mecanismos de inferência<sup>3</sup>,

**Lema 2.1**  $(A', B') \circ \cup_{i=1}^n R_i = \cup_{i=1}^n (A', B') \circ R_i$ .

<sup>2</sup>Múltiplas Entradas e Única Saída

<sup>3</sup>nas equações a seguir,  $(\circ)$  denota o operador composicional sup-min e  $(\bullet)$  denota o operador sup-produto



**Lema 2.2** para as conjunções  $R_c$ ,  $R_p$ ,  $R_{bp}$  e  $R_{dp}$  temos,

$$(A', B') \circ (A_i \text{ e } B_i \rightarrow C_i) = [A' \circ (A_i \rightarrow C_i)] \cap [B' \circ (B_i \rightarrow C_i)]$$

*se*  $\mu_{A_i \times B_i} = \mu_{A_i} \wedge \mu_{B_i}$ ,

$$(A', B') \circ (A_i \text{ e } B_i \rightarrow C_i) = [A' \circ (A_i \rightarrow C_i)][B' \circ (B_i \rightarrow C_i)]$$

*se*  $\mu_{A_i \times B_i} = \mu_{A_i} \cdot \mu_{B_i}$ ,

**Lema 2.3** Se as entradas são "fuzzy singletons", ou seja,  $A' = u_0$ ,  $B' = v_0$ , então as funções de implicação  $R_c$  e  $R_p$  podem ser expressas como,

$$\begin{aligned} R_c &= \hat{\alpha} \wedge \mu_{C_i}(w), \\ R_c &= \hat{\alpha} \wedge \mu_{C_i}(w), \\ R_p &= \hat{\alpha} \cdot \mu_{C_i}(w), \\ R_p &= \hat{\alpha} \cdot \mu_{C_i}(w), \end{aligned} \tag{2.27}$$

onde  $\hat{\alpha} = \mu_{A_i}(u_0) \wedge \mu_{B_i}(v_0)$  e  $\hat{\alpha} = \mu_{A_i}(u_0) \cdot \mu_{B_i}(v_0)$ .

**Lema 2.4**  $(A', B') \bullet \cup_{i=1}^n R_i = \cup_{i=1}^n (A', B') \bullet R_i$ .

**Lema 2.5** para as conjunções  $R_c$ ,  $R_p$ ,  $R_{bp}$  e  $R_{dp}$  temos,

$$(A', B') \bullet (A_i \text{ e } B_i \rightarrow C_i) = [A' \bullet (A_i \rightarrow C_i)] \cap [B' \bullet (B_i \rightarrow C_i)]$$

*se*  $\mu_{A_i \times B_i} = \mu_{A_i} \wedge \mu_{B_i}$ ,

$$(A', B') \bullet (A_i \text{ e } B_i \rightarrow C_i) = [A' \bullet (A_i \rightarrow C_i)][B' \bullet (B_i \rightarrow C_i)]$$

*se*  $\mu_{A_i \times B_i} = \mu_{A_i} \cdot \mu_{B_i}$ ,

Pode-se concluir então que,

$$R_c : \mu_{C'} = \cup_{i=1}^n \alpha_i \wedge \mu_{C_i}, \tag{2.28}$$

$$R_p : \mu_{C'} = \cup_{i=1}^n \alpha_i \cdot \mu_{C_i}, \tag{2.29}$$

onde o fator de peso  $\alpha_i$  é uma medida da contribuição da  $i$ -ésima regra para a ação de controle fuzzy. O fator de peso em questão pode ser determinado por dois métodos o primeiro utiliza a operação de mínimo ( $\wedge$ ) para o produto cartesiano. O segundo emprega o produto algébrico ( $\cdot$ ) para o produto cartesiano.

Por simplicidade assume-se que uma determinada base de regras possui apenas duas regras,

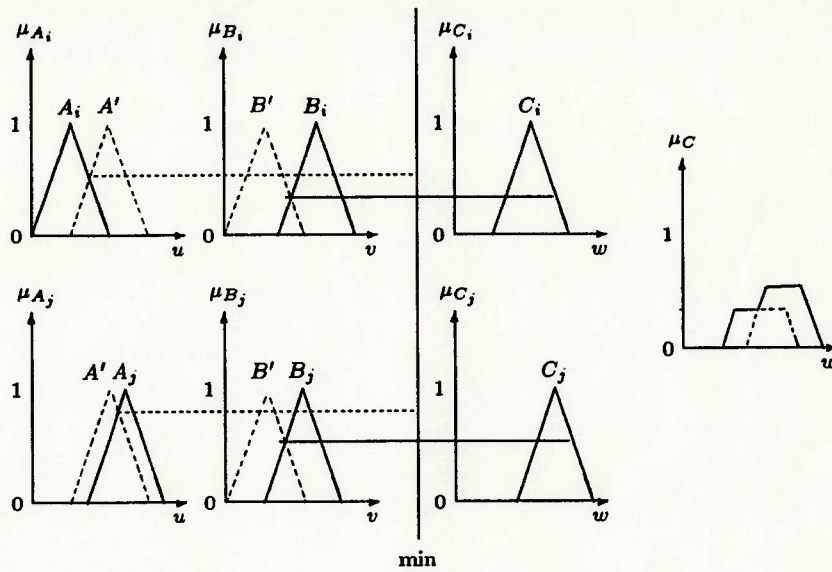


Figura 2.4: Interpretação gráfica com  $\hat{\alpha}$  e  $R_c$ .

$R_1$  : se  $x = A_1$  e  $y = B_1$  então  $z = C_1$

$R_2$  : se  $x = A_2$  e  $y = B_2$  então  $z = C_2$

A figura 2.4 ilustra graficamente o lema 2.2 com  $R_c$  e  $\hat{\alpha}_i$ . A figura 2.5 ilustra graficamente o lema 2.2 com  $R_p$  e  $\hat{\alpha}_i$ .

Em processos *on-line* as entradas são usualmente medidas através de sensores. Em geral, as medidas podem ser tratadas como “fuzzy singletons”. Dessa forma,  $\alpha_1$  e  $\alpha_2$  podem ser expressos como,

$$\begin{aligned}\alpha_1 &= \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0), \\ \alpha_2 &= \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0),\end{aligned}\tag{2.30}$$

onde  $\mu_{A_1}(x_0)$  e  $\mu_{B_1}(y_0)$  fazem o papel do grau de coincidência entre os dados do processo e os dados correspondentes a primeira regra.

Essas relações assumem um papel importante nos quatro tipos de raciocínio empregados atualmente em controladores fuzzy e descritos a seguir.

**Regra da Operação de Mínimo como Função de Implicação Fuzzy:** esse modo de raciocínio criado por Mandani utiliza a regra de operação de mínimo,  $R_c$  como uma função de implicação fuzzy. Nesse modo de raciocínio, a  $i$ -ésima regra conduz a seguinte decisão de controle,

$$\mu_{C_i}(w) = \alpha_i \wedge \mu_{C_i}(w),$$

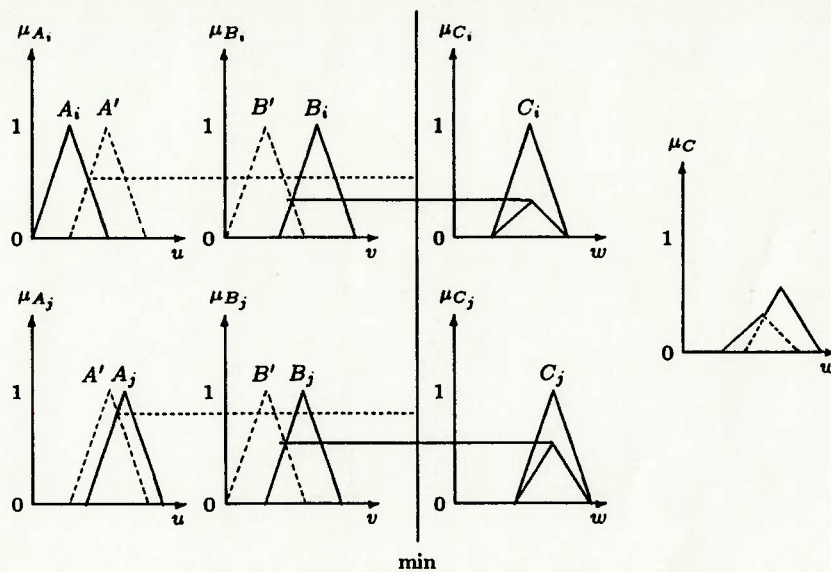


Figura 2.5: Interpretação gráfica com  $\alpha$  e  $R_p$ .

o que implica que a função de pertinência  $\mu_C$  da consequência  $C$  é dada por,

$$\begin{aligned} \mu_C(w) &= \mu_{C_1} \vee \mu_{C_2} \\ &= [\alpha_1 \wedge \mu_{C_1}(w)] \vee [\alpha_2 \wedge \mu_{C_2}(w)]. \end{aligned}$$

Para obter uma ação de controle determinística, uma estratégia para transformação de um conjunto fuzzy em valor escalar deve ser utilizada como será visto adiante. O processo de raciocínio aqui descrito é ilustrado na figura 2.6.

**Regra da Operação de Produto Algébrico como Função de Implicação Fuzzy:** esse método desenvolvido por Larsen utiliza a regra do produto algébrico,  $R_p$  como função de implicação. Nesse caso, a  $i$ -ésima regra, conduz a seguinte decisão de controle,

$$\mu_{C_i'}(w) = \alpha_i \cdot \mu_{C_i}(w),$$

consequentemente, a função de pertinência  $\mu_C$  é dada por,

$$\begin{aligned} \mu_C(w) &= \mu_{C_1'} \vee \mu_{C_2'} \\ &= [\alpha_1 \cdot \mu_{C_1}(w)] \vee [\alpha_2 \cdot \mu_{C_2}(w)]. \end{aligned}$$

O método de raciocínio aqui descrito é ilustrado graficamente na figura 2.7

**Método de Tsukamoto com Termos Linguísticos como Funções de Pertinência Monotônicas:** esse método proposto por Tsukamoto é uma simplificação do raciocínio

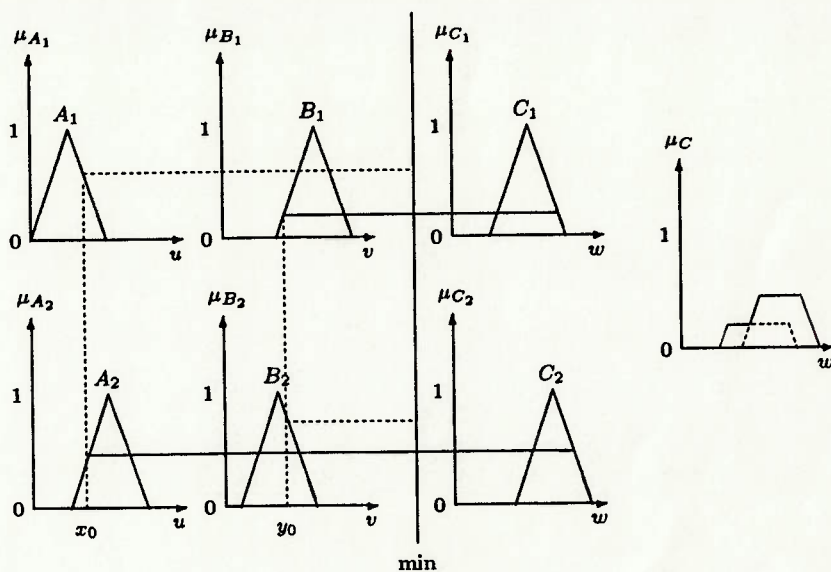


Figura 2.6: interpretação gráfica do raciocínio fuzzy utilizando a operação de mínimo como função de implicação.

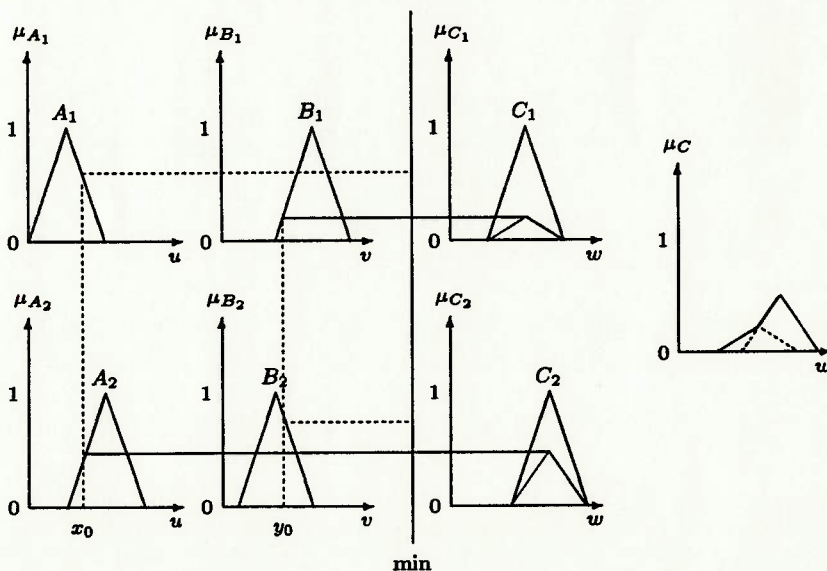


Figura 2.7: interpretação gráfica do raciocínio fuzzy utilizando o produto algébrico como função de implicação.



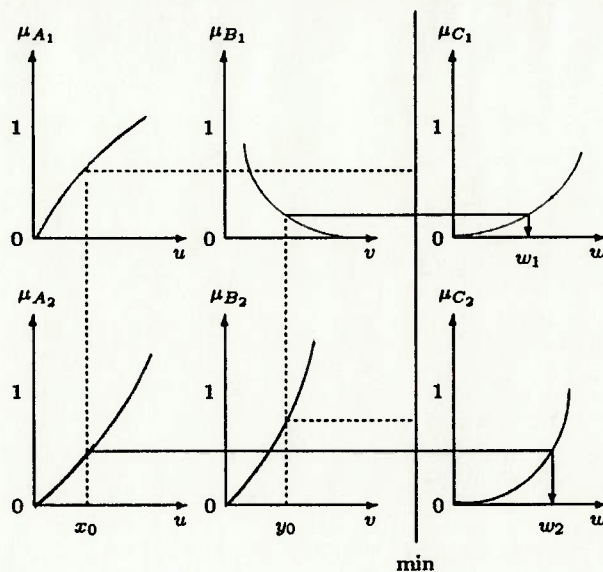


Figura 2.8: representação gráfica do método de Tsukamoto.

fuzzy proposto por Mandani onde as funções de pertinência dos conjuntos fuzzy  $A_i$ ,  $B_i$  e  $C_i$  são monotônicas.

O resultado inferido a partir da primeira regra é  $\alpha_1$  tal que  $\alpha_1 = \mu_{C_1}(w_1)$ . O resultado inferido a partir da segunda regra é  $\alpha_2 = \mu_{C_2}(w_2)$ . Uma ação de controle escalar pode então ser deduzida como,

$$z_0 = \frac{\alpha_1 w_1 + \alpha_2 w_2}{\alpha_1 + \alpha_2}.$$

O processo de raciocínio aqui descrito é representado graficamente na figura 2.8

**O Consequente de uma Regra como Função das Variáveis linguísticas de Entrada:** nesse modo de raciocínio, a  $i$ -ésima regra de controle possui a forma,

$$R_i : \text{se } (x = A_i \text{ e } \dots, \text{ e } y = B_i) \text{ então } z = f_i(x, \dots, y)$$

onde  $x, \dots, y$ , e  $z$  são variáveis linguísticas representando as variáveis de estado do processo e a variável de controle, respectivamente.  $A_i, \dots, B_i$  são valores linguísticos das variáveis  $x, \dots, y$  nos universos de discurso  $U, \dots, V$ , respectivamente, com  $i = 1, 2, \dots, n$ ; e  $f_i$  é função das variáveis de estado  $x, \dots, y$ .

Por simplicidade, vamos assumir que apenas duas regras compõem a base de conhecimento:

$$R_1 : \text{se } x = A_1 \text{ e } y = B_1 \text{ então } z = f_1(x, y)$$

$$R_2 : \text{se } x = A_2 \text{ e } y = B_2 \text{ então } z = f_2(x, y).$$

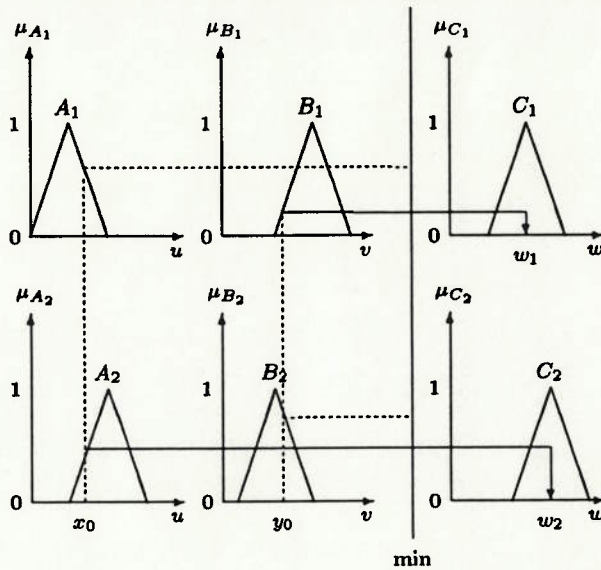


Figura 2.9: método simplificado.

A ação de controle inferida pela primeira e segunda regra são iguais, respectivamente  $a$ ,  $\alpha_1 f_1(x_0, y_0)$  e  $\alpha_2 f_2(x_0, y_0)$ . Correspondendo a uma ação de controle dada por,

$$z_0 = \frac{\alpha_1 f_1(x_0, y_0) + \alpha_2 f_2(x_0, y_0)}{\alpha_1 + \alpha_2}$$

**Método Simplificado:** esse método, largamente utilizado devido ao reduzido número de operações, é uma simplificação do método proposto por Mandani como o método de Tsukamoto. Uma ação de controle, pode então ser deduzida como,

$$z_0 = \frac{\alpha_1 w_1 + \alpha_2 w_2}{\alpha_1 + \alpha_2}$$

onde  $w_1$  e  $w_2$  correspondem ao valor de máximo das funções triangulares  $C_1$  e  $C_2$ . A figura 2.9 ilustra tal método.

## 2.10 Estratégias de Transformação Fuzzy-Escalar

Basicamente, a transformação fuzzy-escalar <sup>4</sup> corresponde a um mapeamento do espaço de ações de controle fuzzy definido sobre o universo de discurso para o espaço de ações de controle não-fuzzy ou escalares.

No presente, as estratégias mais comumente utilizadas são:

<sup>4</sup>defuzzification

**Método do Critério Máximo:** esse método produz como ação de controle o valor no qual a função de pertinência assume o valor máximo.

**Método da Média dos Máximos:** essa estratégia gera uma ação de controle obtida pelo valor médio de todas as ações de controle locais onde a função de pertinência assume o valor máximo. Mais especificamente, no caso de um universo discreto, a ação de controle pode ser expressa como,

$$z_0 = \sum_{j=1}^l \frac{w_j}{l},$$

onde  $w_j$  é o valor de suporte no qual a função de pertinência  $\mu_z(w_j)$  assume o seu valor máximo, e  $l$  é o número de valores suporte.

**Método do Centro de Gravidade:** é o método mais utilizado, e se baseia no cálculo do centro de gravidade da função de pertinência. No caso de um universo discreto o cálculo do centro de gravidade resulta em,

$$z_0 = \frac{\sum_{j=1}^n \mu_z(w_j) \cdot w_j}{\sum_{j=1}^n \mu_z(w_j)},$$

onde  $n$  é o número de níveis de quantização da saída.

# Capítulo 3

## Controle Fuzzy

### 3.1 Introdução

Um controlador fuzzy compreende quatro componentes principais: interface escalar-fuzzy, base de conhecimento, lógica de tomada de decisões, e interface fuzzy-escalar.

1. A interface escalar-fuzzy envolve as seguintes funções:
  - medição dos valores das variáveis de entrada.
  - mapeamento que transfere o intervalo dos valores das variáveis de entrada nos correspondentes universos de discurso.
  - transformação dos valores escalares das variáveis de entrada em valores linguísticos que correspondem a rótulos de conjuntos fuzzy <sup>1</sup>.
2. A base de conhecimento compreende o conhecimento do projetista a respeito do processo. Consiste de uma base de dados e um conjunto de regras de controle fuzzy:
  - a base de dados provê as definições necessárias para estabelecer as regras fuzzy e para a manipulação das informações do controlador.
  - o conjunto de regras caracteriza os objetivos de controle e a política de controle de especialistas.
3. A lógica de tomada de decisões é o núcleo de um controlador fuzzy, possui a capacidade de simular a habilidade humana de tomada de decisões baseada em conceitos fuzzy, inferindo ações de controle através do emprego de implicações fuzzy e regras de inferência em lógica fuzzy.
4. A interface fuzzy-escalar realiza as seguintes funções:
  - mapeamento que converte valores relativos aos universos de discurso nos intervalos correspondentes aos intervalos das variáveis de saída.

---

<sup>1</sup>fuzzification



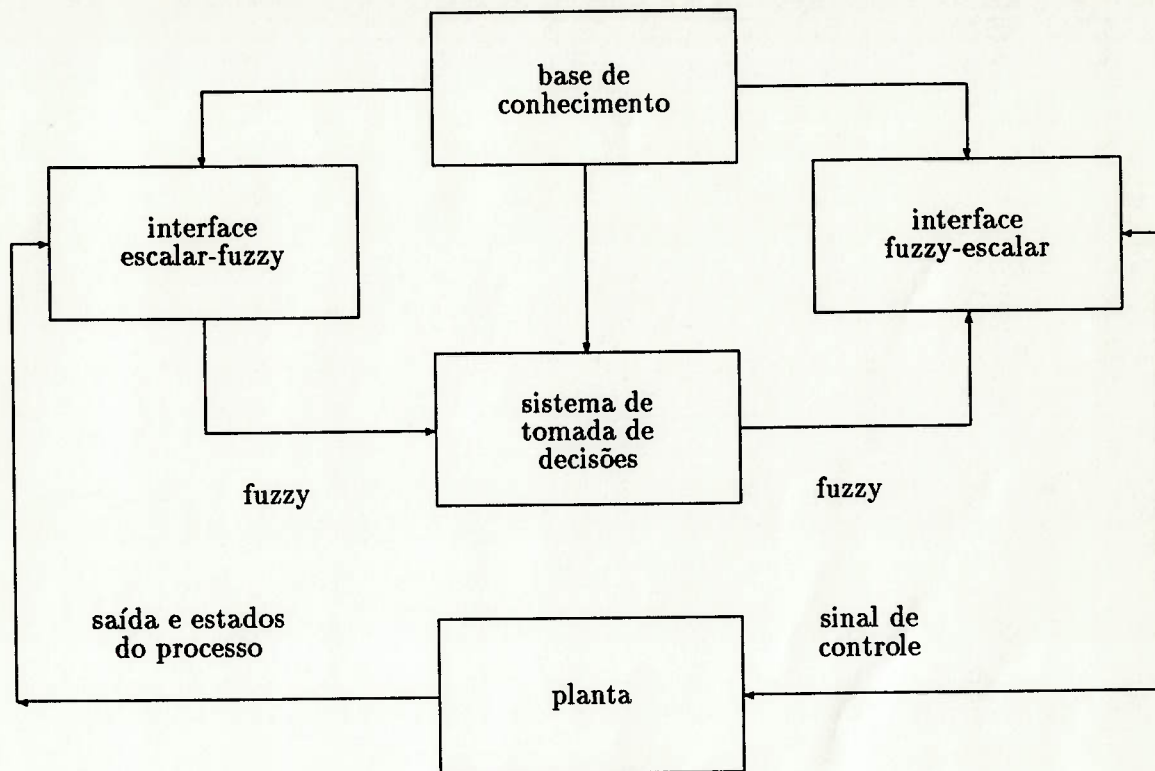


Figura 3.1: Controlador Fuzzy

- transformação fuzzy-escalar, que transforma uma ação de controle fuzzy numa ação de controle escalar.

A figura 3.1 ilustra a configuração básica de um controlador fuzzy.

Esse capítulo apresenta com detalhes a arquitetura de controladores fuzzy. As seções 3.2–3.6 apresentam alguns conceitos de lógica fuzzy e sua relação com controladores fuzzy. Os parâmetros estruturais para o projeto de controladores são discutidos na seção 3.7.

## 3.2 Declarações Condicionais Fuzzy e Regras de Controle Fuzzy

Em um controlador fuzzy o comportamento dinâmico de um sistema é caracterizado através de um conjunto de regras de descrição linguísticas baseadas num conhecimento especialista. O conhecimento especialista é usualmente expresso sob a forma:

se (um conjunto de condições são satisfeitas) então (um conjunto de conseqüências pode ser inferido).

Já que os antecedentes e consequentes de tais regras **se-então** estão associados a conceitos fuzzy, costuma-se denominá-las de declarações condicionais fuzzy. Nesse texto, o termo regra de controle fuzzy estará relacionado a uma declaração condicional fuzzy na qual o antecedente representa o estado do sistema e o consequente é a ação de controle.

Basicamente, regras de controle fuzzy permitem expressar políticas de controle e conhecimento a respeito do processo. A possibilidade de inclusão de múltiplas variáveis linguísticas tanto nos antecedentes como nos consequentes permite a representação de sistemas fuzzy de múltiplas entradas e múltiplas saídas (MIMO). Por exemplo, no caso de um sistema de duas entradas e uma saída, as regras de controle fuzzy assumem a seguinte forma:

$R_1$  : se  $x = A_1$  e  $y = B_1$  então  $z = C_1$ ,

$R_2$  : se  $x = A_2$  e  $y = B_2$  então  $z = C_2$ ,

...

$R_n$  : se  $x = A_n$  e  $y = B_n$  então  $z = C_n$ ,

onde  $x$ ,  $y$ , e  $z$  são variáveis linguísticas representando duas variáveis de estado e uma variável de controle;  $A_i$ ,  $B_i$ ,  $C_i$  são valores linguísticos das variáveis  $x$ ,  $y$ , e  $z$  no universo de discurso  $U$ ,  $V$  e  $W$ , respectivamente, com  $i = 1, 2, \dots, n$ . Um conectivo de sentenças liga implicitamente todas as regras de controle.

Uma regra de controle fuzzy, como por exemplo, “se ( $x = A_i$  e  $y = B_i$ ) então ( $z = C_i$ )” é implementado através de uma implicação (relação) fuzzy  $R_i$  que é definida como se segue:

$$\mu_{R_i}(u, v, w) \triangleq [\mu_{A_i}(u) \text{ e } \mu_{B_i}(v)] \rightarrow \mu_{C_i}(w), \quad (3.1)$$

onde ( $A_i$  e  $B_i$ ) é um conjunto fuzzy em  $A_i \times B_i$  no domínio  $U \times V$ ;  $R_i \triangleq (A_i \text{ e } B_i) \rightarrow C_i$  é uma implicação (relação) fuzzy em  $U \times V \times W$ ; e  $\rightarrow$  indica uma função de implicação fuzzy.

### 3.3 Operador de Transformação Escalar-Fuzzy

Um operador de transformação escalar-fuzzy tem o efeito de transformar dados escalares em conjuntos fuzzy. Simbolicamente, podemos expressar,

$$x = \text{tef}(x_0),$$

onde  $x_0$  é um valor escalar de uma variável do processo;  $x$  é um conjunto fuzzy, e  $\text{tef}$  representa um operador de transformação escalar-fuzzy.

### 3.4 Operadores de Conectivos de Sentença

Um controlador fuzzy consiste de um conjunto de regras de controle que estão relacionadas através dos conceitos de implicação fuzzy e regra de inferência composicional sup-star. As regras são combinadas através de conectivos **e** e **também**. Já que cada regra de controle é representada através de uma relação fuzzy, o comportamento global de um

sistema fuzzy é caracterizado através dessas relações. Ou seja, Um sistema fuzzy pode ser caracterizado por uma única relação fuzzy que representa a combinação de todas as regras. A combinação em questão envolve o conectivo de sentenças também. Simbolicamente podemos representar,

$$R = \text{também}(R_1, R_2, \dots, R_i, \dots, R_n),$$

onde também representa um conectivo de sentenças.

### 3.5 Operador Composicional

Para inferir a saída  $z$  para um dado estado do processo representado por  $x$  e  $y$  e pela relação fuzzy  $R$ , a regra de inferência composicional é aplicada,

$$z = y \circ (x \circ R). \quad (3.2)$$

### 3.6 Operador de Transformação Fuzzy-Escalar

A saída de qualquer processo de inferência é um conjunto fuzzy, especificando a distribuição de possibilidade da ação de controle. Na prática, entretanto, para a utilização em controle de processos, uma ação de controle escalar é requerida. Conseqüentemente torna-se necessário transformar o conjunto fuzzy inferido num valor escalar, através da utilização de um operador de transformação fuzzy-escalar, denotado por,

$$z_0 = \text{tfe}(z).$$

Onde  $z_0$  é a ação de controle escalar e  $\text{tfe}$  é o operador de transformação fuzzy-escalar.

### 3.7 Parâmetros de Projeto para um Controlador Fuzzy

Os principais parâmetros para um projeto de um controlador fuzzy são os seguintes:

1. estratégias de transformação escalar-fuzzy e a interpretação do operador  $\text{tfe}$ ;
2. base de dados:
  - discretização/normalização dos universos de discurso,
  - partição dos espaços de entrada e saída através de conjuntos fuzzy,
  - completeza,
  - escolha da função de pertinência para os conjuntos fuzzy;
3. base de regras:
  - escolha das variáveis de estado (entrada) e variáveis de controle (saída),



- fonte e derivação de regras de controle fuzzy,
  - tipo de regras de controle fuzzy,
4. lógica de tomada de decisões:
- definição de uma implicação fuzzy,
  - interpretação do conectivo de sentenças e,
  - interpretação do conectivo de sentenças também,
  - definição de um operador composicional,
  - mecanismo de inferência;
5. estratégias de transformação fuzzy-escalar e interpretação do operador tfe.

### 3.7.1 Estratégias de Transformação Escalar-Fuzzy

A transformação escalar-fuzzy está relacionada com o conceito de imprecisão em linguagens naturais. Trata-se de uma avaliação subjetiva que transforma uma medida num valor linguístico, e portanto, pode ser definida como um mapeamento entre um espaço observado de entrada e conjuntos fuzzy definidos sobre o mesmo espaço (universo de discurso).

Em aplicações de controle fuzzy os dados observados são usualmente escalares. Já que a manipulação das informações em controladores fuzzy são baseados na teoria de conjuntos fuzzy, a transformação escalar-fuzzy se faz necessária no início do tratamento das informações.

Experiências com o projeto de controladores fuzzy, sugerem os seguintes procedimentos para a transformação fuzzy-escalar:

- um operador de transformação escalar-fuzzy “conceitualmente” converte um valor escalar num fuzzy singleton (definição 2.2) dentro de um determinado universo de discurso. Basicamente um fuzzy singleton é um valor preciso e portanto nenhuma imprecisão é introduzido nesse caso. Essa estratégia tem sido largamente utilizada em aplicações de controle devido a sua facilidade de implementação. Uma entrada  $x_0$  é interpretada como um conjunto fuzzy  $A$  com uma função de pertinência  $\mu_A(x)$  igual a zero, exceto no ponto  $x_0$ , onde  $\mu_A(x_0) = 1$ .
- dados sob observação são perturbados em geral por ruídos aleatórios. Nesse caso, um operador de transformação escalar-fuzzy pode converter dados probabilísticos em números fuzzy (dados possibilísticos). Em [69] utiliza-se um triângulo isósceles, como transformação escalar-fuzzy, cujo vértice corresponde à média de um conjunto de valores e a base é igual a duas vezes o desvio padrão.
- em sistemas de grande porte, algumas observações relacionadas ao comportamento do sistema são precisas, enquanto que outras são medidas apenas no sentido estatístico, e algumas referidas como *híbridas*, requerem caracterizações probabilísticas e possibilísticas. A estratégia de transformação escalar-fuzzy nesse caso requer a utilização do conceito de *números híbridos* [45], que envolvem tanto incerteza (números fuzzy) como aleatoriedade (números aleatórios).



### 3.7.2 Base de Dados

A base de conhecimento de um controlador fuzzy compreende duas componentes, uma base de dados e uma base de regras de controle. Os conceitos associados à base de dados são utilizados para caracterizar as regras de controle e a manipulação de dados fuzzy. Tais conceitos são definidos subjetivamente e baseados na experiência. Nessa seção serão discutidos alguns aspectos importantes relacionados a construção de base de dados para controladores fuzzy.

#### Discretização/Normalização de Universos de Discurso

A representação de informações incertas através de conjuntos fuzzy leva ao problema da representação de tais informações em computadores. Em geral, a representação depende da natureza do universo de discurso. Um universo de discurso num controlador fuzzy pode ser discreto ou contínuo.

**Discretização de Um Universo de Discurso:** a discretização de um universo é freqüentemente referida como uma quantização. Uma quantização discretiza um universo em um determinado número de segmentos (níveis de quantização). Cada segmento é rotulado como um elemento genérico. Um conjunto fuzzy é então definido assinalando um valor do grau de pertinência a cada elemento do universo discreto.

O número de quantizações deve ser grande o suficiente para uma adequada aproximação e ainda pequeno o suficiente para evitar um elevado consumo de memória nos computadores. A escolha dos níveis de quantização têm uma influência direta na precisão que pode ser obtida durante o controle. Por exemplo, se um determinado universo é quantizado para cada 5 unidades de medida ao invés de 10 unidades, então o controlador é duas vezes mais sensível em relação às variáveis observadas.

Para o propósito de discretização, um mapeamento é requerido, para a transformação das variáveis medidas em valores do universo discreto.

O mapeamento pode ser uniforme (linear) ou não uniforme, ou ambos. A escolha dos níveis de quantização reflete obviamente um conhecimento *a priori*.

Por exemplo num sistema fuzzy com 3 variáveis de entrada e uma variável de saída pode-se utilizar regras de controle do seguinte tipo:

Pode-se representar simbolicamente um controlador fuzzy com tais regras como,

$$K_4[u(k)] = F[K_1.e(k), K_2.ie(k), K_3.ce(k)],$$

onde  $F$  denota a relação fuzzy definida pela base de regras,  $e(k)$  representa o erro do processo,  $ie(k)$  representa a integral do erro,  $ce(k)$  representa a variação do erro e  $(K_1, K_2, K_3, K_4)$  representam mapeamentos adequados.

Um exemplo de discretização é mostrado na tabela 3.1, onde um universo de discurso é discretizado em 13 níveis  $(-6, -5, \dots, 0, \dots, +6)$  com sete conjuntos fuzzy definidos sobre ele: *NB* (*Negative-Big*), *NM* (*Negative-Medium*), *NS* (*Negative-Small*), *ZE* (*Zero*), *PS* (*Positive-Small*), *PM* (*Positive-Medium*), *PB* (*Positive-Big*). Em geral, devido a discretização, o desempenho de um controlador fuzzy é pouco sensível a pequenos desvios nos valores das variáveis de estado.

Tabela 3.1: Quantização

nível	intervalo	<i>NB</i>	<i>NM</i>	<i>NS</i>	<i>ZE</i>	<i>PS</i>	<i>PM</i>	<i>PB</i>
-6	$x_0 \leq -3.2$	1.0	0.3	0.0	0.0	0.0	0.0	0.0
-5	$-3.2 < x_0 \leq -1.6$	0.7	0.7	0.0	0.0	0.0	0.0	0.0
-4	$-1.6 < x_0 \leq -0.8$	0.3	1.0	0.3	0.0	0.0	0.0	0.0
-3	$-0.8 < x_0 \leq -0.4$	0.0	0.7	0.7	0.0	0.0	0.0	0.0
-2	$-0.4 < x_0 \leq -0.2$	0.0	0.3	1.0	0.3	0.0	0.0	0.0
-1	$-0.2 < x_0 \leq -0.1$	0.0	0.0	0.7	0.7	0.0	0.0	0.0
0	$-0.1 < x_0 \leq +0.1$	0.0	0.0	0.3	1.0	0.3	0.0	0.0
+1	$+0.1 < x_0 \leq +0.2$	0.0	0.0	0.0	0.7	0.7	0.0	0.0
+2	$+0.2 < x_0 \leq +0.4$	0.0	0.0	0.0	0.3	1.0	0.3	0.0
+3	$+0.4 < x_0 \leq +0.8$	0.0	0.0	0.0	0.0	0.7	0.7	0.0
+4	$+0.8 < x_0 \leq +1.6$	0.0	0.0	0.0	0.0	0.3	1.0	0.3
+5	$+1.6 < x_0 \leq +3.2$	0.0	0.0	0.0	0.0	0.0	0.7	0.7
+6	$+3.2 \leq x_0$	0.0	0.0	0.0	0.0	0.0	0.3	1.0

Tabela 3.2: Normalização e Conjuntos Fuzzy Primários Utilizando Definição Funcional

Universo Normalizado	Segmentos Normalizados	Intervalo	$u_f$	$\sigma_f$	Conjuntos fuzzy primários
[-1.0, +1.0]	[-1.0, -0.5]	[-6.9, -4.1]	-1.0	0.4	<i>NB</i>
	[-0.5, -0.3]	[-4.1, -2.2]	-0.5	0.2	<i>NM</i>
	[-0.3, 0.0]	[-2.2, 0.0]	-0.2	0.2	<i>NS</i>
	[0.0, +0.2]	[0.0, +1.0]	0.0	0.2	<i>ZE</i>
	[+0.2, +0.6]	[+1.0, +2.5]	0.2	0.2	<i>PS</i>
	[+0.6, +1.0]	[+2.5, +4.5]	0.5	0.2	<i>PM</i>
			1.0	0.2	<i>PB</i>

**Normalização de Universos de Discurso:** a normalização de um universo de discurso requer a discretização do mesmo num número finito de segmentos, onde cada segmento é mapeado num outro segmento correspondente no universo normalizado. Nesse caso, um conjunto fuzzy é então definido assinalando-se uma função explícita para a sua função de pertinência. A normalização de um universo contínuo também envolve um conhecimento *a priori* a respeito do intervalo de valores ocupados pela variáveis de estado e de controle. O mapeamento pode ser uniforme, não uniforme ou ambos. Um exemplo é mostrado na tabela 3.2 onde o universo de discurso [-6.9, +4.5], é transformado no intervalo normalizado [-1.0, +1.0]. Os parâmetros  $u_f$  e  $\sigma_f$  se referem a definição de funções gaussianas como poderá ser visto a seguir na equação 3.4.

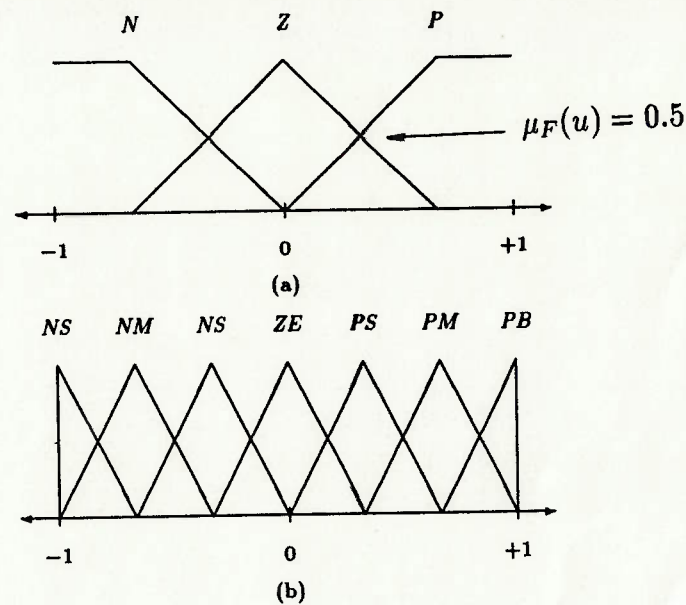


Figura 3.2: representação gráfica de partições fuzzy.

### Partição Fuzzy de Universos de Discurso

Uma variável linguística no antecedente de uma regra de controle fuzzy forma um espaço fuzzy de entrada com respeito a um determinado universo de discurso, enquanto que uma variável no conseqüente de uma regra forma um espaço fuzzy de saída. Em geral uma variável linguística está associada com um conjunto de termos, onde cada termo é definido sobre o mesmo universo de discurso. Uma partição fuzzy, então, determina quantos termos poderiam existir em um dado universo. Esse processo é equivalente a encontrar o número de conjuntos fuzzy primários. O número de conjuntos fuzzy primários determina a granularidade do controle obtido com um controlador fuzzy. Os conjuntos fuzzy primários usualmente possuem um significado, tal como *NB*: *Negative-Big*; *NM*: *Negative-Medium*; *NS*: *Negative-Small*; *ZE*: *Zero*; *PS*: *Positive-Small*; *PM*: *Positive-Medium*; *PB*: *Positive-Big*; *N*: *Negative*; *Z*: *Zero*; *P*: *Positive*. Dois exemplos de partição sobre o mesmo universo de discurso normalizado  $[-1, +1]$  são mostrados na figura 3.2. As funções de pertinência utilizadas possuem a forma de triângulos e trapézios. A cardinalidade de um conjunto de termos no espaço fuzzy de entrada determina o número máximo de regras de controle que podem ser construídas. Por exemplo, no caso de um sistema fuzzy com duas variáveis de entrada ( $x$  e  $y$ ) e uma variável de saída ( $z$ ), se a cardinalidade do conjunto de termos  $T(x)$  é igual a 3 e do conjunto de termos  $T(y)$  é igual a 7, então o número máximo de regras é  $3 \times 7$ . Deve-se notar que partição não é determinística e não possui solução única. Um procedimento heurístico é usualmente necessário para se estabelecer uma partição ótima.



### Completeza

Intuitivamente, um algoritmo de controle fuzzy deve sempre ser capaz de inferir uma ação de controle apropriada para cada estado do processo. Essa propriedade é denominada de *completeza*. A completeza de um controlador fuzzy está relacionada a base de dados, base de regras ou ambas.

**Base de Dados:** a estratégia relativa a base de dados está relacionada com os suportes (definição 2.2) nos quais os conjuntos fuzzy primários são definidos. A união desses suportes deve cobrir o universo de discurso em relação a algum nível  $\epsilon$  de certeza. Essa propriedade de um controlador fuzzy é denominada *completeza- $\epsilon$* . Em geral, escolhe-se o nível  $\epsilon$  no ponto de crossover (definição 2.2) como mostrado na figura 3.2. Nesse sentido, uma regra dominante sempre existe e está associada com um grau de certeza maior do que 0.5. Num caso extremo, duas regras dominantes são ativadas com um grau de certeza igual a 0.5.

**Base de Regras:** a estratégia relativa a base de regras está relacionada com as próprias regras de controle. A propriedade de completeza deve ser observada. Uma regra é adicionada sempre que uma determinada condição não está incluída na base de regras, ou sempre que o grau de certeza para alguma condição de entrada é menor que algum nível por exemplo 0.5.

### Função de Pertinência para um Conjunto Fuzzy Primário

Existem dois métodos para se definir conjuntos fuzzy dependendo do tipo de universo de discurso utilizado:

- Universo discreto: definição numérica,
- Universo contínuo: definição funcional.

**Definição Numérica:** nesse caso, a função de pertinência é representada por um vetor de números cuja dimensão depende do grau de discretização. Um exemplo ilustrativo é mostrado na tabela 3.1. Nesse caso, a função de pertinência de cada conjunto fuzzy possui a forma,

$$\mu_f(u) = \sum_{i=1}^5 a_i/u_i, \quad (3.3)$$

onde

$$a = [0.3, 0.7, 1.0, 0.7, 0.3].$$

Com exceção de *NB* e *PB* que possuem apenas 3 elementos no conjunto suporte.

**Definição Funcional:** a definição funcional expressa a função de pertinência sob a forma funcional, tipicamente uma função gaussiana, função triangular, função trapezoidal, etc. A definição funcional pode facilmente ser adaptada às mudanças na normalização



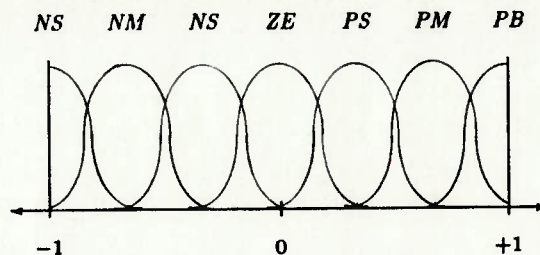


Figura 3.3: definição funcional

do universo. A tabela 3.2 e figura 3.3, mostram um exemplo de definição funcional expressada como:

$$\mu_f(x) = \exp \left\{ \frac{-(x - u_f)^2}{2\sigma_f^2} \right\}, \quad (3.4)$$

note que se o universo normalizado é alterado, os parâmetros  $u_f$ ,  $\sigma_f$  podem ser alterados de acordo. A escolha dos graus de pertinência está baseado em critérios subjetivos de decisão. Em particular, como já foi mencionado antes, se os dados medidos são perturbados por ruídos, as funções de pertinência devem ser suficientemente largas para reduzir a sensibilidade ao ruído.

### 3.7.3 Base de Regras

#### Escolha das Variáveis de Estado do Processo e Variáveis de Controle

A escolha apropriada das variáveis de estado do processo e das variáveis de controle é essencial para a caracterização da operação de um sistema fuzzy. Como já foi enfatizado, a experiência desempenha um papel importante nesse processo de escolha. Tipicamente as variáveis de estado são: o erro, a derivada do erro, a integral do erro, etc.

#### Fonte e Derivação de Regras de Controle

Em geral, considera-se quatro modos de derivação de regras de controle fuzzy. Esses quatro modos de derivação não podem ser considerados como mutuamente exclusivos.

1. *Experiência de Especialistas*: deve-se notar que a maioria das decisões tomadas pelos seres humanos se referem a dados linguísticos e não a dados numéricos. Visto sob esse enfoque, regras de controle fuzzy constituem uma excelente ferramenta para a caracterização do comportamento humano e análise de decisão. Isso explica porque a maioria dos controladores fuzzy se apóiam no conhecimento e na experiência de especialistas expressos através de regras *se-então*.
2. *Modelagem das Ações do Operador*: em muitos casos, aonde é difícil obter um modelo matemático preciso do processo utiliza-se operadores humanos para a realização

do controle dinâmico. Um operador humano pode controlar muitos sistemas com bastante sucesso sem a utilização de uma avaliação numérica precisa das variáveis do processo.

Para substituir um operador humano por um controlador fuzzy, basta modelar as ações de controle do operador através de regras fuzzy.

3. *Modelo Fuzzy do Processo*: a descrição linguística das características de um processo pode ser visto como um modelo fuzzy do processo. Baseado no modelo fuzzy do processo, pode-se criar regras de controle que garantam um desempenho ótimo para o sistema. Apesar desse método ser mais difícil de ser utilizado, ele permite obter melhor desempenho e segurança, e provê uma estrutura para tratamento teórico dos controladores fuzzy [90,92].
4. *Aprendizagem*: muitos controladores tem sido criados para emular o comportamento humano de tomadas de decisões, mas um número pequeno é baseado no processo de aprendizagem, ou seja, na habilidade de criar regras de controle fuzzy e modificá-las baseado na experiência. Procyk e Mandani [78] desenvolveram o primeiro *Controlador com Auto-Organização* (SOC). Tal controlador possui uma estrutura hierárquica contendo duas bases de regras. O nível inferior corresponde a um controlador fuzzy convencional. O segundo nível é composto por um conjunto de "meta-regras" que simulam a habilidade humana de aprendizagem criando e modificando as regras do nível inferior objetivando um aumento de desempenho do sistema.

### Tipos de Regras de Controle

Basicamente, podemos considerar dois tipos de regras de controle fuzzy,

1. *Regras de Controle Fuzzy de Avaliação de Estado*: no caso de sistemas MISO são caracterizados por uma coleção de regras do seguinte tipo:

$$\begin{aligned}
 R_1 &: \text{se } x = A_1, \dots, \text{e } y = B_1 \text{ então } z = C_1 \\
 R_2 &: \text{se } x = A_2, \dots, \text{e } y = B_2 \text{ então } z = C_2 \\
 &\dots \\
 R_n &: \text{se } x = A_n, \dots, \text{e } y = B_n \text{ então } z = C_n
 \end{aligned}$$

onde  $x, \dots, y$  e  $z$  são variáveis linguísticas representando as variáveis de estado do processo e a variável de controle;  $A_i, \dots, B_i$  e  $C_i$  ( $i = 1, \dots, n$ ) são valores linguísticos das variáveis  $x, \dots, y$  e  $z$  nos universos de discurso  $U, \dots, V$  e  $W$ , respectivamente.

O conseqüente pode assumir uma forma mais genérica se for representado como uma função das variáveis de estado,

$$R_i : \text{se } x = A_i, \dots, \text{e } y = B_i \text{ então } z = f_i(x, \dots, y).$$

O controlador baseado em regras de avaliação de estado inicialmente identifica o valor do estado do processo (por exemplo, o erro, a integral do erro) no instante  $k$  e calcula a ação de controle para o instante  $k + 1$  como uma função de  $x, \dots, y$  e do conjunto de regras declarado.

2. *Regras de Controle para Avaliação de Objetivos*: Yasunobu, Miyamoto e Ihara [113] propuseram um novo algoritmo onde a ação de controle  $u$  no instante  $k$  é decidida avaliando-se qual será o seu efeito sobre os estados e os objetivos de controle no instante  $k+1$ . Tal técnica é denominada *Controle Fuzzy com Avaliação de Objetivos* ou *Controle Fuzzy Preditivo*. As regras em questão são denominadas Regras de Controle para Avaliação de Objetivos. Uma regra típica possui a seguinte forma,

$$R_i : \text{se } (u = C_i \rightarrow (x = A_i \text{ e } y = B_i)) \text{ então } u = C_i.$$

Em termos linguísticos, a regra é interpretada como: “se o índice de desempenho  $x$  é igual a  $A_i$  e o índice  $y$  é igual a  $B_i$  quando a ação de controle é escolhida como  $u$  então essa regra é selecionada e a ação de controle é feita igual a  $u$ ”. Nesse sentido a ação de controle é otimizada segundo vários índices de desempenho simultaneamente.

Esse método foi aplicado com sucesso no sistema de operação automático de trens, desenvolvido pela Hitachi Corporation para o metrô da cidade de Sendai (Japão) e se consagrou como uma das primeiras aplicações comerciais de controladores fuzzy.

#### 3.7.4 Lógica de Tomada de Decisões

Uma análise detalhada a respeito de lógica de tomada de decisões foi apresentada no capítulo 2

#### 3.7.5 Estratégias de Transformação Fuzzy-Escalar

Veja capítulo 2.



## Capítulo 4

# Controladores Fuzzy × Controladores Lineares

### 4.1 Introdução

Esse capítulo estabelece uma relação entre controladores fuzzy e controladores lineares clássicos do tipo PID (P,PI,PD).

A escolha apropriada das variáveis de estado e das variáveis de controle utilizadas para compor a base de regras são essenciais na determinação do comportamento de um controlador fuzzy, como mostrado na seção 4.2. Determina-se quais são as variáveis fuzzy que devem ser escolhidas para se obter controladores fuzzy do tipo Proporcional, Proporcional-Derivativo, Proporcional-Integral e Proporcional-Integral-Derivativo.

Controladores fuzzy possuem em geral um comportamento não linear. Se uma base de regras adequada for construída, pode ser provado que a medida que o número de regras cresce, o comportamento do controlador fuzzy torna-se cada vez mais linear. No limite, número de regras tendendo a infinito, obtem-se um controlador fuzzy linear. Esse comportamento limite é discutido na seção 4.3.

A seção 4.3.1 estabelece uma interpretação geométrica para tal comportamento limite, relacionando nesse caso as constantes desse controlador fuzzy hipotético e as constantes que determinam o comportamento de um controlador PI.

A seção 4.3.2 enuncia os teoremas limites estabelecidos por Buckley e Ying [19]. É provado algebricamente que para um número de entradas quaisquer, cada uma delas representando uma derivada  $i$ -ésima do erro do processo, o controlador fuzzy assume o comportamento limite desde que sejam adotadas algumas hipóteses sobre: o processo de transformação escalar-fuzzy, as regras de controle e o procedimento empregado para avaliar tais regras, e o algoritmo de transformação fuzzy-escalar.

Num caso especial, um controlador fuzzy pode assumir um comportamento linear com um número finito de regras como pode ser observado na seção 4.4.

Na seção 4.5 são realizadas algumas simulações com controladores fuzzy construídos de forma a obter um comportamento próximo de um controlador PI linear.



## 4.2 Escolha de Variáveis

A escolha apropriada das variáveis de estado e das variáveis de controle é essencial para a caracterização de um sistema fuzzy e corresponde ao o que denominamos de *identificação estrutural* na teoria de identificação de sistemas. Tipicamente as variáveis de estado são: o erro ( $e$ ), a variação do erro ( $ce$ ), a integral do erro ( $ie$ ), etc. Em sistemas MISO as variáveis de controle são: o esforço de controle ( $u$ ) ou a variação do esforço de controle ( $cu$ ).

A quase totalidade dos controladores fuzzy se utilizam do erro ( $e$ ) e da variação do erro ( $ce$ ) como variáveis de estado e a variação do esforço de controle ( $cu$ ) como variável de controle. Esse controlador é em geral denominado *Controlador Fuzzy Proporcional-Integral* em função de seu comportamento análogo a um controlador PI linear. Um controlador fuzzy com tais variáveis possui em geral um comportamento não linear, como será visto a seguir.

Através das variáveis citadas acima pode-se derivar uma família de controladores fuzzy do tipo PID (P,PI,PD) como poderá ser observado.

### 4.2.1 Controlador Fuzzy PID

Um *Controlador Fuzzy PID* pode ser obtido utilizando-se o o erro do processo  $e = r - y$  onde  $r$  é o sinal de referência e  $y$  é a saída do sistema, a variação do erro  $ce$  e a variação do erro de segunda ordem  $se$  como variáveis de estado e a variação do esforço de controle  $cu$  como variável de controle. Nesse caso as regras de controle assumem a seguinte forma,

$$\text{se } e = A_i \text{ e } ce = B_i \text{ e } se = C_i \text{ então } cu = D_i.$$

O comportamento desse controlador pode ser representado por,

$$cu(k) = F(e(k), ce(k), se(k)), \quad (4.1)$$

onde  $F$  representa o algoritmo de raciocínio fuzzy,  $ce(k) = \Delta e(k)$ ,  $cu(k) = \Delta u(k)$  e  $se(k) = \Delta^2 e(k)$ .

A transformada de Laplace de um controlador PID é dado por,

$$U(s) = K_P \left( 1 + \frac{1}{T_I s} + T_D s \right) E(s), \quad (4.2)$$

onde  $K_P$  é a constante de proporcionalidade,  $T_I$  é a constante de tempo integrativa e  $T_D$  é a constante de tempo derivativa.

Discretizando-se através da aproximação de Euler,

$$s = \frac{z - 1}{h} \quad (4.3)$$

onde  $h$  é o intervalo de amostragem, obtém-se a seguinte equação de diferenças,

$$\Delta u(k) = K_P \Delta e(k) - \frac{K_P h}{T_I} e(k) - \frac{K_P T_D}{h} \Delta^2 e(k). \quad (4.4)$$

De 4.4 conclue-se que um controlador PID linear possui a saída  $\Delta u(k)$  como combinação linear de  $e(k)$ ,  $\Delta e(k)$  e  $\Delta^2 e(k)$ .

Logo pode-se interpretar 4.1 como,

$$cu(k) = K_1 e(k) + K_2 ce(k) + K_3 se(k), \quad (4.5)$$

onde  $K_1, K_2, K_3 = f(e(k), ce(k), se(k))$ .

De maneira similar pode-se derivar as regras que resultam em controladores fuzzy P, PI e PD.

### 4.2.2 Controlador Fuzzy Proporcional

Um *Controlador Fuzzy Proporcional* pode ser obtido utilizando-se o erro  $e$  e o esforço de controle  $u$  como variáveis. Nesse caso as regras de controle possuem a seguinte forma,

$$\text{se } e = A_i \text{ então } u = B_i.$$

### 4.2.3 Controlador Fuzzy Proporcional Derivativo

Um *Controlador Fuzzy Proporcional Derivativo* pode ser obtido utilizando-se o erro  $e$  e a variação do erro  $ce$  como variáveis de estado e o esforço de controle  $u$  como variável de controle.

As regras de controle com tais variáveis assume a forma,

$$\text{se } e = A_i \text{ e } ce = B_i \text{ então } u = C_i$$

### 4.2.4 Controlador Fuzzy Proporcional Integral

Um *Controlador Fuzzy Proporcional Integral* pode ser obtido através da utilização do erro  $e$  e da variação do erro  $ce$  como variáveis de estado e do esforço de controle  $cu$  como variável de controle.

As regras de controle com tais variáveis assume a forma,

$$\text{se } e = A_i \text{ e } ce = B_i \text{ então } cu = C_i$$

## 4.3 Comportamento Limite em Controladores Fuzzy

Nessa seção, prova-se que é possível obter controladores fuzzy lineares numa condição limite em que o número de regras é levado ao infinito.

A seção 4.3.1 estabelece uma interpretação geométrica para tal comportamento limite, relacionando nesse caso as constantes desse controlador fuzzy e as constantes de um controlador PI linear. Esse resultado foi desenvolvido por Tang e Mulholland em [95].

A seção 4.3.2 enuncia os teoremas limites estabelecidos por Buckley e Ying [19]. É provado algebricamente que para um número de entradas quaisquer, um controlador MISO com um número de regras tendendo a infinito, pode assumir um comportamento linear desde que hipóteses adequadas sejam adotadas.

### 4.3.1 Interpretação Geométrica

Considere um controlador discreto onde o erro  $e(k)$  no instante  $k$  é a diferença entre a saída do processo e o valor de referência ( $e(k) = y(k) - r(k)$ ). O erro e a sua variação incremental que é dado por,

$$ce(k) = e(k) - e(k-1), \quad (4.6)$$

são as entradas do controlador e a saída é dada por,

$$u(k) = u(k-1) + cu(k). \quad (4.7)$$

As regras de controle, nesse caso, assumem a forma,

$$\text{se } e(k) = A_i \text{ e } ce(k) = B_i \text{ então } cu(k) = C_i. \quad ^1$$

O comportamento do controlador fuzzy pode então ser representado pela seguinte equação:

$$K_3[cu(k)] = F[K_1.e(k), K_2.ce(k)], \quad (4.8)$$

onde  $F$  denota as operações realizadas pelo controlador (transformação escalar-fuzzy, algoritmo de inferência, transformação fuzzy-escalar) sobre a base de conhecimento definida e  $K_i$  ( $i = 1, 2, 3$ ).

Considere as seguintes meta-regras:

1. se  $e(k)$  e  $ce(k)$  são zero, então mantenha a presente ação de controle,
2. se as condições são tais que  $e(k)$  converge a zero numa taxa satisfatória então mantenha a presente ação de controle,
3. se  $e(k)$  não converge a zero, então a variação da ação de controle  $cu(k)$  não é zero e depende do sinal e magnitude de  $e(k)$  e  $ce(k)$ .

Utilizando essas meta-regras MacVicar-Whelan [65] projetou as regras de controle para o *Controlador Fuzzy PI* mostradas na figura 4.1. Os universos de discurso correspondentes a cada variável são particionados em oito conjuntos fuzzy, como observado, onde  $L=large$ ,  $M=medium$ ,  $S=small$  e  $Z=zero$ .

Se o número de regras for levado ao infinito, através do aumento dos níveis de quantização das variáveis de entrada e das variáveis de saída, mantendo-se a simetria em relação à diagonal principal, uma estrutura de controlador PI linear pode ser definida. A figura 4.2 mostra o resultado da adoção de tais hipóteses aplicadas à política de controle. Esse controlador fuzzy torna-se um plano no espaço  $E \times CE \times CU$  e o denominaremos superfície de controle. Na figura 4.2 a superfície de controle é representada através de suas curvas de nível. As variáveis são definidas sobre o mesmo universo de discurso  $[-D, +D]$ .

Assuma que a política de controle possui inclinação  $-C$  (inclinação das curvas de nível no plano  $E \times CE$  correspondentes à superfície de controle) observado na figura 4.2.

<sup>1</sup>é suposto aqui que o controlador não introduz nenhum atraso, ou seja,  $cu(k) = f(e(k), ce(k))$ .



		$ce(k)$							
		-L	-M	-S	-Z	+Z	+S	+M	+L
$e(k)$	+L	-Z	-S	-M	-L	-L	-L	-L	-L
	+M	+S	-Z	-S	-M	-M	-M	-L	-L
	+S	+M	+S	-Z	-S	-S	-S	-M	-L
	+Z	+M	+M	+S	+Z	-Z	-S	-M	-M
	-Z	+M	+M	+S	+Z	-Z	-S	-M	-M
	-S	+L	+M	+S	+S	+S	+Z	-S	-M
	-M	+L	+L	+M	+M	+M	+S	+Z	-S
	-L	+L	+L	+L	+L	+L	+M	+S	+Z

Figura 4.1: Tabela de decisões para o controlador fuzzy PI de MacVicar-Whelan.

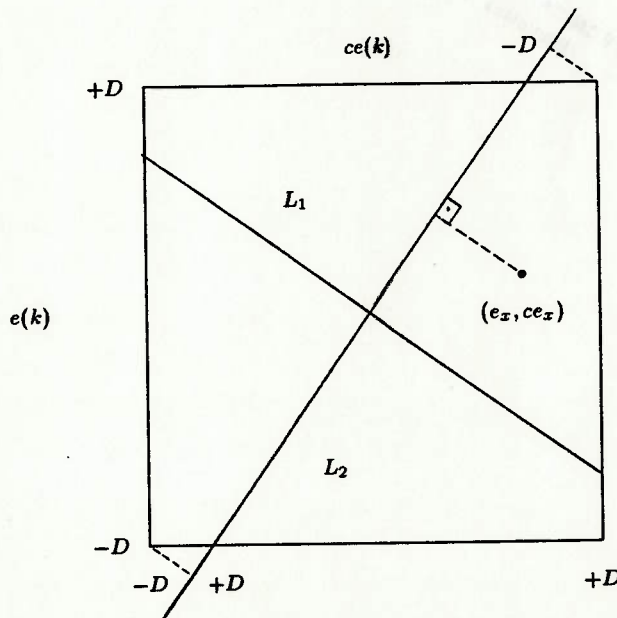


Figura 4.2: Tabela de decisões idealizada para um controlador PI linear, onde a linha  $L_1$  (inclinação  $-C$ ) representa a fronteira de controle e a linha  $L_2$  (inclinação  $1/C$ ) representa a faixa das ações de controle



Para um particular ponto de operação  $(e_x, ce_x)$ , a correspondente ação de controle  $cu_x$  pode ser determinada. Considere a reta com inclinação  $-C$  através do ponto de operação:

$$\frac{e - e_x}{ce - ce_x} = -C. \quad (4.9)$$

A reta perpendicular a 4.9 que passa pela origem é dada por,

$$\frac{e}{ce} = \frac{1}{C}. \quad (4.10)$$

Resolvendo 4.9 e 4.10 tem-se,

$$e = \left( \frac{1}{1 + C^2} \right) (e_x + Cce_x), \quad (4.11)$$

e

$$ce = \left( \frac{C}{1 + C^2} \right) (e_x + Cce_x). \quad (4.12)$$

Portanto, a distância projetada a partir da origem, resulta em

$$(e^2 + ce^2)^{\frac{1}{2}} = \left( \frac{1}{1 + C^2} \right)^{\frac{1}{2}} (e_x + Cce_x). \quad (4.13)$$

Assumindo que a distância é linearmente proporcional a ação de controle

$$cu_x = P \left( \frac{1}{1 + C^2} \right)^{\frac{1}{2}} (e_x + Cce_x), \quad (4.14)$$

onde  $P$  é a constante de proporcionalidade (inclinação da superfície de controle em relação à reta  $L_2$ ). Considerando as constantes  $(K_1, K_2, K_3)$  a equação 4.14 torna-se,

$$\frac{cu_x^*}{K_3} = P \left( \frac{1}{1 + C^2} \right)^{\frac{1}{2}} (K_1 e_x + CK_2 ce_x), \quad (4.15)$$

onde  $cu_x^* = K_3 cu_x$ .

A equação 4.15 corresponde à equação de diferenças de um controlador PI linear dado por,

$$\Delta u(k) = K_P \Delta e(k) + \frac{K_P h}{T_I} e(k), \quad (4.16)$$

logo,

$$K_P = P \left( \frac{1}{1 + C^2} \right)^{\frac{1}{2}} CK_3 K_2, \quad (4.17)$$

$$K_I = P \left( \frac{1}{1 + C^2} \right)^{\frac{1}{2}} \cdot \frac{K_3 K_1}{h} \quad (4.18)$$

As equações 4.17 e 4.18 estabelecem uma relação entre as constantes  $K_P$  e  $K_I$  de um controlador PI linear e as constantes que definem um controlador fuzzy com infinitas regras.

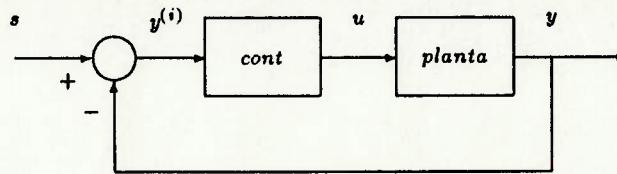


Figura 4.3: sistema de controle fuzzy.

### 4.3.2 Teoremas Limites

Considere um sistema de controle fuzzy, representado na figura 4.3, com uma saída  $y(t)$ , uma entrada  $u(t)$ , e um valor de referência  $s$ . O erro do sistema é dado por  $y(t) - s$  e a  $i$ -ésima derivada do erro é indicada por  $y^{(i)}(t)$  <sup>2</sup> para  $0 \leq i \leq n$ .

Constantes  $c_i$  são escolhidas de tal forma que  $-1 \leq c_i y^{(i)}(t) \leq 1$  (normalização). Seja  $p^{(i)}(t) = c_i y^{(i)}(t)$ ,  $0 \leq i \leq n$ . Nas expressões que se seguem assume-se que a entrada do controlador seja dada por  $p^{(i)}(t)$ ,  $0 \leq i \leq n$ , cujos valores estão sempre no intervalo  $[-1, 1]$ .

A saída normalizada do controlador fuzzy será denotada por  $\delta(t)$ , cujos valores também pertencem ao intervalo  $[-1, 1]$  para  $t \geq 0$ . Seja  $\Delta$  o período de amostragem e  $k\Delta$  o instante em que se observa  $p^{(i)}(t)$  e se infere  $\delta(t)$ . Sob tais considerações o esforço de controle será dado por,

$$u(t) = u(t - \Delta) + (\delta(t))(\Delta) \quad (4.19)$$

$$t = \Delta, 2\Delta, \dots$$

Buckley e Ying [19] dividem o controlador fuzzy em três partes principais:

1. o processo de transformação escalar-fuzzy, ou os números fuzzy utilizados sobre cada universo de discurso,
2. as regras de controle fuzzy e o procedimento empregado para avaliar tais regras,
3. o algoritmo de transformação fuzzy-escalar.

A seguir serão discutidos cada um dos componentes

#### Números Fuzzy

Devem ser definidos  $2N + 1$ ,  $N \geq 1$ , números fuzzy para cada uma das  $n + 1$  entradas  $p^{(i)}$ .  $P_j^{(i)}$  denotará o número fuzzy associado a  $p^{(i)}$ , para  $1 \leq j \leq 2N + 1$ . O valor central<sup>3</sup> de  $P_j^{(i)}$  será dado por,  $v(j) = [j - (N + 1)]/N$ .

<sup>2</sup>nas expressões que se seguem o índice ( $i$ ) se refere a grandezas relacionadas a  $i$ -ésima derivada

<sup>3</sup>ponto de simetria da função de pertinência que define o número fuzzy

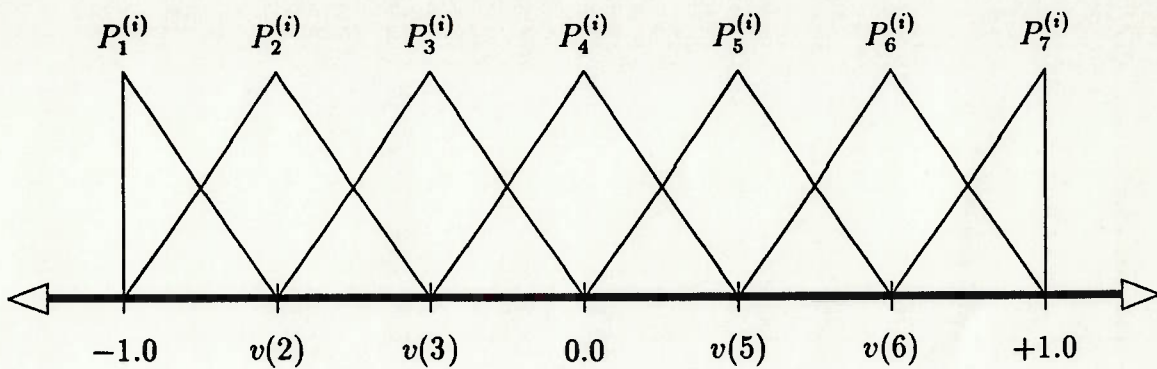


Figura 4.4: Números fuzzy triangulares para  $2N + 1 = 7$ .

A função de pertinência de  $P_j^{(i)}$ ,  $z = \mu(x|P_j^{(i)})$ , possuirá as seguintes características para  $2 \leq j \leq 2N$ ,

1. assume valor zero fora dos limites de  $(v(j - 1), v(j + 1))$ ,
2. assume valor unitário em  $v(j)$ ,
3. é uma função positiva contínua não-decrescente em  $x$  sobre o intervalo  $(v(j - 1), v(j))$  e,
4. é uma função positiva contínua não-crescente em  $x$  sobre o intervalo  $[v(j), v(j + 1))$

Por exemplo,  $z = \mu(x|P_j^{(i)})$  pode ser uma função triangular, trapezoidal, normal (gaussiana) sobre  $(v(j - 1), v(j + 1))$  para  $2 \leq j \leq 2N$ .

$z = \mu(x|P_1^{(i)})$  é uma função positiva contínua não-crescente de  $x$ , assumindo o valor 1 em  $x = -1$  e 0 para  $x \geq v(2)$ .

$z = \mu(x|P_{2N+1}^{(i)})$  é uma função positiva contínua não-decrescente de  $x$  sobre  $(v(2N), 1]$ , assumindo o valor 0 para  $x \leq v(2N)$  e 1 em  $x = 1$ . Todos os números fuzzy possuem o seu suporte em  $[-1, 1]$ .

Dado um valor de  $p^{(i)}$  em  $[-1, 1]$  existe apenas um  $j$  em  $1, 2, \dots, 2N$ , digamos  $j(i)$ , tal que  $p^{(i)}$  se encontra no intervalo  $[v(j(i)), v(j(i) + 1))$  se  $1 \leq j(i) \leq 2N$  ou  $p^{(i)}$  se encontra em  $[1 - N^{-1}, 1]$  se  $j(i) = 2N$ .

Seja  $\mu_j^{(i)} = \mu(x|P_j^{(i)})$ , para cada  $i$  apenas dois  $\mu_j^{(i)}$  podem ser diferentes de zero e esses ocorrem quando  $j = j(i)$  e  $j = j(i) + 1$ .

Um exemplo de números fuzzy triangulares que observam as hipóteses acima, pode ser visto na figura 4.4.

### Regras

As regras de controle serão denotadas por  $R_k$  com  $1 \leq k \leq K = (n + 1)(2N) + 1$ . Seja  $\varphi^{(i)}$  uma variável fuzzy relativa a  $p^{(i)}$ . Por exemplo,  $\varphi^{(0)}$  pode ser o erro e  $\varphi^{(1)}$  pode ser a variação do erro.

Dessa forma, a primeira e a última regra tornam-se:

$R_1$ : se  $[\varphi^{(0)} = P_1^{(0)}]$  e ... e  $[\varphi^{(n)} = P_1^{(n)}]$  então  $O = O_K$ ,  
 $R_K$ : se  $[\varphi^{(0)} = P_{2N+1}^{(0)}]$  e ... e  $[\varphi^{(n)} = P_{2N+1}^{(n)}]$  então  $O = O_1$ .

onde  $O$  denota saída. A saída assume como valores os conjuntos fuzzy  $O_l, 1 \leq l \leq K$ .  
 Os valores centrais de  $O_l$  serão denotados por,

$$w(l) = \frac{l - [(n + 1)N + 1]}{(n + 1)N}. \quad (4.20)$$

A função de pertinência  $z = \mu(x|O_l)$  assume as mesmas características de  $z = \mu(x|P_j^{(i)})$ .

Se  $\Psi_j, 1 \leq j \leq J$  denota uma cláusula tal como  $[\varphi^{(i)} = P_j^{(i)}]$ , então  $\text{ou}\{\Psi_j | 1 \leq j \leq J\}$  se refere a,

$$\Psi_1 \text{ ou } \Psi_2 \text{ ou } \dots \text{ ou } \Psi_J. \quad (4.21)$$

As demais regras de controle são definidas como,

$R_k$  :  $\text{ou}\{[\varphi^{(0)} = P_{i_0}^{(0)}]$  e ... e  $[\varphi^{(n)} = P_{i_n}^{(n)}] \mid i_0 + \dots + i_n = n + k\}$ ,  
 então  $O = O_{K-k+1}$

para  $1 < k < K$ . As regras acima são denominadas *lineares* porque o conectivo *ou* afeta todas as regras cujas cláusulas possuem números fuzzy com soma constante,  $i_0 + \dots + i_n = n + k$ . Também deve-se notar que cada regra de controle possui uma conclusão distinta. Existem  $K$  regras de controle fuzzy e  $O$  possui  $K$  números fuzzy distintos e equiespaçados sobre  $[-1, 1]$ .

Para avaliar uma regra, dado  $p^{(i)}$ , deve-se inicialmente avaliar as cláusulas  $[\varphi^{(i)} = P_j^{(i)}]$ . O valor de  $[\varphi^{(i)} = P_j^{(i)}]$  é  $\mu_j^{(i)} = \mu(p^{(i)}|P_j^{(i)})$ .

Seja  $T$  qualquer norma triangular extendida (por associatividade) a  $n+1$  argumentos e seja  $C$  qualquer co-norma triangular extendida (por associatividade) a  $n+1$  argumentos. Para o caso especial em que  $n = 0$ ,  $T(\mu) = \mu$  e  $C(\mu) = \mu$ . Por exemplo [20],  $C$  pode ser um *ou* de LuKasiewicz (soma limitada),

$$C(x_0, \dots, x_n) = \min \left( \sum_{i=0}^n x_i, 1 \right), \quad (4.22)$$

e  $T$  pode ser um *e* probabilístico (produto algébrico),

$$T(x_0, \dots, x_n) = \prod_{i=0}^n x_i. \quad (4.23)$$

Quando  $R_k$  é executada, o valor inferido é dado por  $\Gamma_k$ ,

$$\Gamma_1 = T(\mu_1^{(0)}, \dots, \mu_1^{(n)}), \quad (4.24)$$

$$\Gamma_K = T(\mu_{2N+1}^{(0)}, \dots, \mu_{2N+1}^{(n)}), \quad (4.25)$$



e

$$\Gamma_k = C\{T(\mu_{i_0}^{(0)}, \dots, \mu_{i_n}^{(n)}) \mid i_0 + \dots + i_n = n + k, \} \quad (4.26)$$

onde  $1 < k < K$ .

### Transformação Fuzzy Escalar

O primeiro método de transformação fuzzy escalar utiliza o valor central de cada número fuzzy em  $O$ ,

$$\delta_0 = \frac{\sum_{l=1}^K \Gamma_l w(K-l+1)}{\sum_{l=1}^K \Gamma_l} \quad (4.27)$$

Seja,

$$Q_l = \frac{\Gamma_l}{\sum_{l=1}^K \Gamma_l}, \quad (4.28)$$

para  $1 \leq l \leq K$ . Os valores  $Q_l$  podem ser considerados probabilidades definidas sobre os valores centrais de  $O_l$ , dessa forma,  $\delta_0$  pode ser dado por,

$$\delta_0 = \sum_{l=1}^K Q_l w(K-l+1). \quad (4.29)$$

O segundo procedimento utiliza um número  $\alpha_l$  em  $[w(l-1), w(l+1)]$  se  $2 \leq l \leq K-1$ , ou  $\alpha_1$  em  $[-1, w(2)]$ , ou  $\alpha_K$  em  $[w(K-1), 1]$ , tal que  $\mu(\alpha_l | O_l) = \Gamma$  em lugar do valor central utilizado na equação 4.27. Esse método permite dois valores possíveis para  $\alpha_l$ ,  $1 \leq l \leq K-1$ , um à direita do valor central e um à esquerda do valor central. Para os resultados a serem obtidos aqui isso torna-se indiferente.  $\delta_1$  é então definido como,

$$\delta_1 = \frac{\sum_{l=1}^K \alpha_l \Gamma_l}{\sum_{l=1}^K \Gamma_l} = \sum_{l=1}^K P_l \alpha_l. \quad (4.30)$$

Seja,

$$\ell = - \sum_{i=0}^n p^{(i)} / (n+1). \quad (4.31)$$

Portanto,  $\ell$ , é uma combinação linear das entradas  $p^{(i)}$ ,  $0 \leq i \leq n$ .

#### Teorema 4.1

$$\lim_{N \rightarrow \infty} |\delta_0 - \ell| = 0. \quad (4.32)$$

#### Teorema 4.2

$$\lim_{N \rightarrow \infty} |\delta_1 - \ell| = 0. \quad (4.33)$$

As demonstrações detalhadas de tais teoremas se encontram em [19]. Esse resultado, pode ser estendido [19] para um operador  $\delta_2$  de transformação fuzzy escalar mais genérico e que engloba as definições de  $\delta_0$ ,  $\delta_1$  além do *Método do Critério Máximo* e o *Método do Centro de Gravidade*.

Conclui-se então que ao se utilizar *regras de controle lineares*, para qualquer tipo de números fuzzy unimodais<sup>4</sup> utilizados, para qualquer algoritmo de inferência fuzzy utilizado, para qualquer operador de transformação fuzzy escalar utilizado (que pertença à definição de  $\delta_2$ ), a saída do controlador se aproxima de uma função linear de suas entradas a medida que o número de regras cresce.

Esse resultado foi estendido por Buckley em [20] para controladores com múltiplas entradas e múltiplas saídas (MIMO).

## 4.4 Controladores Fuzzy Lineares

Como um caso particular, Buckley e Ying [20] demonstraram que um controlador fuzzy pode assumir um comportamento linear desde que se utilize,

1. uma operação de transformação escalar-fuzzy linear,
2. regras de controle fuzzy lineares,
3. e uma operação de transformação fuzzy-escalar linear.

### 4.4.1 Números Fuzzy

Como no caso anterior devem ser utilizados  $2N+1$  números fuzzy para cada uma das  $n+1$  entradas  $p^{(i)}$ . Nesse caso a função de pertinência  $z = \mu(x|P_j^{(i)})$  é uma função triangular sobre  $(v(j-1), v(j+1))$  para  $2 \leq j \leq 2N$ .  $z = \mu(x|P_1^{(i)})$  é um segmento de reta de  $(-1, 1)$  a  $(-1 + N^{-1}, 0)$  sobre  $[-1, -1 + N^{-1}]$  e zero caso contrário.  $z = \mu(x|P_{2N+1}^{(i)})$  é um segmento de reta de  $(1 - N^{-1}, 0)$  até  $(1, 1)$  e zero caso contrário.

Um exemplo de números fuzzy triangulares pode ser visto na figura 4.4.

Dado um valor para  $p^{(i)}$  em  $[-1, 1]$  existe um único valor  $j$  em  $\{1, \dots, 2N\}$ , digamos  $j(i)$ , tal que  $p^{(i)}$  está em  $(v(j(i)), v(j(i)+1))$  se  $1 \leq j(i) < 2N$ , ou  $p^{(i)}$  está em  $[1 - N^{-1}, 1]$  se  $j(i) = 2N$ .

Seja  $\mu_j^{(i)} = \mu(p^{(i)}|P_j^{(i)})$ . A partir do que foi exposto acima, para  $0 \leq i \leq n$ , temos,

$$\mu_j^{(i)} = \begin{cases} \alpha_{j(i)}^{(i)} & \text{se } j = j(i), \\ 1 - \alpha_{j(i)}^{(i)} & \text{se } j = j(i) + 1, \\ 0 & \text{caso contrário,} \end{cases} \quad (4.34)$$

onde

$$\alpha_{j(i)}^{(i)} = j(i) - N(p^{(i)} + 1). \quad (4.35)$$

<sup>4</sup>números fuzzy com as características já mencionadas

Isso mostra que  $\mu_j^{(i)}$  é uma função linear da entrada  $p^{(i)}$ , e dessa forma conclue-se que a operação de transformação escalar-fuzzy é linear.

#### 4.4.2 Regras

As regras empregadas são as regras de controle lineares definidas na seção 4.3.2 acima.

A avaliação das regras é feita utilizando-se o **ou** de *Lukasiewicz* (soma limitada) (C=LOR) como co-norma triangular e o **e** *probabilístico* (produto algébrico) (T=PAND) como norma triangular.

#### 4.4.3 Transformação Escalar-Fuzzy

A operação de transformação escalar-fuzzy linear é dada por,

$$\delta = \sum_{l=1}^K \Gamma_l w(K-l+1). \quad (4.36)$$

onde  $w(l)$  é o valor central para o conjunto  $O_l$ .

**Teorema 4.3** *Se T=PAND e C=LOR, então  $\delta = -\frac{1}{n+1} \sum_{i=0}^n r^{(i)}$ . Ou seja, sob as hipóteses consideradas acima a saída  $\delta$  inferida, torna-se uma combinação linear das entradas<sup>5</sup>.*

### 4.5 Estudo de Caso: Sistema Linear de Primeira-Ordem

O objetivo nesse ítem é demonstrar que um controlador fuzzy construído adequadamente pode ter um comportamento aproximadamente linear.

A planta escolhida como objeto de controle é um sistema de primeira-ordem linear cuja função de transferência pode ser representada por,

$$\frac{Y(s)}{U(s)} = \frac{1}{s+2}. \quad (4.37)$$

Inicialmente, procurou-se sintonizar um controlador PI linear para a planta dada por 4.37. Através de simulações foram escolhidos os parâmetros ( $K_P = 2.0, T_I = K_P/K_I = 0.35$ ), que resultam num desempenho dinâmico para uma resposta a degrau que pode ser observado na tabela 4.1.

O controlador fuzzy escolhido é do tipo Proporcional-Integral, ou seja, as variáveis escolhidas são:  $e, ce$  e  $cu$ . A figura 4.5 apresenta o diagrama de blocos do sistema.

O comportamento global do controlador fuzzy pode ser representado por,

$$K_{cu}cu = f(K_e e, K_{ce} ce). \quad (4.38)$$

<sup>5</sup>A demonstração detalhada desse teorema se encontra em [20].



tempo de subida (s) (0-100%)	sobresinal	tempo de acomodação em 5% (s)
1.1	0.02	0.9

Tabela 4.1: desempenho dinâmico do controlador PI.

Como foi demonstrado na seção 4.3.1, a relação entre um controlador PI linear e um controlador fuzzy PI linear é dada por,

$$K_P = P \left( \frac{1}{1 + C^2} \right)^{\frac{1}{2}} C K_{cu} K_{ce}, \quad (4.39)$$

$$K_I = P \left( \frac{1}{1 + C^2} \right)^{\frac{1}{2}} \cdot \frac{K_{cu} K_e}{h} \quad (4.40)$$

Vamos supor que um controlador fuzzy linear pode ser construído através de um conjunto de regras lineares segundo o conceito de Buckley e Ying (veja seção 4.3.2) com  $2N + 1 = 5$  números fuzzy para as variáveis de entrada (*NL: Negative-Large*, *NS: Negative-Small*, *ZE: Zero*, *PS: Positive-Small*, *PL: Positive-Large*) e  $K = 9$  números fuzzy para a variável de saída (*NL: Negative-Large*, *NM: Negative-Medium*, *NS: Negative-Small*, *NVS: Negative-Very-Small*, *ZE: Zero*, *PVS: Positive-Very-Small*, *PS: Positive-Small*, *PM: Positive-Medium*, *PL: Positive-Large*).

As figuras 4.6 e 4.7 ilustram os números fuzzy definidos e a tabela 4.4 representa as regras de controle fuzzy. As tabelas 4.2-4.3 apresentam as definições quantitativas correspondentes, incluindo a normalização dos espaços considerados <sup>6</sup>.

Para esse controlador fuzzy linear os parâmetros  $C$  e  $P$  que se referem ao núcleo do controlador, ou seja, não considerando as constantes ( $K_{cu}$ ,  $K_e$ ,  $K_{ce}$ ), valem respectivamente,  $C = 1.0$  e  $P = \sqrt{2}/2$ . Como existem duas equações e três constantes a serem determinadas podemos escolher  $K_{cu} = 2.0$ , o que resulta em,  $K_e = 0.2857$  e  $K_{ce} = 2.0$ , considerando um tempo de amostragem  $h = 0.05$ seg. As equações 4.39 e 4.40 foram deduzidas sem considerar a normalização dos universos de discurso, entretanto, tais equações também são válidas para esse caso porque a normalização ocorre sobre o mesmo intervalo  $([-4.0, +4.0])$  para as variáveis  $e$ ,  $ce$  e  $cu$ .

A figura 4.8 ilustra a resposta a degrau do sistema considerando o controlador PI linear e o controlador fuzzy com regras lineares considerando três métodos distintos:

1. Algoritmo de Mandani com o centro de gravidade como método de transformação fuzzy-escalar,

<sup>6</sup>nos experimentos realizados no decorrer desse trabalho a normalização dos universos de discurso ocorre de maneira uniforme, ou seja, um único segmento  $[a, b]$  corresponde ao universo normalizado  $[-1.0, +1.0]$ . Os números fuzzy utilizados são do tipo triangular ou trapezoidal para o caso do número mais negativo ou mais positivo.  $\alpha_p$  e  $\alpha_b$  são respectivamente o ponto de máximo da função triangular e a base do triângulo isósceles correspondente. Os números fuzzy mais negativo e mais positivo correspondem à metade desse triângulo isósceles.



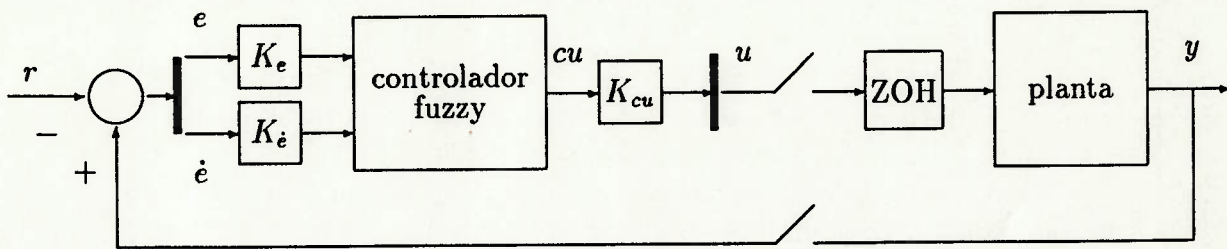


Figura 4.5: estrutura de controle para sistema de primeira-ordem.

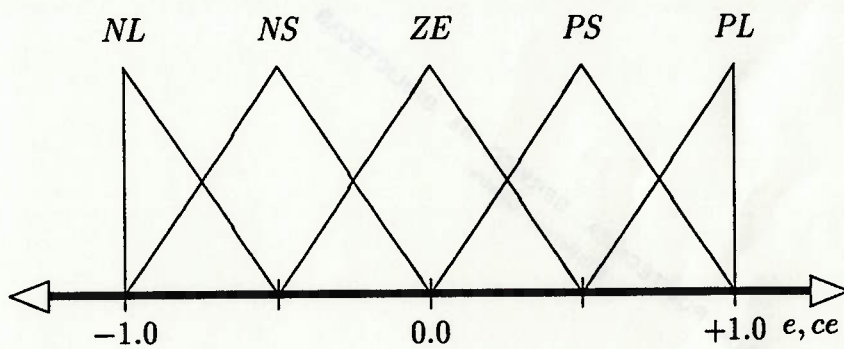


Figura 4.6: números fuzzy para as variáveis e e ce.

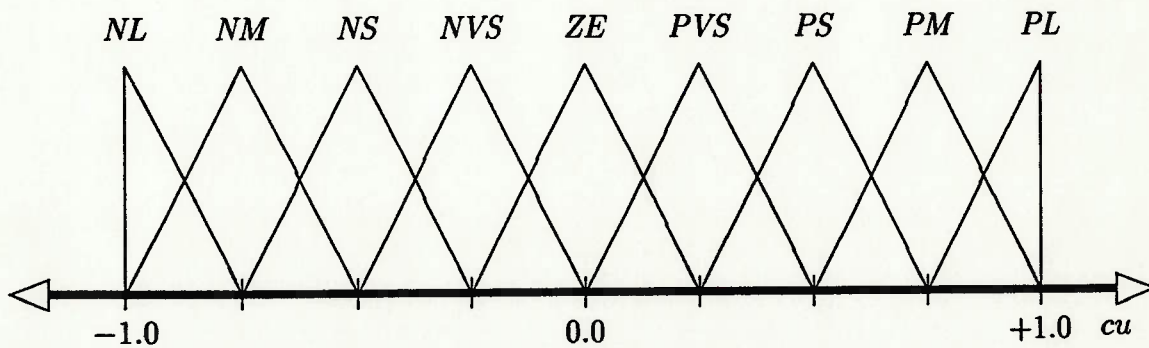


Figura 4.7: números fuzzy para a variável cu.

intervalos	segmentos		$\alpha_p$	$\alpha_b$	números fuzzy
	normalizados				
[-4.0, +4.0]	[-1.0, +1.0]		-1.0	1.0	NL
			-0.5	1.0	NS
			0.0	1.0	ZE
			+0.5	1.0	PS
			+1.0	1.0	PL

Tabela 4.2: normalização e funções de pertinência para as variáveis  $e$  e  $ce$ . Veja nota de rodapé na página 59.

intervalos	segmentos		$\alpha_p$	$\alpha_b$	números fuzzy
	normalizados				
[-4.0, +4.0]	[-1.0, +1.0]		-1.0	0.5	NL
			-0.75	0.5	NM
			-0.5	0.5	NS
			-0.25	0.5	NVS
			0.0	0.5	ZE
			+0.25	0.5	PVS
			+0.5	0.5	PS
			+0.75	0.5	PM
			+1.0	0.5	PL

Tabela 4.3: normalização e funções de pertinência da variável  $cu$ . Veja nota de rodapé na página 59.

$e \setminus ce$	NL	NS	ZE	PS	PL
PL	ZE	NVS	NS	NM	NL
PS	PVS	ZE	NVS	NS	NM
ZE	PS	PVS	ZE	NVS	NS
NS	PM	PS	PVS	ZE	NVS
NL	PL	PM	PS	PVS	ZE

Tabela 4.4: tabela de decisões  $(e, ce) \rightarrow cu$ .

intervalos	segmentos normalizados	$\alpha_p$	$\alpha_b$	números fuzzy
		-1.0	0.2858	$E_1$
		-0.857	0.2858	$E_2$
		-0.714	0.2858	$E_3$
		⋮	⋮	⋮
		-0.143	0.2858	$E_7$
[-4.0, +4.0]	[-1.0, +1.0]	0.0	0.2858	$E_8$
		+0.143	0.2858	$E_9$
		⋮	⋮	⋮
		+0.714	0.2858	$E_{13}$
		+0.857	0.2858	$E_{14}$
		+1.0	0.2858	$E_{15}$

Tabela 4.5: normalização e funções de pertinência para as variáveis  $e$  e  $ce$ . Veja nota de rodapé na página 59.

2. Método de Buckley e Ying com  $\delta_0$  como método de transformação fuzzy-escalar (seção 4.3.2),
3. Método Simplificado (seção 2.9).

Pode ser observado que o desempenho do controlador fuzzy com regras lineares é equivalente ao do controlador PI linear. Um exame mais detalhado a respeito da linearidade de tais controladores pode ser realizado observando-se a *superfície de controle*  $cu = f(e, ce)$  e as *curvas de nível* para cada um dos algoritmos adotados nas figuras 4.9-4.11.

A superfície de controle para um controlador PI é um plano no espaço tridimensional. Qualquer um dos três algoritmos possui aproximadamente um plano como superfície de controle, entretanto, o método de Buckley e Ying é o que se aproxima mais de um plano perfeito. O algoritmo de Mandani e o método simplificado possuem não linearidades como pode ser observado nas figuras 4.9 e 4.11.

Com o intuito de se comprovar que um aumento do número de regras provoca um aumento da linearidade do controlador fuzzy vamos construir um controlador com  $2N + 1 = 15$  números fuzzy para as variáveis de entrada e  $K = 29$  números fuzzy para a variável de saída, totalizando portanto,  $15 \times 15 = 225$  regras. As tabelas 4.5 e 4.6 apresentam as definições quantitativas para as funções de pertinência e a tabela 4.7 apresenta o conjunto de regras adotado.

A figura 4.12 mostra resposta a degrau para o controlador fuzzy considerando o algoritmo de Mandani e a figura 4.13 apresenta a superfície de controle e as curvas de nível correspondentes.

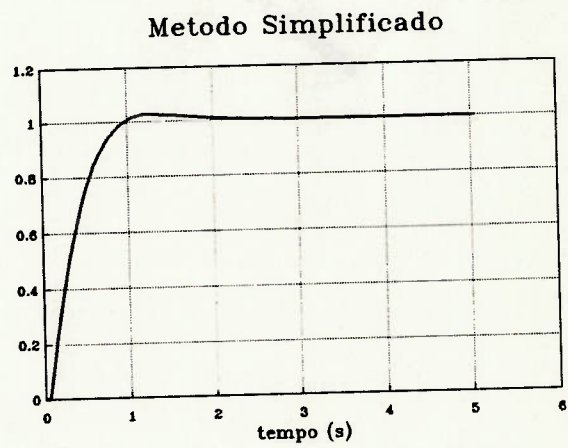
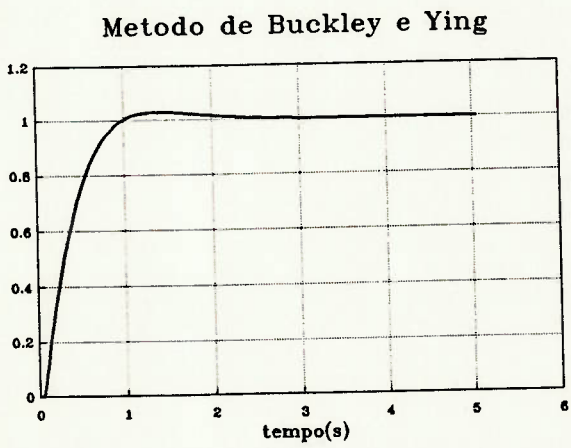
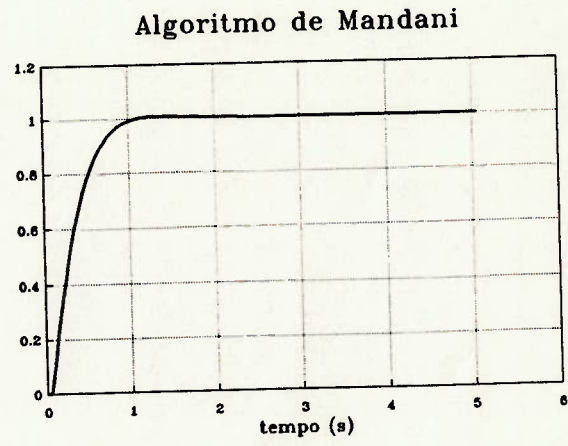
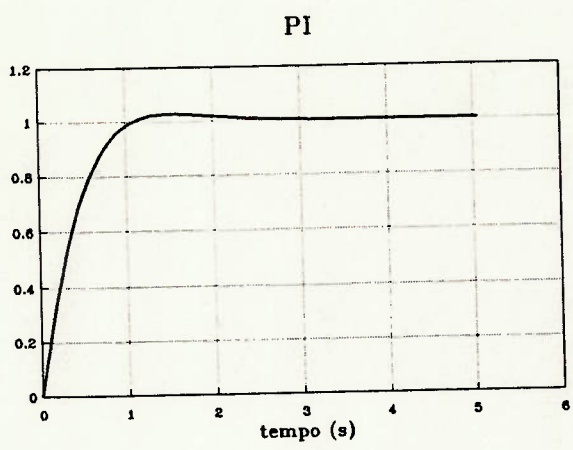


Figura 4.8: resposta a degrau.



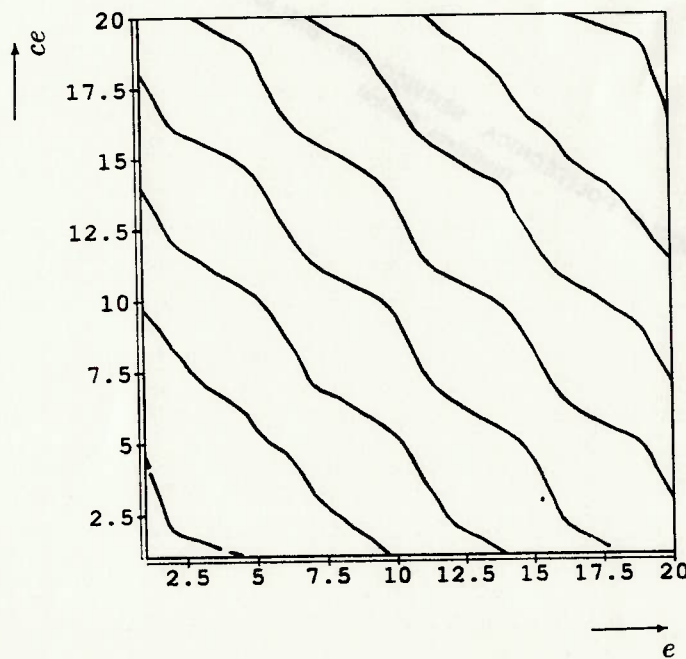
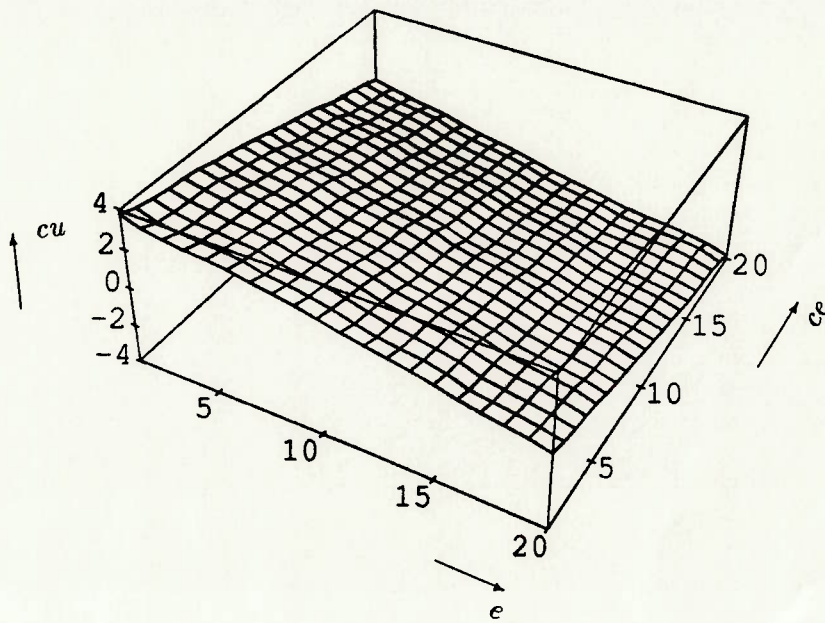


Figura 4.9: superfície de controle e curvas de nível para o algoritmo de Mandani. O intervalo  $[0, 20]$  indicado nos eixos correspondentes às variáveis  $e$  e  $ce$  se refere ao intervalo  $[-4.0, +4.0]$ . A escala referente ao eixo  $cu$  corresponde a valores reais.

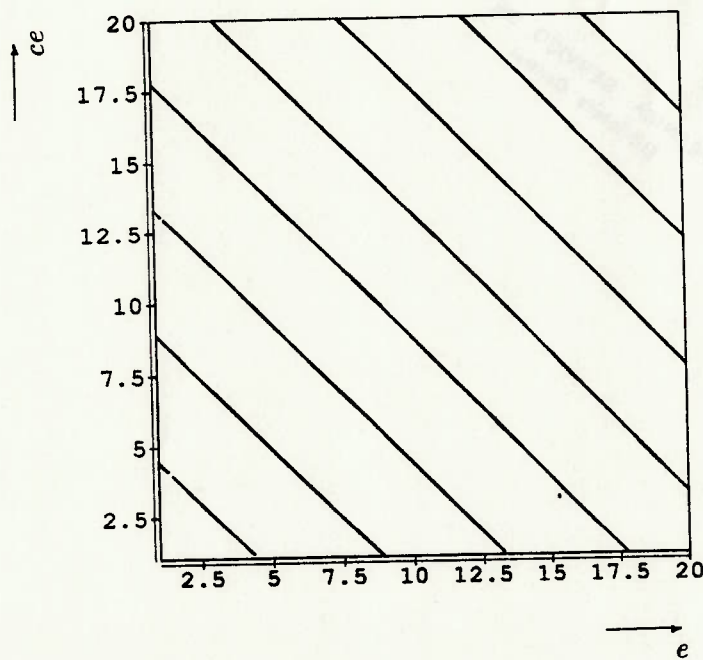
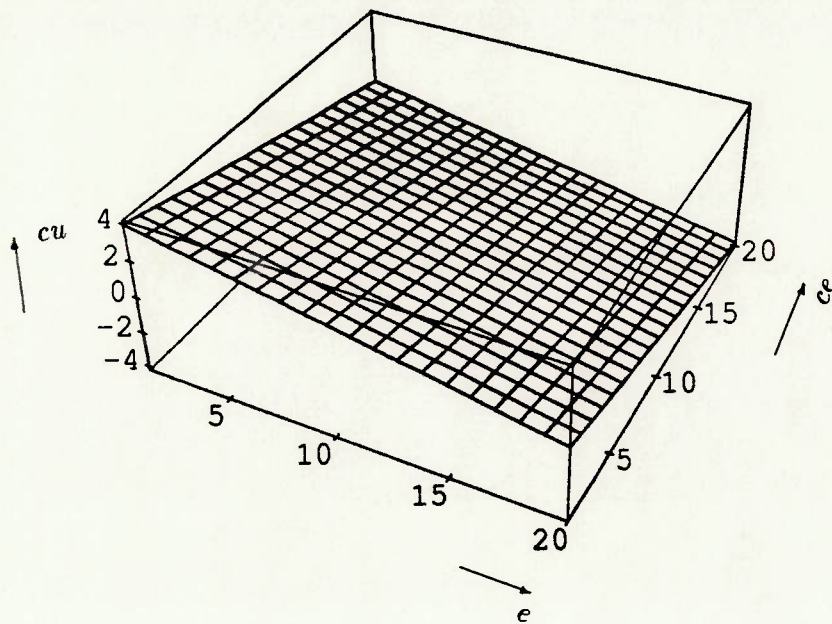


Figura 4.10: superfície de controle e curvas de nível para o método de Buckley e Ying. O intervalo  $[0, 20]$  indicado nos eixos correspondentes às variáveis  $e$  e  $ce$  se refere ao intervalo  $[-4, +4]$ . A escala referente ao eixo  $cu$  corresponde a valores reais.

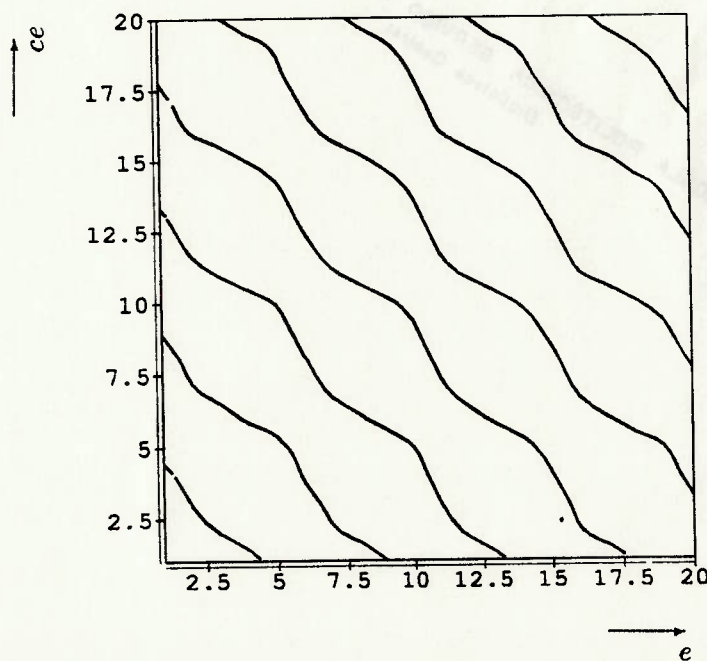
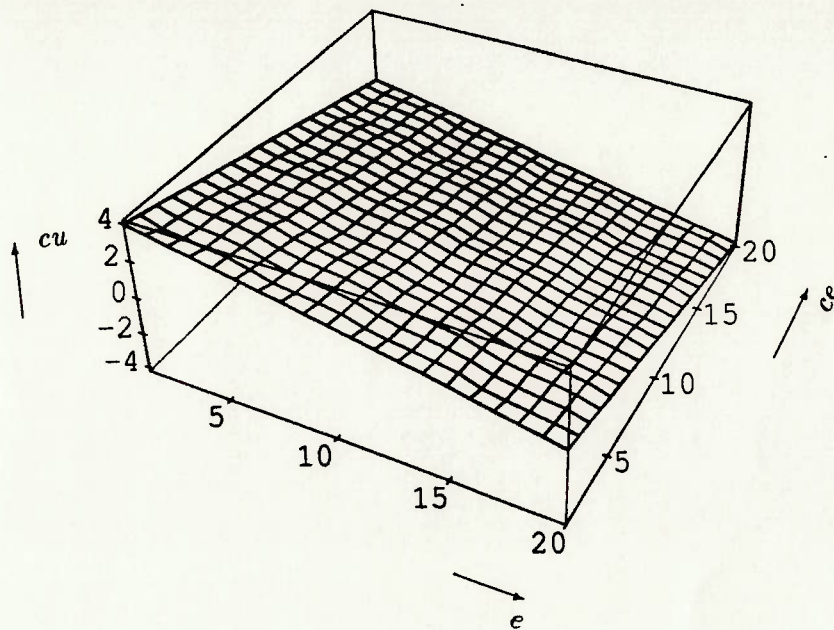


Figura 4.11: superfície de controle e curvas de nível para o método simplificado. O intervalo  $[0, 20]$  indicado nos eixos correspondentes às variáveis  $e$  e  $ce$  se refere ao intervalo  $[-4.0, +4.0]$ . A escala referente ao eixo  $cu$  corresponde a valores reais.

intervalos	segmentos normalizados	$\alpha_p$	$\alpha_b$	números fuzzy
[-4.0, +4.0]	[-1.0, +1.0]	-1.0	0.2858	$O_1$
		-0.857	0.2858	$O_2$
		-0.714	0.2858	$O_3$
		$\vdots$	$\vdots$	$\vdots$
		-0.143	0.2858	$O_{14}$
		0.0	0.2858	$O_{15}$
		+0.143	0.2858	$O_{16}$
		$\vdots$	$\vdots$	$\vdots$
		+0.714	0.2858	$O_{27}$
		+0.857	0.2858	$O_{28}$
		+1.0	0.2858	$O_{29}$

Tabela 4.6: normalização e funções de pertinência para variável *cu*. Veja nota de rodapé na página 59.

$e \setminus ce$	$E_1$	$E_2$	...	$E_7$	...	$E_{14}$	$E_{15}$
$E_{15}$	$O_{14}$	$O_{13}$	...	$O_8$	...	$O_1$	$O_0$
$E_{14}$	$O_{15}$	$O_{14}$	...	$O_9$	...	$O_2$	$O_1$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$		$\vdots$	$\vdots$
$E_7$	$O_{21}$	$O_{20}$	...	$O_{14}$	...	$O_8$	$O_7$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\ddots$	$\vdots$	$\vdots$
$E_2$	$O_{27}$	$O_{26}$	...	$O_{20}$	...	$O_{14}$	$O_{13}$
$E_1$	$O_{28}$	$O_{27}$	...	$O_{21}$	...	$O_{15}$	$O_{14}$

Tabela 4.7: tabela de decisões.



### Algoritmo de Mandani

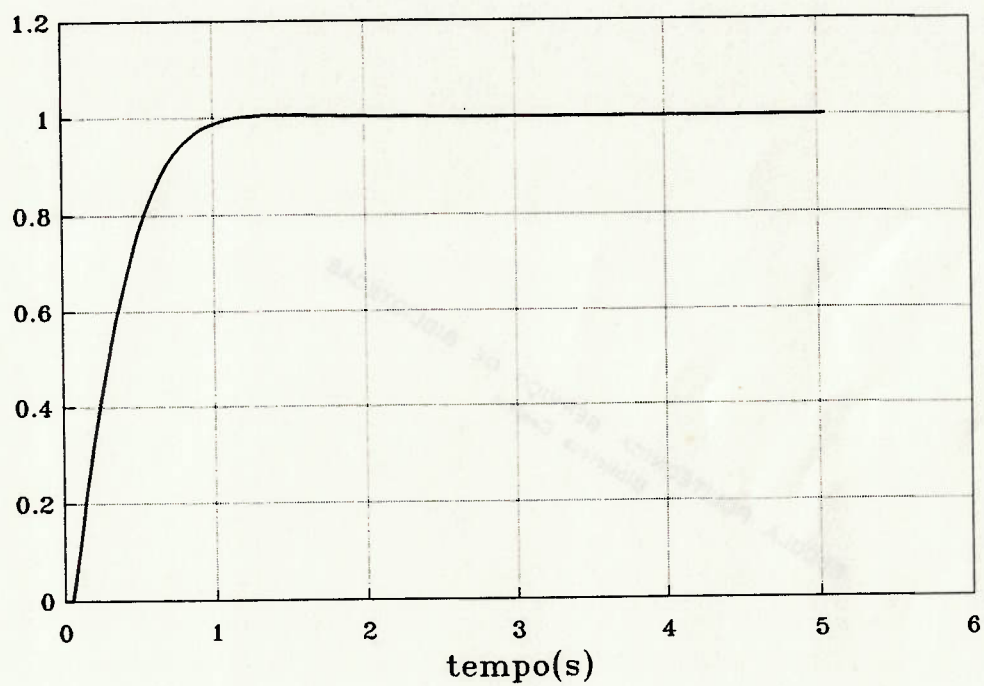


Figura 4.12: resposta a degrau (Algoritmo de Mandani).

## 4.6 Conclusão

Nesse capítulo foi estabelecido uma relação entre controladores fuzzy e controladores lineares do tipo PID. A escolha apropriada das variáveis de estado e das variáveis de controle utilizadas nas regras determinam o comportamento do controlador fuzzy e corresponde à identificação estrutural do sistema.

Se uma base de conhecimento for adequadamente construída, a medida que o número de regras cresce, o comportamento do controlador fuzzy torna-se cada vez mais linear. No limite, com o número de regras tendendo a infinito, obtem-se um controlador linear. Tang e Mulholland desenvolveram uma interpretação geométrica para tal comportamento limite utilizando um controlador fuzzy do tipo PI. Buckley e Ying estabeleceram teoremas limites para sistemas MISO.

Os resultados apresentados na seção 4.5, onde uma planta de primeira-ordem é controlada por um controlador fuzzy PI, comprovam tais afirmações. O desempenho dinâmico obtido pelos controladores fuzzy de 25 regras é bastante semelhante ao obtido pelo controlador linear. Um aumento do número de regras para 225 torna a superfície de controle bastante linear.

Sob o ponto de vista prático a utilização de controladores fuzzy lineares ou aproximadamente lineares não é viável devido ao elevado número de parâmetros envolvidos. Entretanto os resultados aqui apresentados permitem explicar muitos dos bons resultados obtidos através de controladores fuzzy do tipo PI.

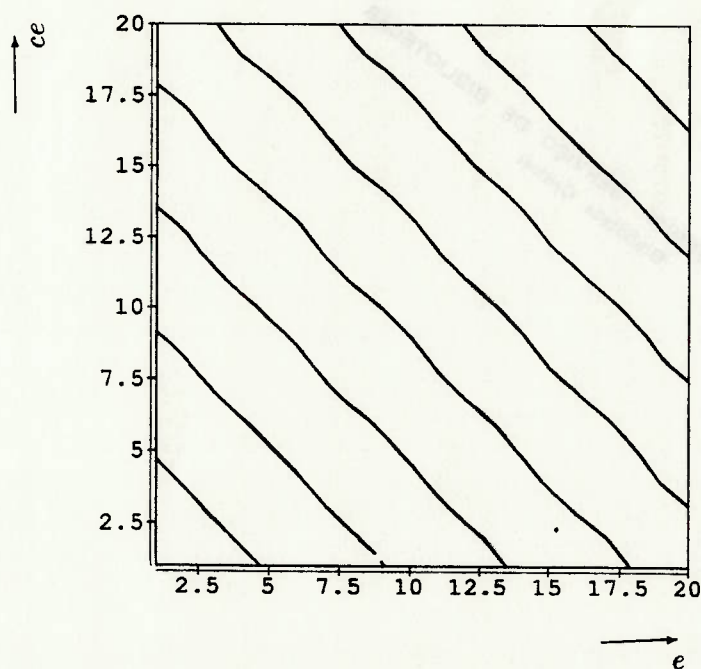
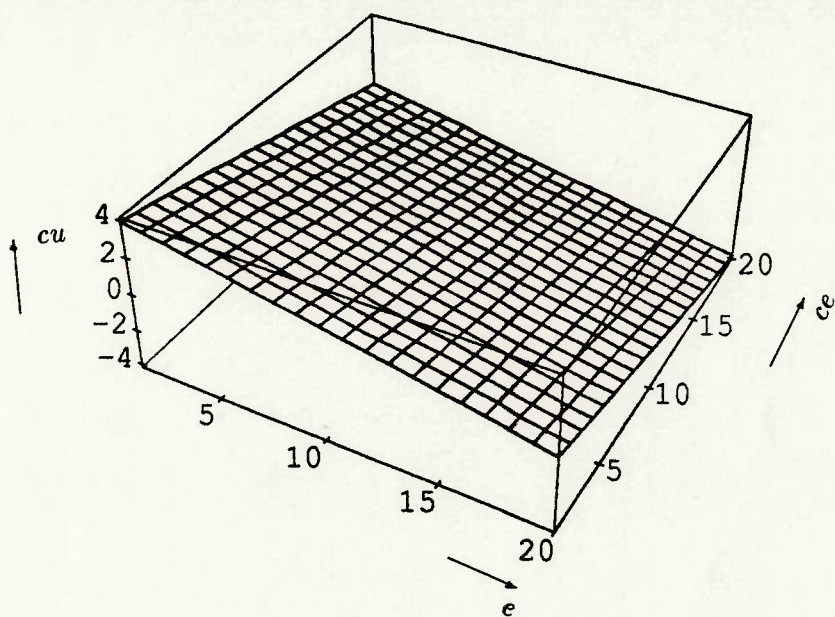


Figura 4.13: superfície de controle e curvas de nível para o Algoritmo de Mandani (225 regras). O intervalo  $[0, 20]$  indicado nos eixos correspondentes às variáveis  $e$  e  $ce$  se refere ao intervalo  $[-4.0, +4.0]$ . A escala referente ao eixo  $cu$  corresponde a valores reais

## Capítulo 5

# Não Linearidades em Controladores Fuzzy

### 5.1 Introdução

No capítulo 4 foi demonstrado que um controlador fuzzy pode na prática assumir um comportamento linear ou aproximadamente linear desde que construído adequadamente. Entretanto, a utilização de controladores fuzzy como um controlador linear é inadequado devido a sua complexidade de construção. A grande vantagem de controladores fuzzy está na possibilidade de se gerar superfícies de controle não lineares baseadas em conhecimento heurístico para o controle de plantas não lineares cujo modelo matemático pode ser conhecido ou não.

No estágio atual o perfeito domínio da influência de cada uma das partes que compõem um controlador fuzzy ainda não foi obtida. A experiência prévia ainda é o fator dominante.

Este capítulo está dividido em três partes: a seção 5.2 enuncia as partes principais que compõem um controlador fuzzy e sua influência sobre as não linearidades globais obtidas. As seções 5.3 e 5.4 apresentam como exemplos, respectivamente, o projeto de controladores fuzzy para um sistema de primeira-ordem com tempo morto e para um pêndulo invertido.

### 5.2 Fontes de Não-Linearidades em Controladores Fuzzy

Como já pode ser observado, as não-linearidades podem ocorrer em vários pontos da estrutura de um controlador fuzzy:

**algoritmo de transformação escalar-fuzzy:** está relacionado à forma (triangular, gaussiana, trapezoidal, etc.) e a distribuição das funções de pertinência sobre o universo de discurso das variáveis de entrada. Como foi visto na seção 4.3.2 esse algoritmo pode ser facilmente feito linear.



**normalização de universos de discurso:** a normalização de um universo de discurso requer uma discretização do mesmo em um número finito de segmentos, onde cada segmento é mapeado em um segmento adequado no universo normalizado. O mapeamento pode ser uniforme ou não uniforme.

**regras:** constitui o principal elemento a ser manipulado para a obtenção das não-linearidades adequadas a cada caso.

**lógica de tomada de decisões:** está relacionado à escolha de uma função de implicação, interpretação do conectivo e, interpretação do conectivo de sentenças também, definição de um operador composicional e de um mecanismo de inferência. Em geral, utiliza-se o algoritmo de inferência de Mandani ou de Larsen (seção 2.9). Alguns pesquisadores como [67,68,50,51] investigaram a utilização de outros algoritmos definidos através de diferentes operadores, entretanto os resultados obtidos ainda não permitem conclusões definitivas.

**algoritmo de transformação fuzzy-escalar:** a transformação fuzzy-escalar é um mapeamento das ações de controle fuzzy para o espaço escalar. Em geral, os algoritmos adotados são não lineares, entretanto, é possível como foi visto na seção 4.4 adotar um algoritmo linear.

### 5.3 Estudo de Caso: Sistema de Primeira-Ordem com Tempo Morto

Um sistema de primeira-ordem com tempo morto pode ser representado pela seguinte função de transferência,

$$\frac{Y(s)}{U(s)} = C \frac{e^{(-as)}}{s + b}, \quad (5.1)$$

Nesse caso, adota-se um controlador fuzzy do tipo Proporcional-Integral, ou seja, as regras assumem a seguinte forma:

$$\text{se } e = A_i \text{ e } ce = B_i \text{ então } cu = C_i,$$

onde  $e$  representa o erro,  $ce$  representa a variação do erro e  $cu$  representa a variação do esforço de controle.

A figura 5.1 apresenta a estrutura de controle adotada.

A tabela 5.1 apresenta a normalização adotada e as definições quantitativas das funções de pertinência para as variáveis  $e$ ,  $ce$  e  $cu$ . A figura 5.2 ilustra os números fuzzy adotados (*Negative-Large*, *Negative-Medium*, *Negative-Small*, *Zero*, *Positive-Small*, *Positive-Medium*, *Positive-Large*).

A estratégia de controle adotada foi desenvolvida por Yamazaki e Sugeno [115] para plantas de primeira-ordem com tempo morto e também adotada por Mizumoto [68]. A tabela 5.2 apresenta as regras de controle desenvolvidas por Yamazaki e Sugeno.

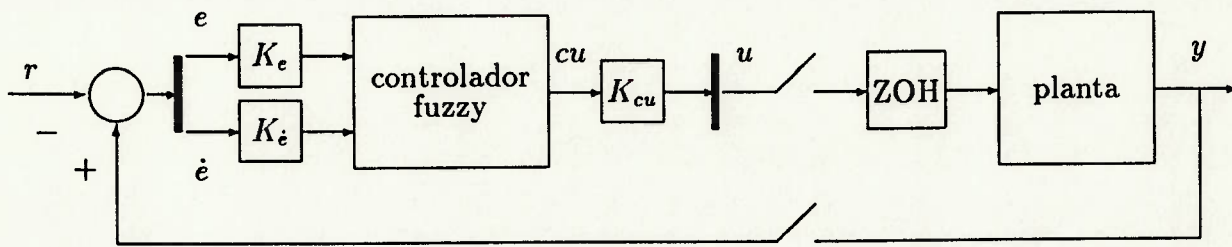


Figura 5.1: estrutura de controle para sistema de primeira ordem com atraso.

intervalos	segmentos normalizados	$\alpha_p$	$\alpha_b$	números fuzzy
[-2.0, +2.0]	[-1.0, +1.0]	-1.0	0.666	NL
		-0.667	0.666	NM
		-0.334	0.666	NS
		0.0	0.666	ZE
		+0.334	0.666	PS
		+0.667	0.666	PM
		+1.0	0.666	PL

Tabela 5.1: normalização e funções de pertinência para as variáveis  $e$ ,  $ce$  e  $cu$ . Veja nota de rodapé na página 59.

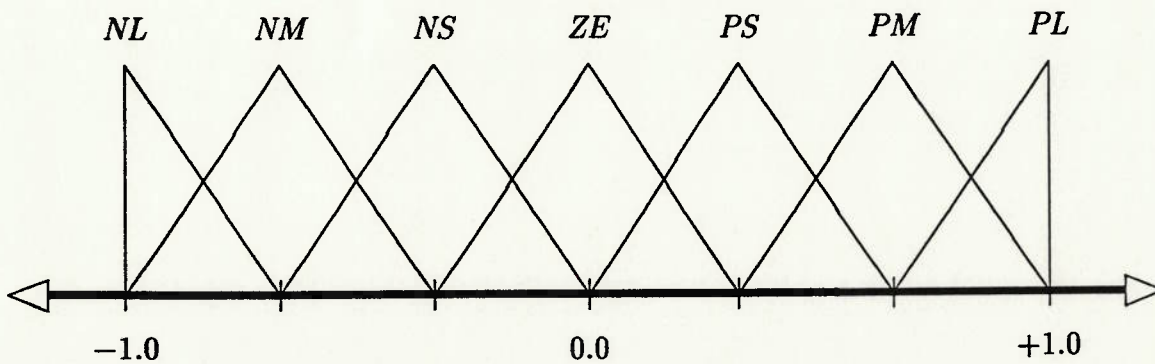


Figura 5.2: números fuzzy para as variáveis  $e$ ,  $ce$  e  $cu$ .

$e \setminus ce$	NL	NM	NS	ZE	PS	PM	PL
NL	ZE	ZE	ZE	PL	ZE	ZE	ZE
NM	ZE	ZE	ZE	PM	ZE	ZE	ZE
NS	ZE	ZE	ZE	PS	ZE	ZE	ZE
ZE	PB	PM	PS	ZE	NS	NM	NL
PS	ZE	ZE	ZE	NS	ZE	ZE	ZE
PM	ZE	ZE	ZE	NM	ZE	ZE	ZE
PL	ZE	ZE	ZE	NL	ZE	ZE	ZE

Tabela 5.2: tabela de decisões ( $e, ce \rightarrow cu$ ).

Após definidas as regras de controle e as funções de pertinência deve-se ajustar as constantes  $K_e$ ,  $K_{ce}$  e  $K_{cu}$  de modo a obter um desempenho satisfatório. As tabelas 5.3–5.6 apresentam resultados de simulações com diversas combinações para as constantes ( $K_e, K_{ce}, K_{cu}$ ) em termos do *tempo de subida*, *sobresinal* e *tempo de acomodação*. Os resultados foram obtidos através de simulações utilizando-se o algoritmo de raciocínio de Mandani (seção 2.9), a discretização do espaço normalizado é realizado com  $n_p = 31$  pontos (veja Apêndice A), tempo de amostragem  $h = 0.05s$ . A planta é simulada através de equações de diferenças.

O papel de  $K_{cu}$  é modificar o valor inferido pela tabela de decisões enquanto que  $K_e$  e  $K_{ce}$  deslocam a posição de entrada na tabela tornando o controlador mais sensível a  $e$  ou a  $ce$  conforme o caso. Observando a tabela de decisões 5.2 podemos verificar que esse controlador fuzzy produz saídas diferentes de zero quando uma das variáveis de entrada ou  $e$  ou  $ce$  se aproxima de zero exclusivamente. Isso indica que no início o controlador é mais sensível ao erro ( $e$ ) e à medida que o erro se aproxima de zero o controlador fica mais sensível à variação do erro  $ce$ .

A tabela 5.3 mostra o desempenho dinâmico com a variação de  $K_{cu}$  mantendo  $K_e$  e  $K_{ce}$  fixos. Para a faixa em que ocorre sobresinal o aumento de  $K_{cu}$  provoca a diminuição do tempo de subida, um aumento do sobresinal e uma diminuição do tempo de acomodação.

A tabela 5.4 mostra o desempenho dinâmico com a variação de  $K_e$  mantendo  $K_{ce}$  e  $K_{cu}$  fixos. Para o aumento de  $K_e$  o controlador se mostra bastante sensível, ocorrendo a diminuição do tempo de subida, aumento do sobresinal e aumento do tempo de acomodação.

As tabelas 5.5 e 5.6 mostram o desempenho dinâmico com a variação de  $K_{ce}$  mantendo  $K_e$  e  $K_{cu}$  fixos. Para pequenas variações de  $K_{ce}$  (tabela 5.5) não ocorrem variações significativas no desempenho dinâmico. Para variações maiores (tabela 5.6) a diminuição de  $K_{ce}$  dentro da faixa em que ocorre sobresinal, provoca uma diminuição do tempo de subida, um aumento do sobresinal e um aumento do tempo de acomodação.

Após diversas simulações foram escolhidos os seguintes valores para as constantes:  $K_e = 0.05$ ,  $K_{ce} = 0.7$  e  $K_{cu} = 1.5$ . A resposta dinâmica para degraus de 30%, 60% e 100% considerando tal escolha pode ser observado na figura 5.3.

Para verificar o efeito da utilização de outros algoritmos de raciocínio no desempenho dinâmico foram realizadas outras simulações considerando os algoritmos mostrados na



	$K_e$	$K_{ce}$	$K_{cu}$	tempo de subida (s) (0-100%)	sobresinal	tempo de acomodação em 5% (s)
1	0.1	0.5	0.5	2.1	0.217	4.75
2	0.1	0.5	0.4	2.5	0.167	5.3
3	0.1	0.5	0.3	3.15	0.107	6.1
4	0.1	0.5	0.2	4.70	0.045	4.15
5	0.1	0.5	0.1	19.6	0.0	9.05

Tabela 5.3: desempenho dinâmico com a variação de  $K_{cu}$ .

	$K_e$	$K_{ce}$	$K_{cu}$	tempo de subida (s) (0-100%)	sobresinal	tempo de acomodação em 5% (s)
1	0.1	0.5	0.3	3.15	0.107	6.1
2	0.2	0.5	0.3	2.0	0.318	6.75
3	0.3	0.5	0.3	1.65	0.460	9.65
4	0.4	0.5	0.3	1.45	0.574	14.1

Tabela 5.4: desempenho dinâmico com a variação de  $K_e$ .

	$K_e$	$K_{ce}$	$K_{cu}$	tempo de subida (s) (0-100%)	sobresinal	tempo de acomodação em 5% (s)
1	0.1	0.1	0.3	3.0	0.176	6.3
2	0.1	0.3	0.3	3.05	0.141	6.2
3	0.1	0.5	0.3	3.15	0.107	6.1

Tabela 5.5: desempenho dinâmico com a variação de  $K_{ce}$ .

	$K_e$	$K_{ce}$	$K_{cu}$	tempo de subida (s) (0-100%)	sobresinal	tempo de acomodação em 5% (s)
1	0.1	0.5	0.3	3.15	0.107	6.1
2	0.1	1.0	0.3	3.60	0.04	3.15
3	0.1	1.5	0.3	5.0	0.0	3.75
4	0.1	2.0	0.3	17.9	0.0	4.6

Tabela 5.6: desempenho dinâmico com a variação de  $K_{ce}$ .



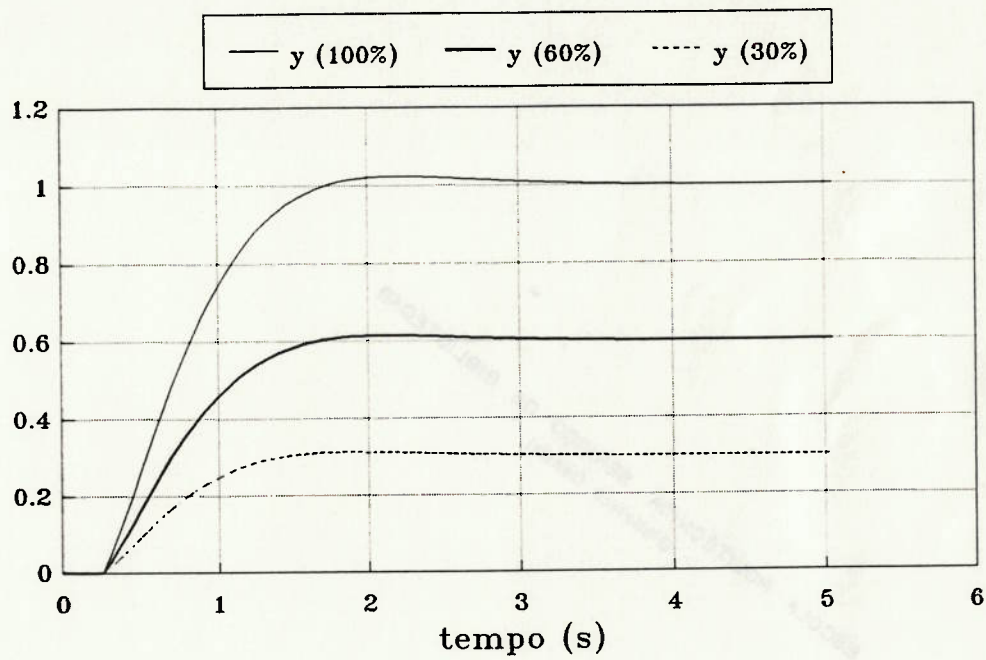


Figura 5.3: resposta dinâmica para degraus de 30%, 60% e 100%.

	função de implicação	e	também
$A_1$	$R_c$	intersecção	união
$A_2$	$R_a$	intersecção	produto limitado
$A_3$	$R_{bp}$	intersecção	união
$A_4$	$R_{dp}$	intersecção	união
$A_5$	$R_p$	intersecção	união
$A_6$	método simplificado		

Tabela 5.7: Algoritmos de Raciocínio Fuzzy.

tabela 5.7 (veja capítulo 2 para detalhes sobre algoritmos). A resposta a degrau  $y$  e o esforço de controle  $u$  para cada um dos algoritmos pode ser observado na figura 5.4. As superfícies de controle geradas para cada um dos algoritmos são apresentadas nas figuras 5.5–5.10. Os valores adotados para as constantes são:  $K_e = 0.05$ ,  $K_{ce} = 0.7$  e  $K_{cu} = 1.5$

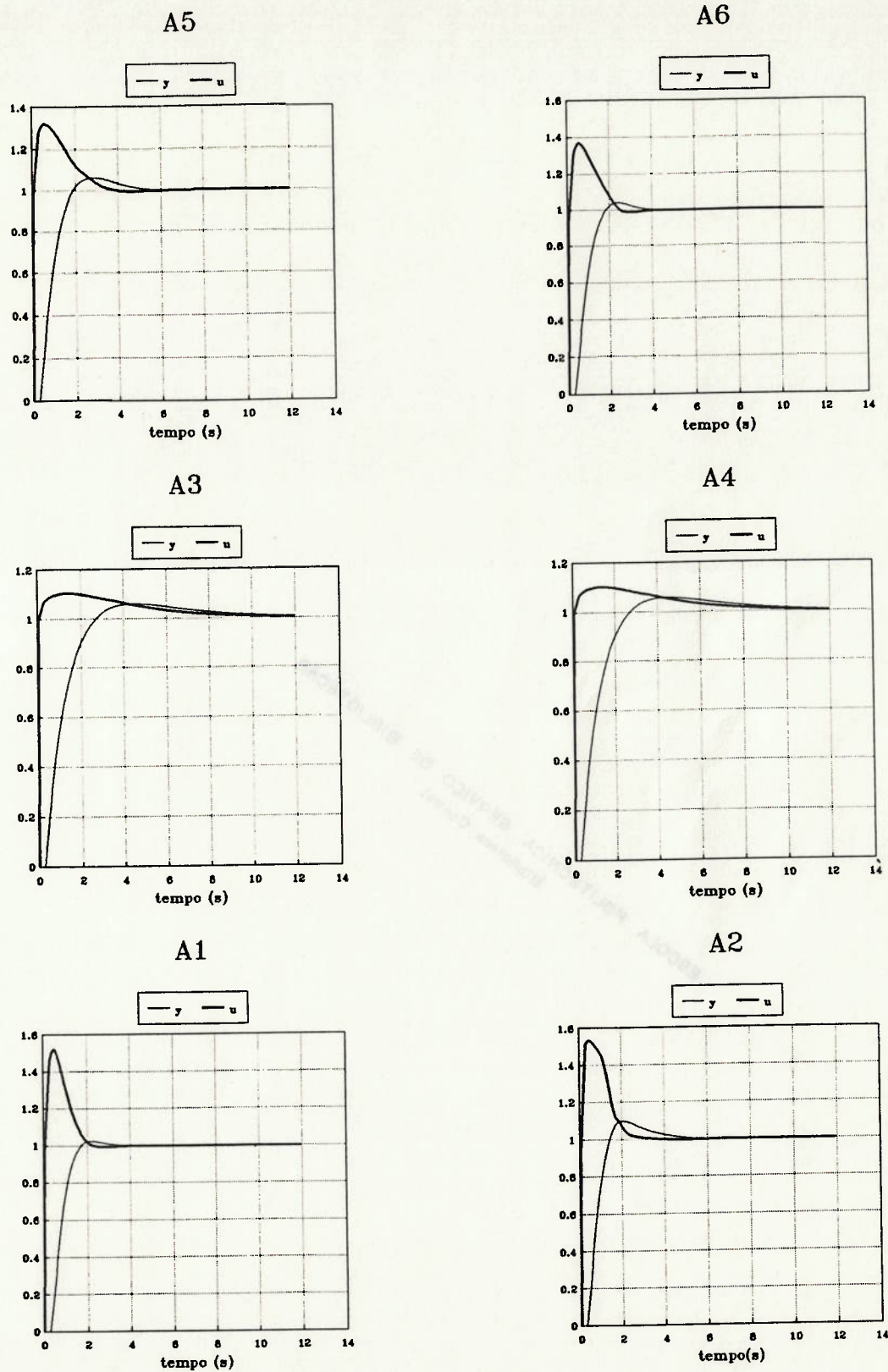


Figura 5.4: resposta a degrau  $y$  e esforço de controle  $u$ .

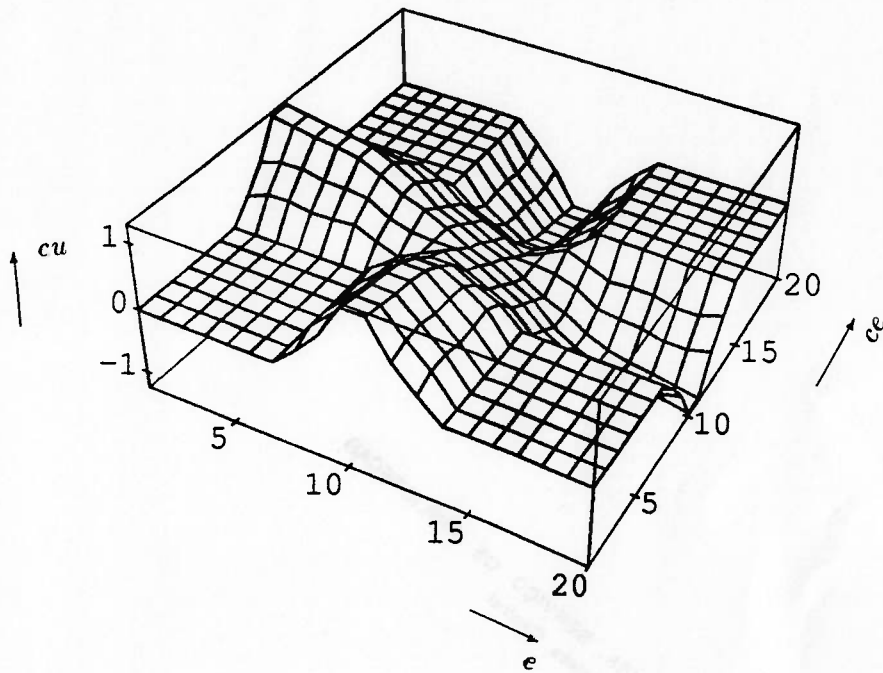


Figura 5.5: superfície de controle -  $A_1$ . O intervalo  $[0, 20]$  indicado nos eixos  $e$  e  $ce$  corresponde ao intervalo real  $[-2.0, +2.0]$ . A escala referente ao eixo  $cu$  corresponde aos valores reais.



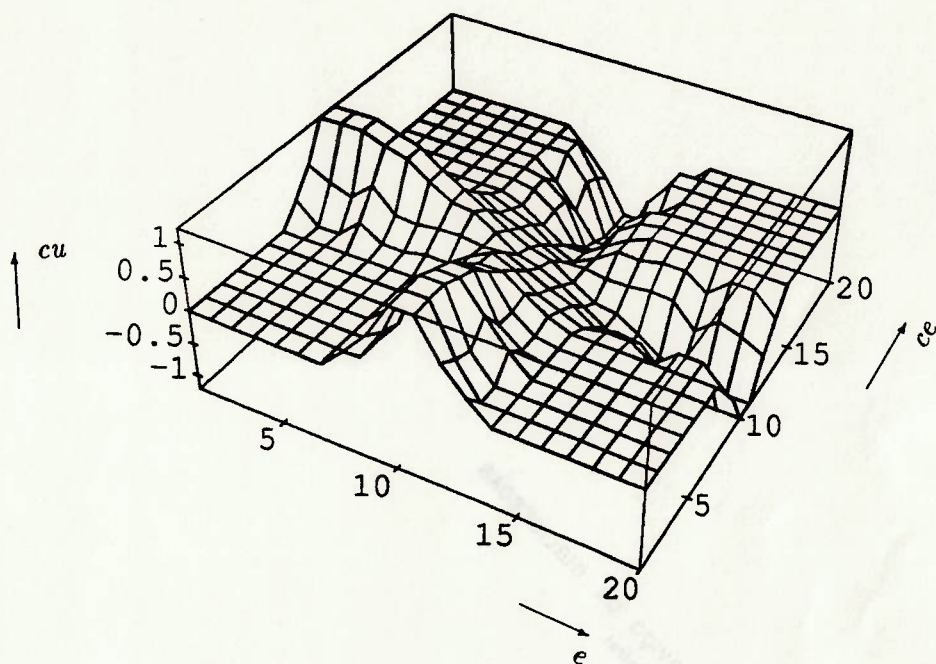


Figura 5.6: superfície de controle -  $A_2$ . O intervalo  $[0, 20]$  indicado nos eixos  $e$  e  $ce$  corresponde ao intervalo real  $[-2.0, +2.0]$ . A escala referente ao eixo  $cu$  corresponde aos valores reais.

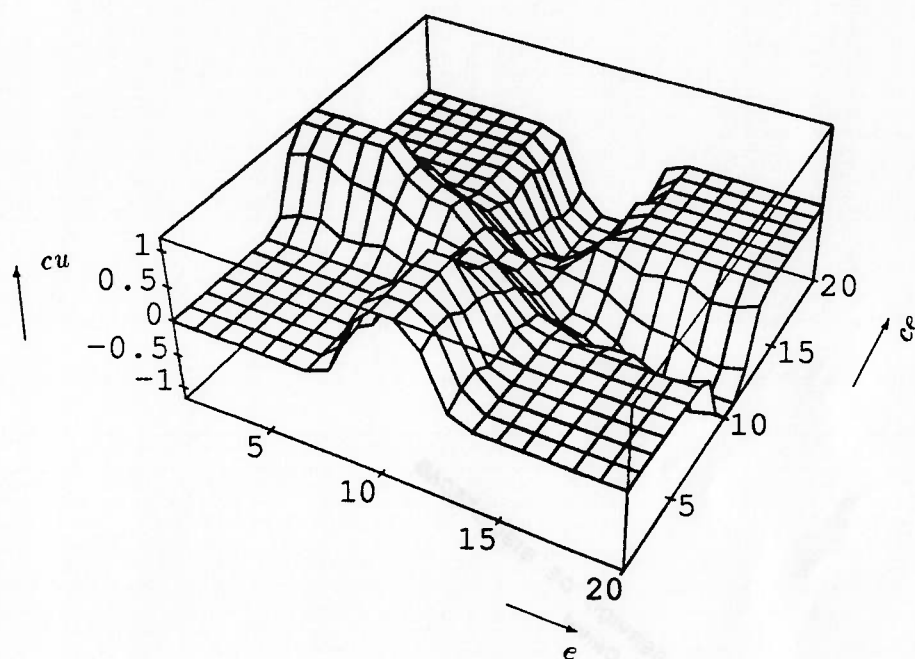


Figura 5.7: superfície de controle -  $A_3$ . O intervalo  $[0, 20]$  indicado nos eixos  $e$  e  $ce$  corresponde ao intervalo real  $[-2.0, +2.0]$ . A escala referente ao eixo  $cu$  corresponde aos valores reais.

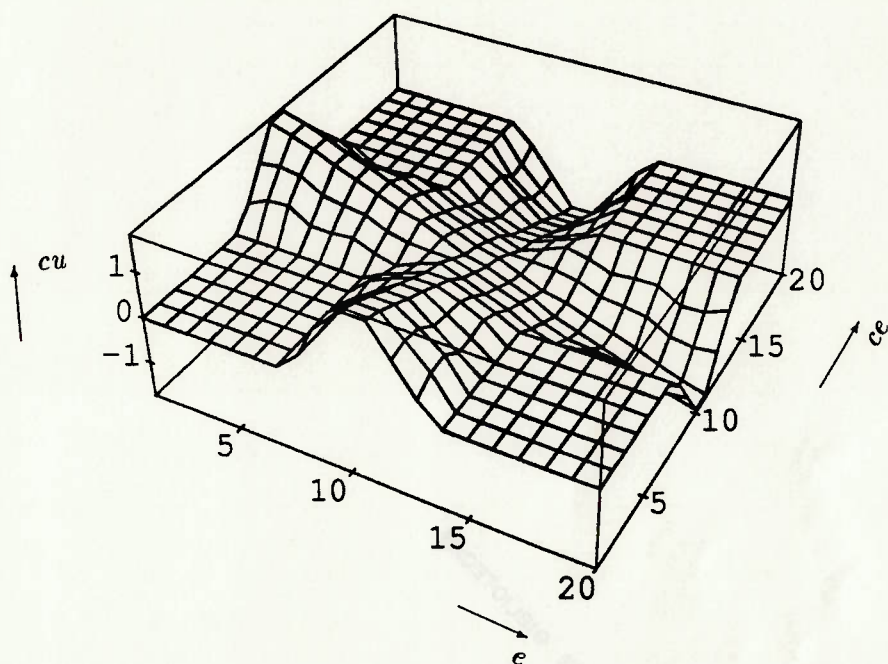


Figura 5.8: superfície de controle -  $A_4$ . O intervalo  $[0, 20]$  indicado nos eixos  $e$  e  $ce$  corresponde ao intervalo real  $[-2.0, +2.0]$ . A escala referente ao eixo  $cu$  corresponde aos valores reais.

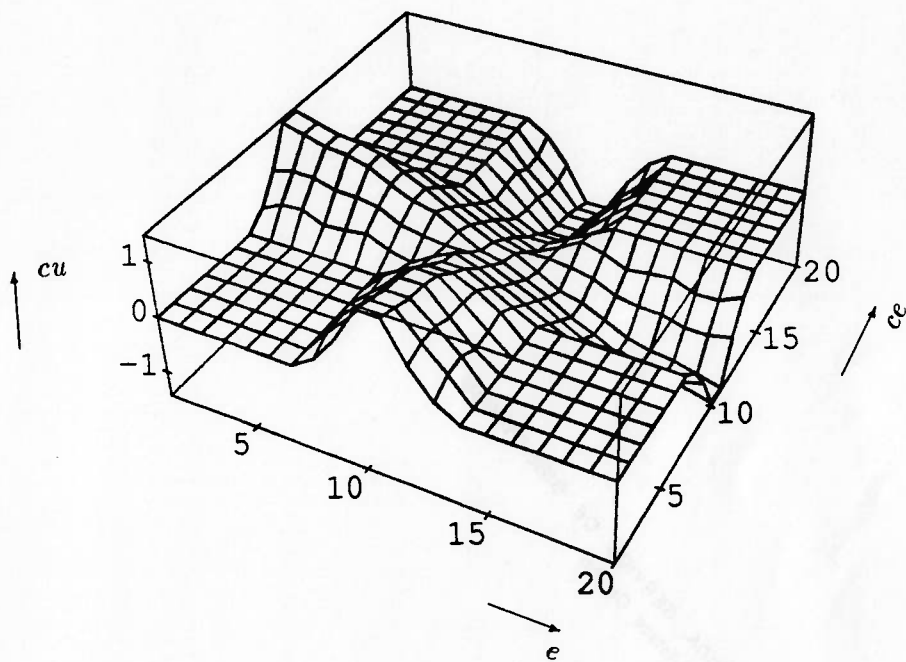


Figura 5.9: superfície de controle -  $A_5$ . O intervalo  $[0,20]$  indicado nos eixos  $e$  e  $ce$  corresponde ao intervalo real  $[-2.0, +2.0]$ . A escala referente ao eixo  $cu$  corresponde aos valores reais.



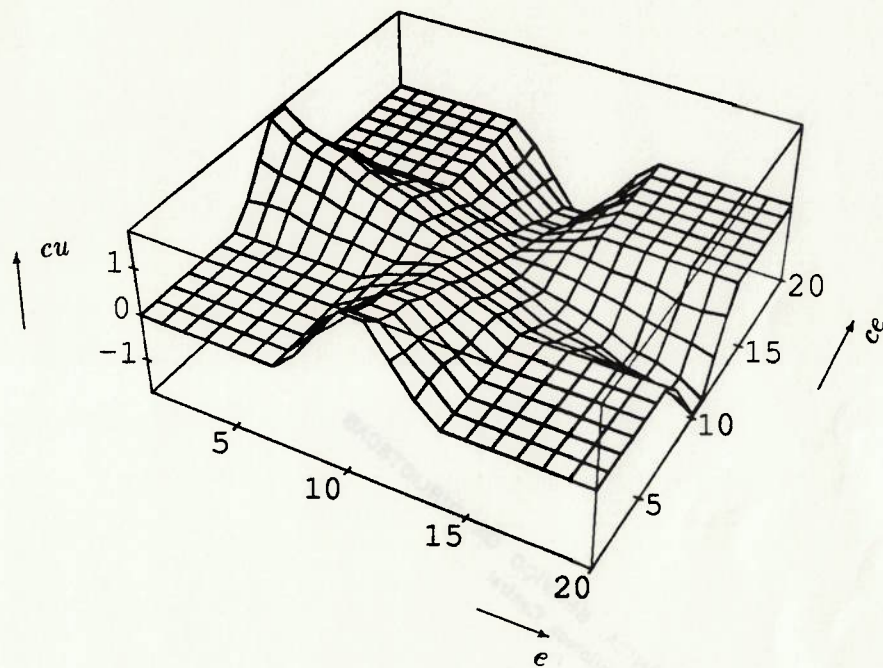


Figura 5.10: superfície de controle -  $A_6$ . O intervalo  $[0, 20]$  indicado nos eixos  $e$  e  $ce$  corresponde ao intervalo real  $[-2.0, +2.0]$ . A escala referente ao eixo  $cu$  corresponde aos valores reais.

## 5.4 Estudo de Caso: Pêndulo Invertido

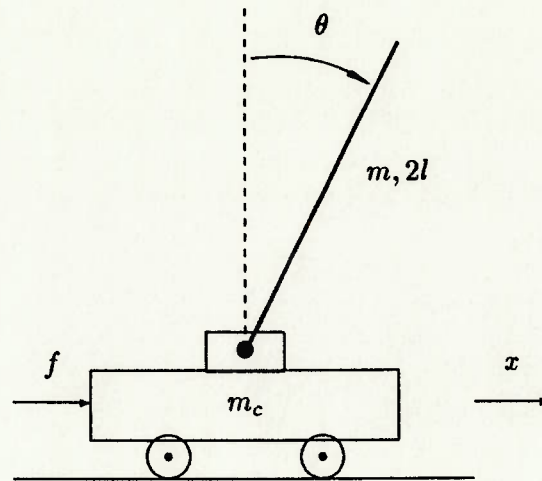


Figura 5.11: pêndulo invertido.

Um pêndulo invertido (veja figura 5.11) consiste de um carro acionado por um motor elétrico sobre o qual uma haste é montada através de um eixo com mancais de rolamento. A haste pode cair livremente.  $\theta$  é o ângulo entre o eixo e a linha vertical;  $m$  é a massa da haste;  $2l$  é o comprimento da haste;  $m_c$  é a massa do carro;  $f$  é a força aplicada no carro e  $x$  é a posição do carro. É suposto que a posição angular  $\theta$  e a velocidade angular  $\dot{\theta}$  podem ser medidos através de sensores apropriados.

O controle do pêndulo invertido consiste em manter a haste equilibrada ( $\theta \approx 0$ ) e o carro em sua posição central ( $x \approx 0$ ). Esse experimento já foi realizado entre outros por Barto [12] utilizando redes neurais, por Lee [55] utilizando redes neurais em associação com lógica fuzzy, e por Yamakawa [112] utilizando lógica fuzzy.

O experimento aqui realizado segue a metodologia desenvolvida por Chen em [22,23]. Inicialmente desenvolve-se o modelo matemático do pêndulo invertido e posteriormente apresenta-se o projeto do controlador fuzzy.

Assumindo o caso ideal, isto é, inexistência de atrito nas juntas, inexistência de escorregamento das rodas, etc., as equações dinâmicas para a haste podem ser deduzidas como,

$$m_c \ddot{x} + m \ddot{x} + ml \ddot{\theta} \cos \theta - ml \dot{\theta}^2 \sin \theta = f, \quad (5.2)$$

$$\frac{1}{3} ml^2 \ddot{\theta} + ml(\ddot{x} \cos \theta + l \ddot{\theta}) - mgl \sin \theta = 0. \quad (5.3)$$

Eliminando  $x$  de 5.2 e 5.3, temos

$$\frac{4}{3}l\ddot{\theta} + \frac{1}{m_c + m}(-ml\ddot{\theta}\cos\theta + ml\dot{\theta}^2\sin\theta + f)\cos\theta - g\sin\theta = 0, \quad (5.4)$$

que pode ser reescrita como:

$$\left(\frac{4l}{3} - m\frac{l}{m_c + m}\cos^2\theta\right)\ddot{\theta} + m\frac{l}{m_c + m}\sin\theta\cos\theta\dot{\theta}^2 - g\sin\theta + \frac{1}{m_c + m}f\cos\theta = 0. \quad (5.5)$$

Escolhendo  $x_1 = \theta$  e  $x_2 = \dot{\theta}$  como variáveis de estado e impondo a limitação  $-\frac{\pi}{2} < x_1 < \frac{\pi}{2}$  as equações de estado do sistema são:

$$\dot{x}_1 = x_2, \quad (5.6)$$

$$\dot{x}_2 = \frac{g\sin x_1 - \frac{ml}{m_c + m}\sin x_1\cos x_1x_2^2 - \frac{1}{m_c + m}f\cos x_1}{\frac{4l}{3} - \frac{ml}{m_c + m}\cos^2 x_1}. \quad (5.7)$$

O projeto do controlador fuzzy nesse caso será realizado com o auxílio de uma função de Lyapunov. Este método será detalhado na seção 6.4.

Seja o pêndulo invertido e o controlador fuzzy representados pelas seguintes equações, respectivamente,

$$\dot{x} = f(x, u), \quad (5.8)$$

$$u = g(x), \quad (5.9)$$

onde  $x$  representa a variável de estado,  $x \in \mathcal{R}^n$ ,  $u$  é a entrada imposta pelo controlador,  $u \in \mathcal{R}^q$ .

Para que o sistema seja estável (veja teoremas 6.2 e 6.3) é necessário que a superfície  $U_f$  que representa a função  $u = g(x)$  esteja totalmente contida dentro da região definida por,

$$U_c = \left\{ u \mid \forall x, \dot{V}(x) = \frac{dV}{dx}f(x, u) \leq 0 \right\}, \quad (5.10)$$

onde  $V(x)$  é uma função localmente definida positiva denominada função de Lyapunov.

Em geral, o método de Lyapunov é utilizado para se analisar a estabilidade de plantas associadas a controladores previamente determinados. Aqui, como proposto por Chen em [22,23], uma função de Lyapunov é escolhida inicialmente e o controlador fuzzy é então projetado de forma a se adequar a essa função, assegurando assim a estabilidade do sistema.

Uma escolha adequada para a função de Lyapunov depende de conhecimento prévio, em geral, funções quadráticas nas variáveis de estado se mostram boas opções.

Para esse caso em especial, vamos escolher a seguinte função de Lyapunov,

$$V = \frac{1}{2}(x_1^2 + x_2^2), \quad (5.11)$$



cuja derivada em função do tempo é,

$$\dot{V} = x_1 \dot{x}_1 + x_2 \dot{x}_2. \quad (5.12)$$

Substituindo 5.6 e 5.7 vem,

$$\dot{V} = x_1 x_2 + x_2 \frac{g \sin x_1 - \frac{ml}{m_c+m} \sin x_1 \cos x_1 x_2^2 - \frac{1}{m_c+m} f \cos x_1}{\frac{4}{3}l - \frac{ml}{m_c+m} \cos^2 x_1}, \quad (5.13)$$

$$= x_2 \left( x_1 + \frac{g \sin x_1 - \frac{ml}{m_c+m} \sin x_1 \cos x_1 x_2^2 - \frac{1}{m_c+m} f \cos x_1}{\frac{4}{3}l - \frac{ml}{m_c+m} \cos^2 x_1} \right). \quad (5.14)$$

Através do teorema 6.2, o sistema pode ser estabilizado se,

$$\dot{V} \leq 0. \quad (5.15)$$

Isso implica que para  $x_2 > 0$  é necessário que,

$$f > \frac{m_c + m}{\cos x_1} \left( g \sin x_1 - \frac{ml}{m_c + m} \sin x_1 \cos x_1 x_2^2 \right), \quad (5.16)$$

e para  $x_2 < 0$ ,

$$f < \frac{m_c + m}{\cos x_1} \left( g \sin x_1 - \frac{ml}{m_c + m} \sin x_1 \cos x_1 x_2^2 \right). \quad (5.17)$$

A superfície associada as inequações 5.16 e 5.17 é mostrada na figura 5.12.

Baseado em  $U_c$  (equação 5.10) pode-se agora projetar o controlador fuzzy. Deve-se escolher um conjunto de regras e dados de forma que a superfície de controle  $U_f$  esteja totalmente contida em  $U_c$ .

Inicialmente, vamos nos preocupar apenas em como equilibrar a haste. Nesse caso a política de controle pode ser representada pelas seguintes regras,

1. se a haste está caindo para a direita, então empurre o carro para a direita,
2. se a haste está caindo para a esquerda, então empurre o carro para a esquerda.

As variáveis escolhidas são: o deslocamento angular  $\theta$ , a velocidade angular  $\dot{\theta}$  e a força aplicada no carrinho  $f$ .

Definem-se três números fuzzy no universo de discurso de  $\theta$  e  $\dot{\theta}$ :  $PO(Positive)$ ,  $ZE(Zero)$ , e  $NE(Negative)$ . A figura 5.13 ilustra as funções de pertinência correspondentes e as tabelas 5.8 e 5.9 mostram as definições quantitativas das funções de pertinência. Nessas duas tabelas  $\alpha_p$  e  $\alpha_b$  são respectivamente o ponto de máximo e a largura da base dos triângulos isósceles e se referem ao universo de discurso normalizado.

Para a variável  $f$  são definidos sete números fuzzy:  $NL(Negative-Large)$ ,  $NM(Negative-Medium)$ ,  $NS(Negative-Small)$ ,  $ZE(Zero)$ ,  $PS(Positive-Small)$ ,  $PM(Positive-Medium)$  e  $PL(Positive-Large)$ . A figura 5.14 ilustra os números fuzzy referentes a  $f$  e a tabela 5.10 mostra as definições quantitativas correspondentes.

As regras de controle para o equilíbrio da haste são as seguintes:



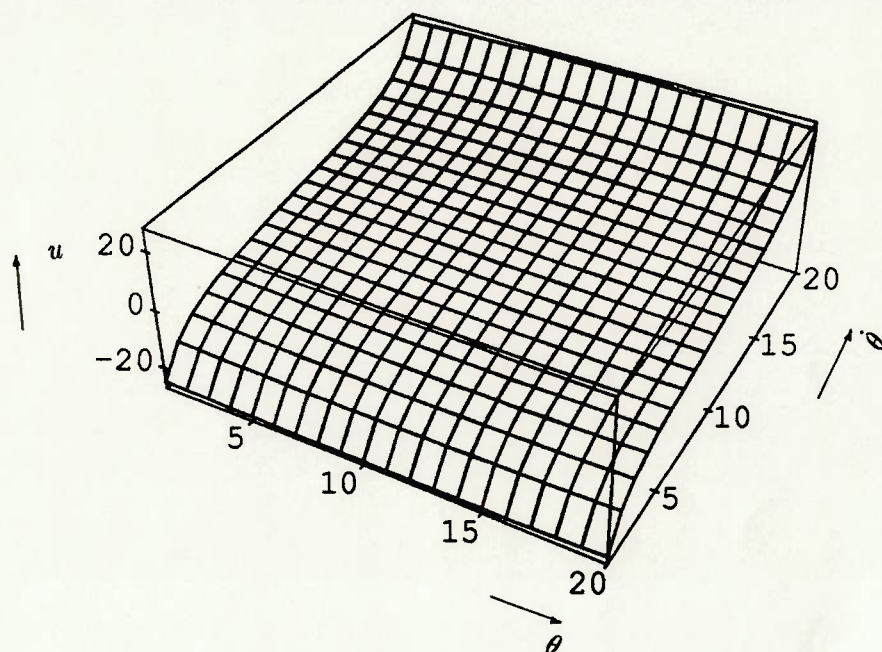


Figura 5.12: Superfície de restrição associada a  $U_c$ . O intervalo  $[0, 20]$  indicado no eixo  $\theta$  corresponde ao intervalo real  $[-2.0, +2.0]$  e no eixo  $\dot{\theta}$  corresponde ao intervalo  $[-15.0, +15.0]$ . A escala referente ao eixo  $u$  corresponde a valores reais.

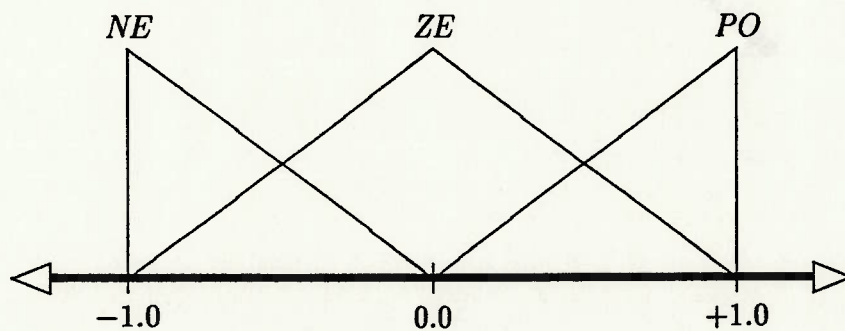


Figura 5.13: números fuzzy para as variáveis de estado.

intervalos	segmentos		$\alpha_p$	$\alpha_b$	número fuzzy
	normalizados				
[-2.0, +2.0]	[-1.0, +1.0]		-1.0	2.0	NE
			0.0	2.0	ZE
			+1.0	2.0	PO

Tabela 5.8: normalização e funções de pertinência para a variável  $\theta$ . Veja nota de rodapé na página 59.

intervalos	segmentos		$\alpha_p$	$\alpha_b$	número fuzzy
	normalizados				
[-15.0, +15.0]	[-1.0, +1.0]		-1.0	2.0	NE
			0.0	2.0	ZE
			+1.0	2.0	PO

Tabela 5.9: normalização e funções de pertinência para a variável  $\dot{\theta}$ . Veja nota de rodapé na página 59.

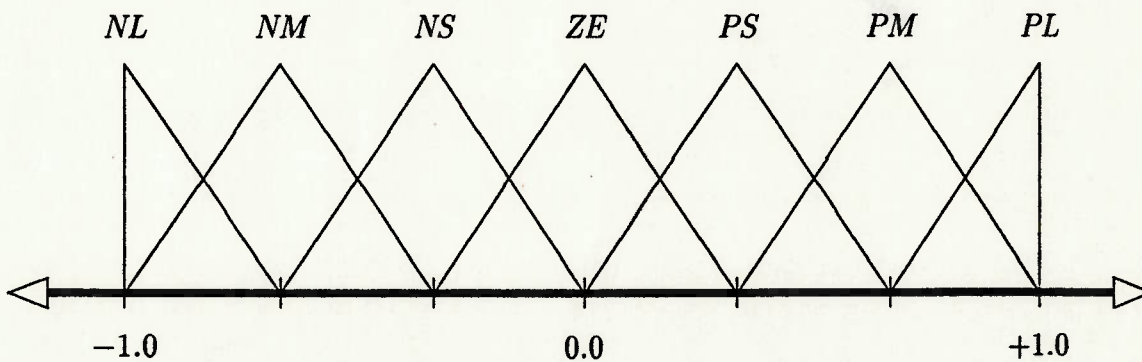


Figura 5.14: números fuzzy para  $f$ .

intervalos	segmentos normalizados	$\alpha_p$	$\alpha_b$	números fuzzy
		-1.0	0.666	NL
		-0.667	0.666	NM
		-0.334	0.666	NS
[-70.0, +70.0]	[-1.0, +1.0]	0.0	0.666	ZE
		+0.334	0.666	PL
		+0.667	0.666	PM
		+1.0	0.666	PL

Tabela 5.10: normalização e funções de pertinência para a variável  $f$ . Veja nota de rodapé na página 59.

Regra 1: se  $\theta = PO$  e  $\dot{\theta} = PO$ , então  $f = PL$ ,

Regra 2: se  $\theta = PO$  e  $\dot{\theta} = ZE$ , então  $f = PM$ ,

Regra 3: se  $\theta = PO$  e  $\dot{\theta} = NE$ , então  $f = ZE$ ,

Regra 4: se  $\theta = ZE$  e  $\dot{\theta} = PO$ , então  $f = PS$ ,

Regra 5: se  $\theta = ZE$  e  $\dot{\theta} = ZE$ , então  $f = ZE$ ,

Regra 6: se  $\theta = ZE$  e  $\dot{\theta} = NE$ , então  $f = NS$ ,

Regra 7: se  $\theta = NE$  e  $\dot{\theta} = PO$ , então  $f = ZE$ ,

Regra 8: se  $\theta = NE$  e  $\dot{\theta} = ZE$ , então  $f = NM$ ,

Regra 9: se  $\theta = NE$  e  $\dot{\theta} = NE$ , então  $f = NL$ .

A figura 5.15 ilustra a superfície  $U_f$  que caracteriza o controlador fuzzy.

A figura 5.16 ilustra a estrutura de controle para o pêndulo invertido. As experiências com o pêndulo invertido foram realizadas através de simulações considerando o algoritmo de inferência simplificado,  $K_\theta = 1.0$ ,  $K_{\dot{\theta}} = 1.0$ ,  $K_f = 1.0$ ,  $T = 0.05s$ . O pêndulo invertido foi simulado através do algoritmo de Runge-Kutta de quarta ordem com o passo de integração,  $h = T/10s$ . Os parâmetros utilizados para o pêndulo invertido foram:  $l = 0.286m$ ,  $m = 0.05kg$  e  $m_c = 0.67kg$ .

As figuras 5.17–5.19 mostram o comportamento das variáveis  $\theta$ ,  $\dot{\theta}$  e  $f$  para as condições iniciais:  $\theta_0 = 0.5rad$  e  $\dot{\theta}_0 = 1.0rad/s$ .

A derivação de regras para o controle da haste é relativamente fácil considerando uma função de Lyapunov para auxílio no projeto. Entretanto, o sistema se torna mais complexo de ser controlado se for considerado a posição do carro. Ou seja, nesse caso, além de equilibrar a haste, o sistema de controle deve fazer com que o carro permaneça em torno da origem, isto é,  $x \approx 0$ .

A política de controle nesse caso pode ser representada pelas seguintes regras:

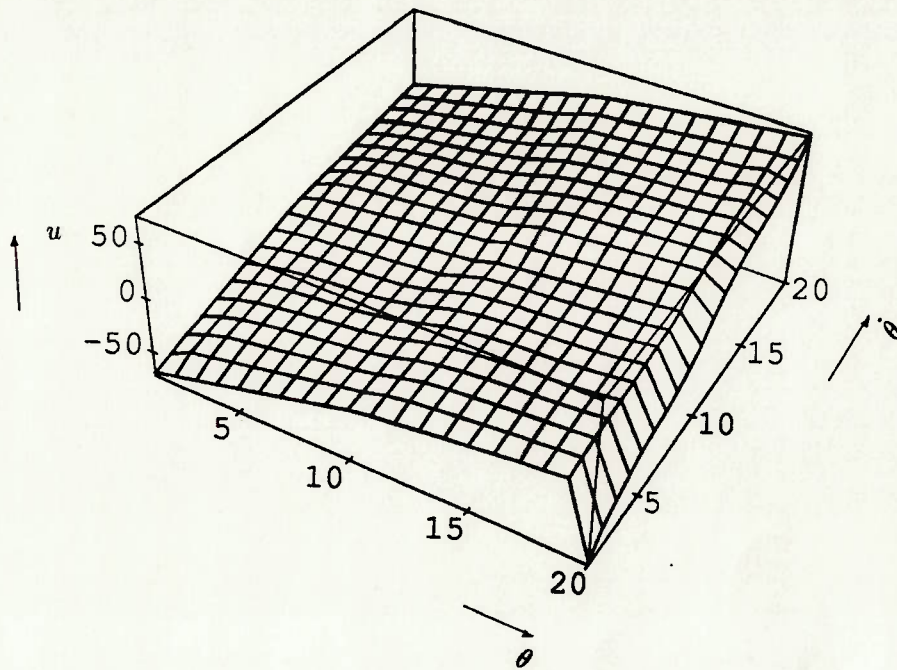


Figura 5.15: superfície de controle  $U_f$ . O intervalo  $[0, 20]$  indicado no eixo  $\theta$  corresponde ao intervalo real  $[-2.0, +2.0]$  e no eixo  $\dot{\theta}$  corresponde ao intervalo  $[-15.0, +15.0]$ . A escala referente ao eixo  $u$  corresponde a valores reais.

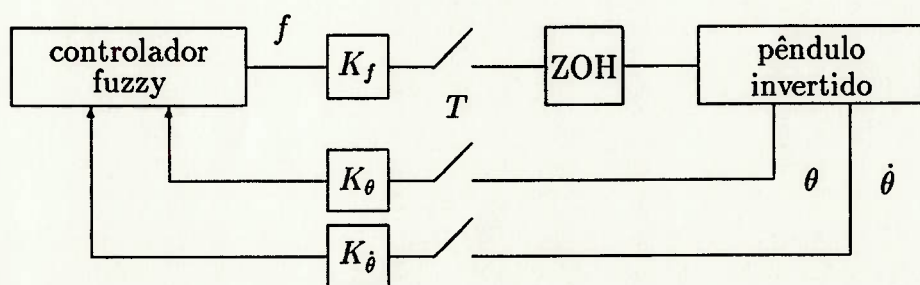


Figura 5.16: estrutura para o controle da haste.



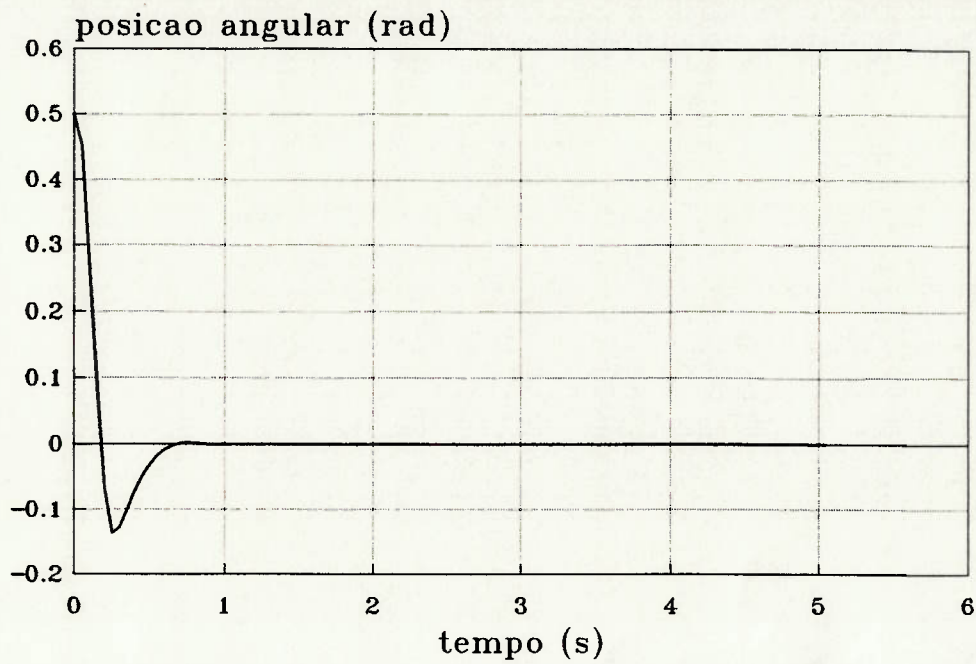


Figura 5.17: posição angular -  $\theta$ .

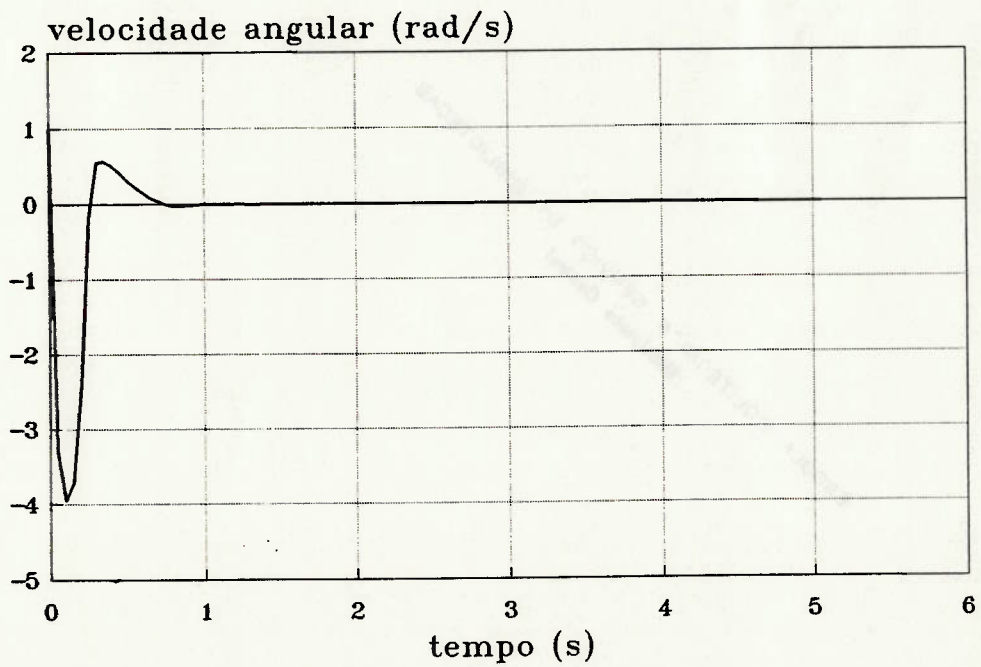
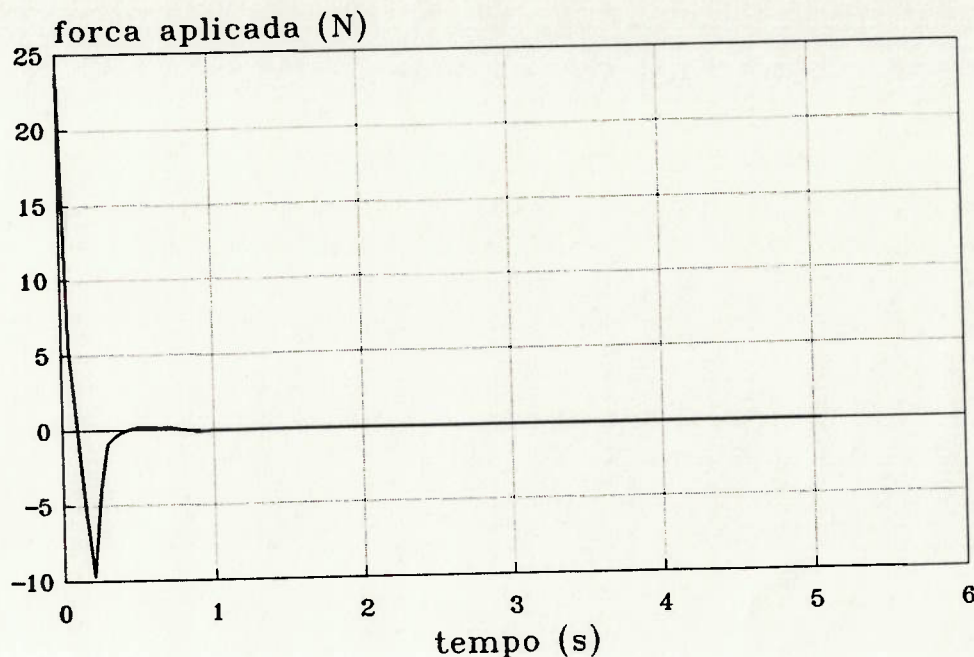


Figura 5.18: velocidade angular -  $\dot{\theta}$ .

Figura 5.19: força aplicada -  $f$ .

1. se a haste está quase equilibrada e o carro está a direita, então empurre o carro para a direita,
2. se a haste está quase equilibrada e o carro está a esquerda. então empurre o carro para a esquerda.

A idéia expressa através de tais regras é empurrar o carro para a mesma direção em que ele se encontra, com uma força suficientemente grande para que a haste comece a cair para a direção oposta. Então, a tentativa de equilibrar a haste conduz o carro para a origem.

As regras acima podem ser sintetizadas em termos das variáveis de estado e da variável de controle através das seguintes regras:

Regra 10: se  $\theta = VS$  e  $\dot{\theta} = VS$  e  $x = PO$   $\dot{x} = PO$  então  $f = PM$ ,

Regra 11: se  $\theta = VS$  e  $\dot{\theta} = VS$  e  $x = PO$   $\dot{x} = PVS$  então  $f = PS$ ,

Regra 12: se  $\theta = VS$  e  $\dot{\theta} = VS$  e  $x = NE$   $\dot{x} = PVS$  então  $f = NM$ ,

Regra 13: se  $\theta = VS$  e  $\dot{\theta} = VS$  e  $x = NE$   $\dot{x} = NVS$  então  $f = NS$ .

onde  $VS=Very-Small$ ,  $PVS= Positive-Very-Small$ , e  $NVS=Negative-Very-Small$ .

As figuras 5.20–5.23 ilustram qualitativamente e quantitativamente os números fuzzy envolvidos nas regras 10–13. Deve-se observar que os números fuzzy foram definidos em universos de discurso não-normalizados.

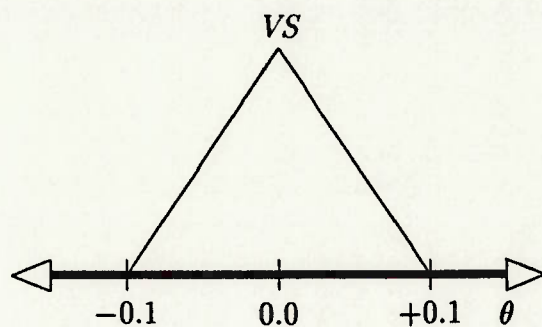


Figura 5.20: VS para a variável  $\theta$ .

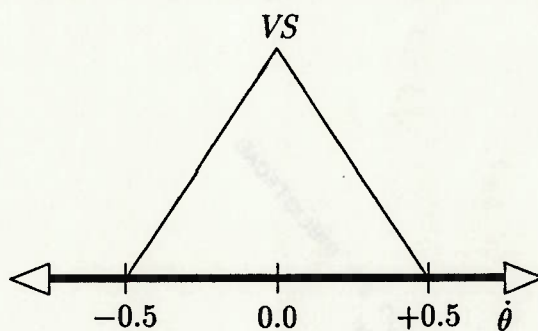


Figura 5.21: VS para a variável  $\dot{\theta}$ .

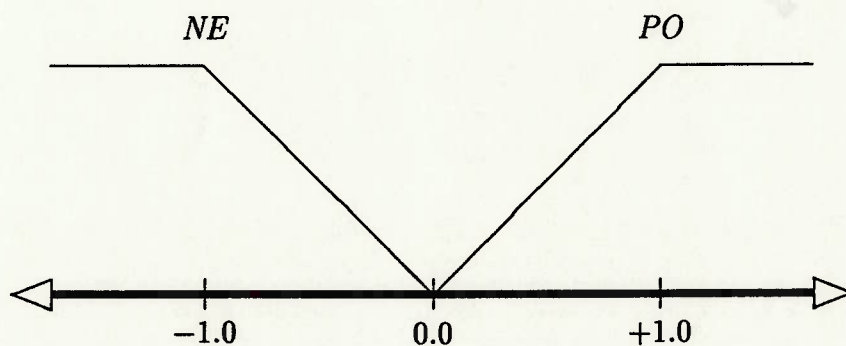


Figura 5.22: NE, PO para a variável  $x$ .

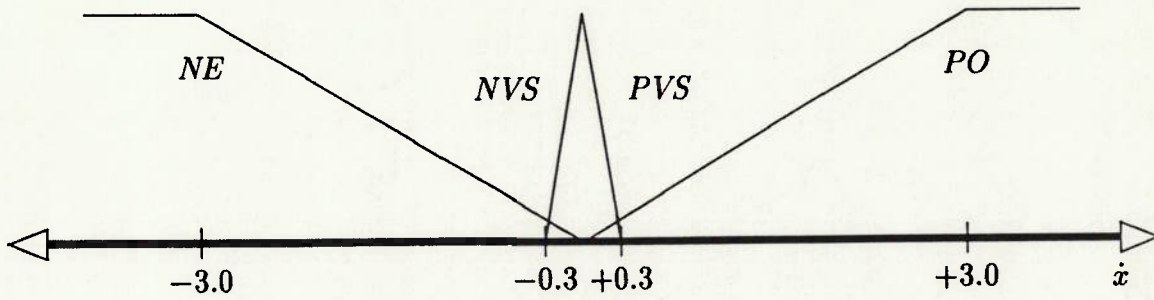


Figura 5.23: *NE, NVS, PVS, PO* para a variável  $\dot{x}$ .

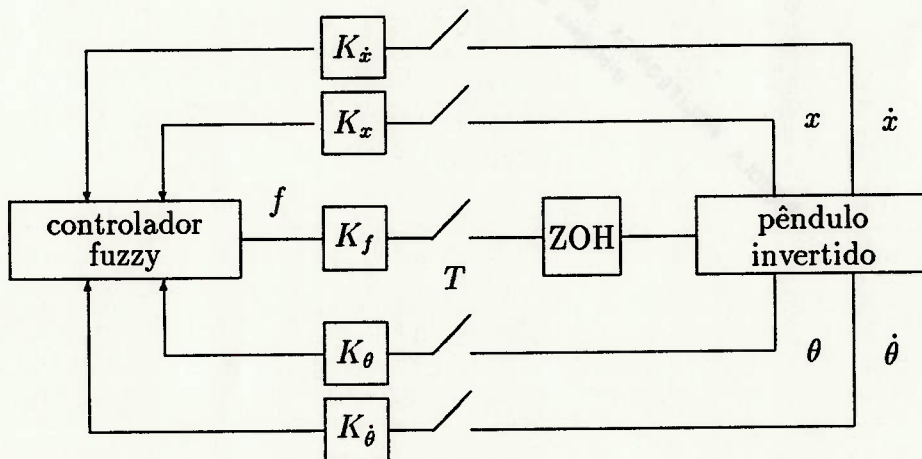
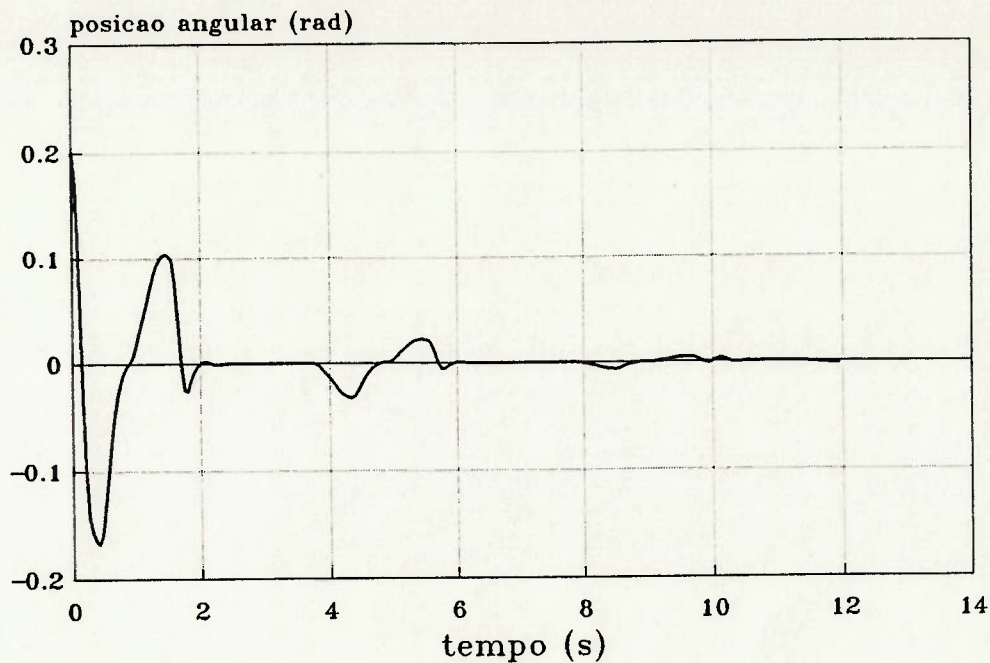


Figura 5.24: estrutura de controle para o pêndulo + carro.



Figura 5.25: posição angular -  $\theta$ .

A estrutura de controle pode agora ser representada pela figura 5.24.

Nessa parte, as simulações foram realizadas considerando o algoritmo de inferência simplificado,  $K_\theta = 1.0$ ,  $K_{\dot{\theta}} = 1.0$ ,  $K_f = 1.5$ ,  $K_x = 0.3$ ,  $K_{\dot{x}} = 0.3$ ,  $T = 0.05s$ . O pêndulo invertido foi simulado através do algoritmo de Runge-Kutta de quarta ordem com o passo de integração,  $h = T/10s$ .

As figuras 5.25-5.29 ilustram o comportamento das variáveis  $\theta$ ,  $\dot{\theta}$ ,  $x$ ,  $\dot{x}$  e  $f$  para as seguintes condições iniciais:  $\theta_0 = 0.2rad$ ,  $\dot{\theta}_0 = 0.3rad/s$ ,  $x_0 = 0m$ ,  $\dot{x}_0 = 0m/s$ .

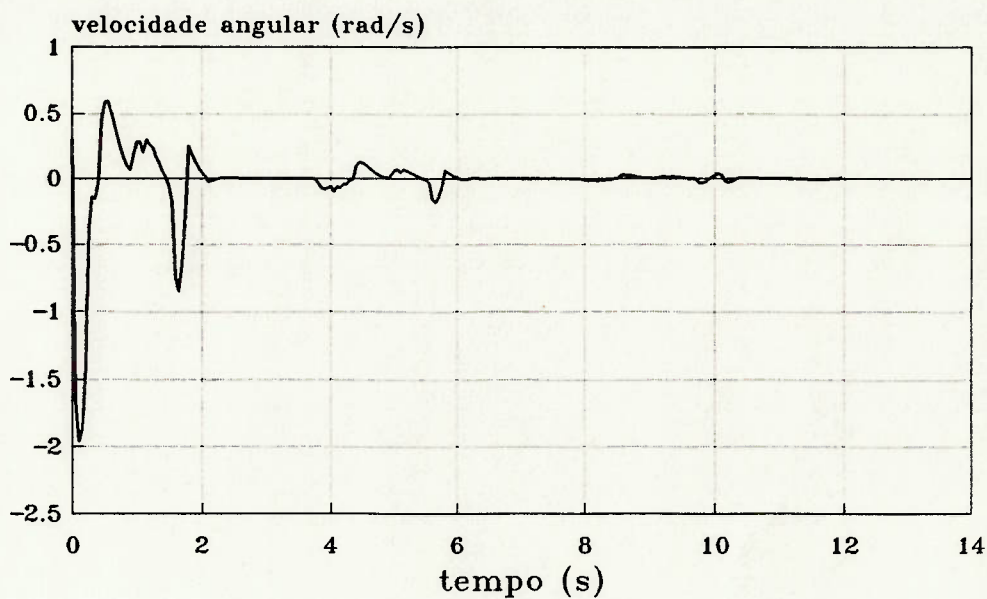


Figura 5.26: velocidade angular -  $\dot{\theta}$ .

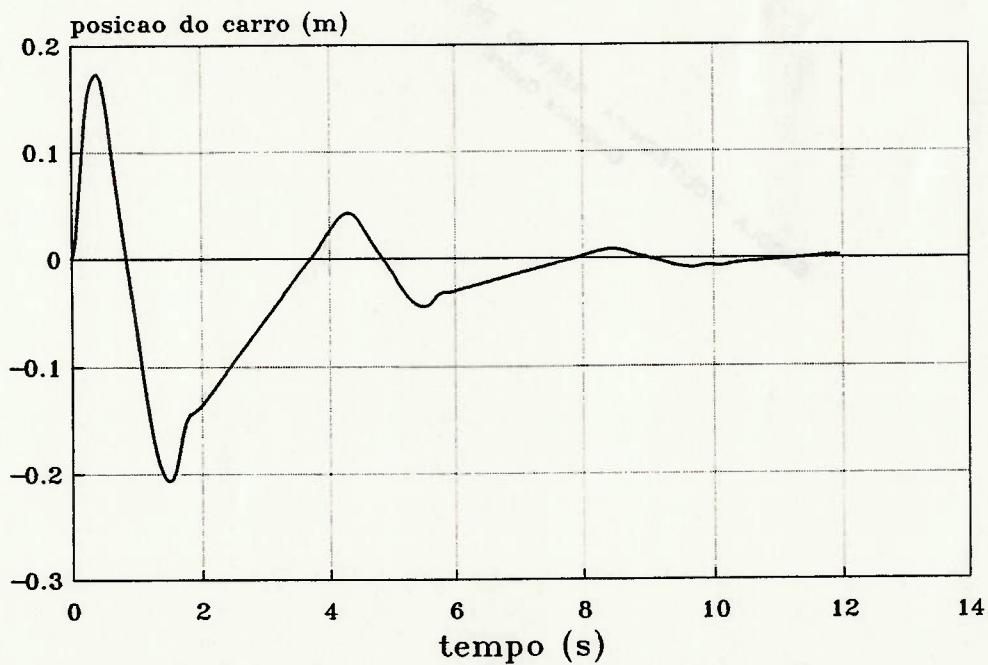


Figura 5.27: posição do carro -  $x$ .

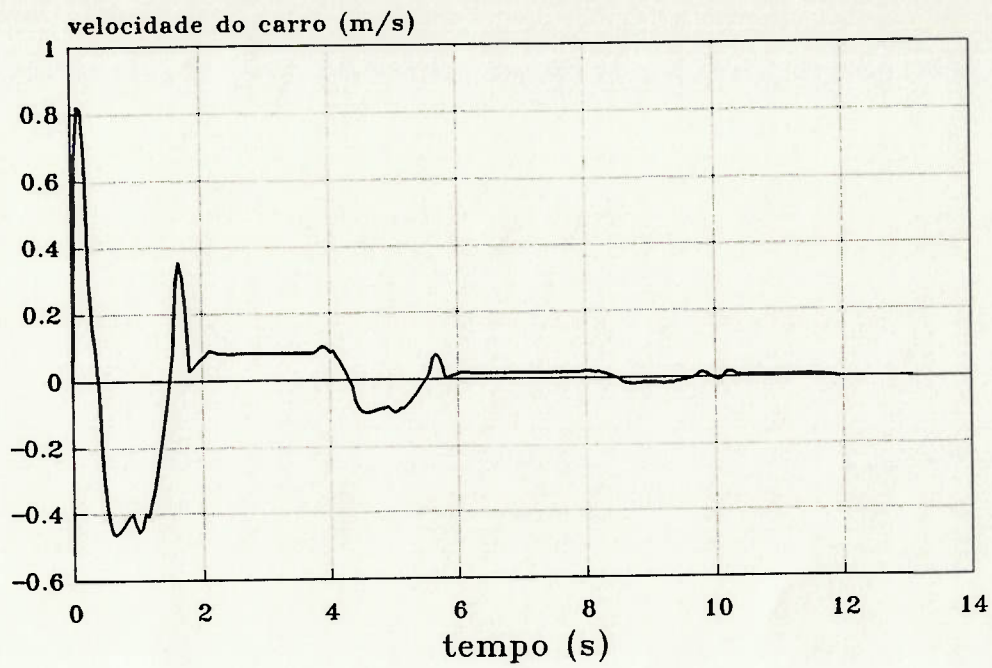


Figura 5.28: velocidade do carro -  $\dot{x}$ .

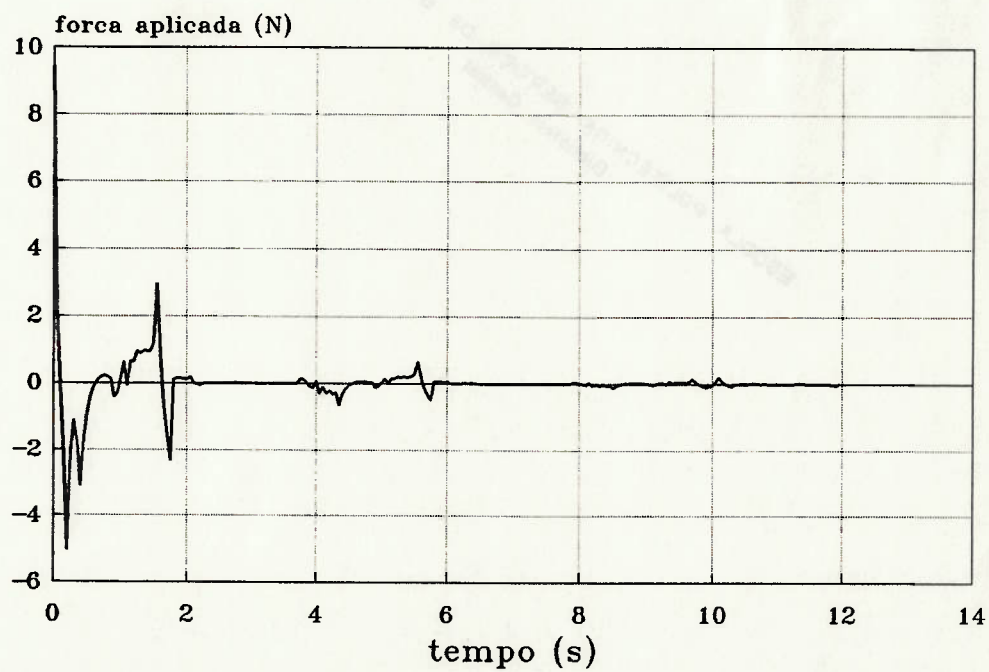


Figura 5.29: força aplicada -  $f$ .

## 5.5 Conclusão

Esse capítulo procurou demonstrar o desempenho de controladores fuzzy como elementos não lineares de controle.

As fontes de não linearidades podem ocorrer em vários pontos da estrutura de um controlador fuzzy: algoritmo de transformação escalar-fuzzy, normalização de universos de discurso, base de regras, lógica de tomada de decisões e algoritmo de transformação fuzzy-escalar. O perfeito domínio a respeito da influência sobre as não linearidades de cada uma das partes que compõem a estrutura do controlador ainda não foi perfeitamente compreendida. A experiência prévia é o aspecto dominante.

No controlador fuzzy projetado para o sistema de primeira-ordem com tempo morto foi utilizado uma base de regras desenvolvida por Yamazaki e Sugeno [115]. Para uma mesma base de conhecimento foram utilizados 6 algoritmos distintos envolvendo escolhas diferentes para o conectivo e conectivo também e a função de implicação fuzzy. Muito embora muitos pesquisadores tenham realizado diversas pesquisas a respeito de novos algoritmos, a quase totalidade dos controladores fuzzy utiliza o algoritmo de Mandani, ou o algoritmo de Larsen ou o algoritmo simplificado. Os três algoritmos citados parecem ser os melhores, principalmente levando-se em conta a simplicidade das operações envolvidas nos algoritmos.

No segundo exemplo desse capítulo, projetou-se um controlador fuzzy para um pêndulo invertido utilizando o auxílio de uma função de Lyapunov. Cabe lembrar que essa técnica é apenas uma condição suficiente, ou seja, mesmo que o controlador não obedeça a superfície de restrição dada pela função de Lyapunov, o sistema ainda pode ser estável.



# Capítulo 6

## Análise de Controladores Fuzzy

### 6.1 Introdução

Esse capítulo trata sobre as técnicas de análise de controladores fuzzy.

Muito embora a técnica de projeto de controladores fuzzy seja bastante simples de ser aplicada, a análise de tais controladores sob o ponto de vista de conceitos como estabilidade e robustez é bastante difícil, em grande parte devido a sua característica linguística e não formal. A impossibilidade de se analisar adequadamente o sistema (planta + controlador) dentro de uma mesma formulação matemática precisa, em geral, só permite que as conclusões sejam tiradas através de implementações práticas.

As técnicas propostas são em geral adaptações de ferramentas utilizadas em análise de sistemas não lineares ou se utilizam de equações relacionais fuzzy. No presente momento, não existe nenhuma técnica que possa ser considerada satisfatória para a análise de controladores fuzzy.

A técnica de *Funções Descritivas* (seção 6.2) e o *Critério de Estabilidade do Círculo* (seção 6.3) foram as primeiras técnicas utilizadas, entretanto, permitem somente a análise de uma classe limitada de controladores fuzzy que se comportam como elementos *Multilevel-Relay*. Na seção 6.4 é mostrado como utilizar funções de Lyapunov para auxílio no projeto de controladores fuzzy. As seções 6.5 e 6.6 tratam de técnicas relacionadas à *Equações Relacionais Fuzzy* que permitem tratar o sistema global dentro do domínio de conjuntos fuzzy. Finalmente a seção 6.7 apresenta o *Método de Mapeamento Célula-a-Célula*.

### 6.2 Funções Descritivas

A aplicação do método de análise de controladores fuzzy através de funções descritivas foi proposto por Kickert e Mandani em [48]. Tal análise só pode ser aplicada para o caso especial em que o controlador fuzzy se comporta como um controlador não-linear do tipo *Multilevel-Relay*. Dessa forma o controlador fuzzy pode ser analisado como um controlador não linear convencional.

Suponha que um controlador fuzzy possua em sua base de conhecimento, regras do seguinte tipo:

$$\text{se } e = A \text{ então } u = C,$$

onde  $e$  é a variável fuzzy correspondente ao erro do processo e  $u$  é o esforço de controle. Nesse caso, ao invés da utilização do algoritmo de Mandani como foi visto no capítulo 2,

$$R_C : \mu_{C'} = \bigcup_{i=1}^n \alpha_i \wedge \mu_{C_i}(u), \quad (6.1)$$

deve-se fazer:

$$R_C : \mu_{C'} = \max_u \max_i \{ \alpha_i \wedge \mu_{C_i}(u) \}, \quad (6.2)$$

o que implica em considerar apenas uma única regra a cada inferência.

O algoritmo de transformação fuzzy-escalar adotado é o método da média dos pontos de máximo,

$$u_0 = \sum_{j=1}^l \frac{u_j}{l}, \quad (6.3)$$

onde  $u_j$  é um ponto onde a função de pertinência assume um valor máximo e  $l$  é o número de pontos onde ocorrem tais máximos.

Como exemplo, suponha que as funções de pertinência correspondentes a  $e$  sejam as funções representadas na figura 6.1(a) e as funções correspondentes a  $u$  estejam representadas na figura 6.1(b). Nesse caso a característica entrada-saída do controlador fuzzy pode ser representado pela figura 6.1(c).

Em resumo, o algoritmo aqui apresentado divide o domínio da variável de entrada em regiões e transforma o domínio da variável de saída em um conjunto de pontos. Um algoritmo desse tipo portanto, transforma o controlador fuzzy num controlador do tipo *Multilevel-Relay*.

### 6.2.1 Funções Descritivas de Elementos Multilevel-Relay

Uma das ferramentas para análise de sistemas não lineares, especialmente para oscilações sustentadas são as funções descritivas. Assume-se aqui que o elemento *Multilevel-Relay* é simétrico em relação ao zero. Deve-se ter em mente que a análise através de funções descritivas é apenas aproximada. Por exemplo, existem situações em que apesar da previsão de oscilações através dessa análise, tais oscilações não ocorrem. Existem situações em que ocorre o inverso. Mais precisamente, podemos dizer que a análise através de funções descritivas prevê a probabilidade de ocorrência de ciclos limites.

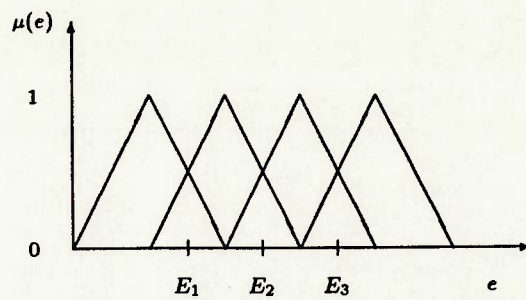
O método de funções descritivas consiste em substituir o elemento  $N$  não linear (veja figura 6.3) por um outro elemento linear calculado através de quasi-linearizações ótimas [103].

Nesse contexto, pode ser provado que para uma entrada senoidal  $x_0(t)$  com período  $2\pi/\omega$  e amplitude  $a$ ,

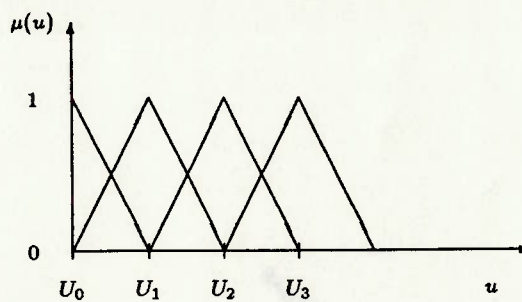
$$x_0 = a \sin \omega t, \quad (6.4)$$

a não-linearidade  $N$  se comporta como um ganho complexo,

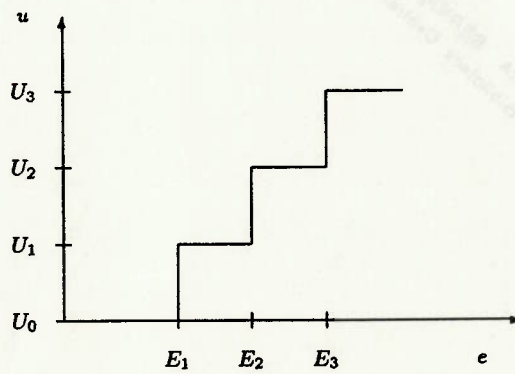
$$\eta(a, \omega) = \frac{d_1}{a} + j \frac{c_1}{a}, \quad (6.5)$$



(a)



(b)



(c)

Figura 6.1: característica entrada-saída do controlador fuzzy.



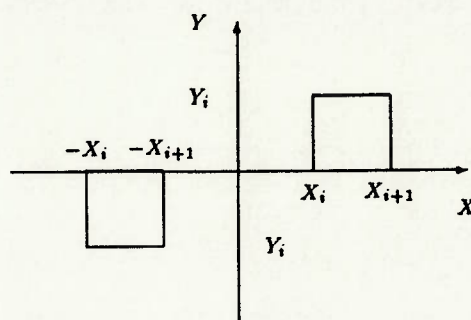


Figura 6.2: elemento não linear.

denominado *ganho complexo equivalente* da não-linearidade  $N$ ; a função  $a \rightarrow \eta(a; \omega)$  é denominada *função descritiva* de  $N$ .

Para o caso em que as não-linearidades pertencem ao primeiro e terceiro quadrantes, a função descritiva  $\eta$  é real, não-negativa e independente de  $\omega$ .

Um elemento *Multilevel-Relay* é uma não linearidade de primeiro e terceiro quadrantes como pode ser visto na figura 6.2

Cada elemento possui a seguinte função descritiva:

$$\eta(a, \omega) = \frac{4Y_i}{\pi a^2} [\sqrt{a^2 - X_i^2} - \sqrt{a^2 - X_{i+1}^2}], \quad (6.6)$$

e o elemento final com zona morta para o último nível  $Y_Q$  possui a função descritiva:

$$\eta(a, \omega) = \frac{4Y_Q}{\pi a^2} \sqrt{a^2 - X_Q^2}. \quad (6.7)$$

Portanto um *Multilevel-Relay* genérico com  $Q$  níveis de entrada  $X_1 \dots X_Q$  e  $Q$  correspondentes níveis de saída  $Y_1 \dots Y_Q$  possui a sua função descritiva definida por,

$$\eta(a, \omega) = \frac{4}{\pi a^2} \left\{ \sum_{i=1}^{Q-1} Y_i [\sqrt{a^2 - X_i^2} - \sqrt{a^2 - X_{i+1}^2}] + Y_Q \sqrt{a^2 - X_Q^2} \right\}. \quad (6.8)$$

Considere o sistema não linear com realimentação mostrado na figura 6.3. O objetivo é determinar se na ausência de um sinal de entrada é possível obter uma solução periódica não nula para  $x(\cdot)$ . Se os dois blocos  $N$  e  $G$  são descritos pelas seguintes equações:

$$y(t) = n[x(t)], \quad (6.9)$$

$$z(t) = \int_0^t g(t - \tau)y(\tau)d\tau, \quad (6.10)$$



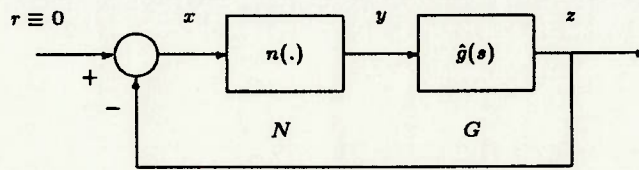


Figura 6.3: sistema de controle não linear.

dessa forma, com entrada nula, o sistema em malha fechada é descrito pela seguinte equação,

$$x(t) = - \int_0^t g(t - \tau) n[x(\tau)] d\tau. \quad (6.11)$$

O problema consiste em determinar se 6.11 possui uma solução não nula. Assumindo que 6.11 possui uma solução periódica da forma,

$$x(t) = a \sin(\omega t). \quad (6.12)$$

a condição para a ocorrência de oscilações é dada pela seguinte equação no domínio da frequência envolvendo as variáveis  $(a, \omega)$ ,

$$1 + \hat{g}(j\omega) \eta(a, \omega) = 0. \quad (6.13)$$

Na prática, a técnica de funções descritivas trabalha bem se  $|\hat{g}(j\omega)|$  decresce rapidamente quando  $\omega \rightarrow \infty$ , ou seja quando o bloco  $G$  se comporta como um filtro passa-baixas.

### 6.3 Critério de Estabilidade do Círculo

A aplicação do critério de estabilidade do círculo ou estabilidade  $L_2$  associado a controladores fuzzy foi proposto por Ray et alii [79,80]. Essa técnica só pode ser aplicada quando o controlador fuzzy pode ser considerado uma não linearidade limitada a um setor (*Sector Bound Non-Linearity*) e se encontra associado a uma planta linear invariante no tempo. Em particular, utiliza-se um controlador fuzzy do tipo *Multilevel-Relay* sem memória como no item anterior.

A figura 6.4 ilustra o sistema considerado. A função  $c(t) = h(t, e(t))$  representa o controlador fuzzy e  $g(t)$  representa uma planta linear invariante no tempo.

**Definição 6.1** Uma função contínua  $h : \mathcal{R}_+ \times \mathcal{R} \rightarrow \mathcal{R}$  é dita pertencente ao setor  $[\alpha, \beta]$ ,  $\alpha \leq \beta$ , se

$$\begin{aligned} h(t, 0) &= 0 \quad \forall t \geq 0, \\ \alpha \leq \frac{h(t, e)}{e} \leq \beta \quad &\forall t \geq 0, \forall x \neq 0. \end{aligned} \quad (6.14)$$

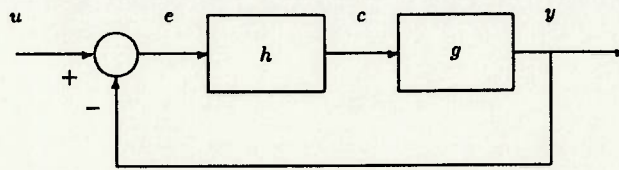


Figura 6.4: sistema com realimentação.

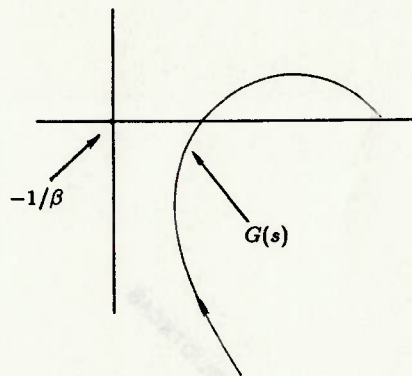


Figura 6.5: representação gráfica para o critério de estabilidade do círculo.

**Teorema 6.1** *Seja*

$$G(j\omega) = \int_0^{\infty} g(t) \exp^{-j\omega t} dt, \quad -\infty < \omega < \infty, \quad (6.15)$$

*o sistema é  $L_2$  estável se a seguinte condição é satisfeita para  $\alpha = 0$ ,*

$$\operatorname{Re}[G(j\omega)] > \frac{-1}{\beta} \quad \forall \omega. \quad (6.16)$$

A figura 6.5 é a representação gráfica para o teorema proposto.

## 6.4 Função de Lyapunov

### 6.4.1 Critério de Estabilidade

Para a elaboração do conceito de Função de Lyapunov, são formulados aqui dois critérios para a estabilidade de um sistema de controle fuzzy. Inicialmente algumas definições preliminares são necessárias [103].

**Definição 6.2** *uma função contínua  $\alpha : \mathcal{R} \rightarrow \mathcal{R}$  é dita pertencente a classe  $K$  se,*

1.  $\alpha(\cdot)$  é não-decrescente,
2.  $\alpha(0) = 0$  e
3.  $\alpha(p) > 0$  sempre que  $p > 0$ .

**Definição 6.3** *Uma função contínua  $V : \mathcal{R}_+ \times \mathcal{R}^n \rightarrow \mathcal{R}$  é uma função localmente definida positiva (lpdf) se existe uma função contínua não decrescente  $\alpha : \mathcal{R} \rightarrow \mathcal{R}$  tal que,*

1.  $\alpha(0) = 0, \alpha(p) > 0$  sempre que  $p > 0$ ,
2.  $V(t, 0) = 0 \forall t \geq 0$ ,
3.  $V(t, x) \geq \alpha(x) \forall t \geq 0, \forall x \in B_r$ , onde  $B_r$  é uma região de raio  $r$  e centro  $0$ .

**Definição 6.4** *Uma função contínua  $V : \mathcal{R}_+ \times \mathcal{R}^n \rightarrow \mathcal{R}$  é dita decrescente se existe uma função classe  $K, \beta(\cdot)$  tal que,*

$$V(t, x) \leq \beta(x), \quad (6.17)$$

$\forall t \geq 0, \forall x \in B_r \subset \mathcal{R}^n$ , para algum  $r > 0$ .

Os seguintes teoremas de estabilidade são extensões dos teoremas de estabilidade de Lyapunov para análise de sistemas não lineares. Dada uma planta não linear associada a um controlador fuzzy, o sistema completo é estável segundo Lyapunov se existe uma função, denominada função de Lyapunov, que satisfaz as condições dos teoremas. A função de Lyapunov pode ser considerada como uma função de energia generalizada. A convergência de tal função de energia para zero indica a convergência das variáveis de estado e portanto a estabilidade do sistema.

**Teorema 6.2 (Estabilidade)** *Considere uma planta  $P$  e um controlador fuzzy  $C$  representados por:*

$$\dot{x} = f(x, u), \quad (6.18)$$

$$u = g(x), \quad (6.19)$$

onde  $x$  é uma variável de estado,  $x \in \mathcal{R}^n$ ,  $u$  é a entrada,  $u \in \mathcal{R}^q$ .

*Se existe uma função continuamente diferenciável, decrescente lpdf  $V$ , tal que,*

$$\frac{d}{dt}V(t, x) \leq 0 \quad (6.20)$$

$\forall t \geq t_0, x \in X$ , onde  $X$  é o universo de discurso, então o equilíbrio  $0$  do sistema de controle fuzzy é uniformemente estável sobre  $[t_0, \infty)$ .

**Teorema 6.3 (Estabilidade Assintótica)** *considere uma planta não linear  $P$  e um controlador fuzzy  $C$ ,*

*Se existe uma função continuamente diferenciável, decrescente lpdf  $V$ , tal que  $-\frac{d}{dt}V$  é uma função lpdf, então o equilíbrio  $0$  do sistema de controle fuzzy é uniformemente assintoticamente estável sobre  $[t_0, \infty)$ .*

Os teoremas 6.2 e 6.3 mostram que pode-se provar a estabilidade de um sistema de controle fuzzy no sentido de Lyapunov encontrando-se uma função de Lyapunov conveniente. Tal afirmação é válida não apenas para sistemas de controle fuzzy mas também para um sistema dinâmico não linear geral. Entretanto a função  $g()$  na equação 6.19 é geralmente uma função altamente não linear. Na prática, é muito difícil encontrar uma função de Lyapunov adequada, por outro lado, é mais simples projetar o controlador fuzzy baseado numa determinada função de Lyapunov, como será visto a seguir.

### 6.4.2 Método de Projeto

Considere uma planta não linear P descrita por:

$$\dot{x} = f(x, u). \quad (6.21)$$

Suponha que  $V$  é uma função de Lyapunov escolhida por um especialista baseado no seu conhecimento do sistema. Para satisfazer as condições do teorema 6.2, é necessário resolver a seguinte inequação para as entradas admissíveis,

$$\dot{V}(x) = \frac{dV}{dx} \dot{x}, \quad (6.22)$$

$$= \frac{dV}{dx} f(x, u) \leq 0. \quad (6.23)$$

A superfície que restringe a entrada pode ser definida como,

$$U_c = \{u \mid \forall x, \frac{dV}{dx} f(x, u) \leq 0\}. \quad (6.24)$$

A superfície  $U_c$  é o conjunto de entradas que satisfaz o critério de estabilidade estabelecido pelo teorema 6.2. A superfície denotada por  $U_f$ , é o conjunto de entradas gerada pelo controlador fuzzy. No processo de construção da base de regras, um especialista pode realizar uma comparação entre  $U_c$  e  $U_f$ . Se  $U_f$  está completamente contida em  $U_c$ , então o sistema de controle fuzzy é estável de acordo com o teorema 6.2 ou o teorema 6.3.

## 6.5 Equações Relacionais

Como foi visto no capítulo 2 um sistema fuzzy pode ser representado através de equações relacionais. Uma equação relacional, é uma equação do tipo,

$$y = x \circ R, \quad (6.25)$$

onde  $x$  e  $y$  são variáveis fuzzy,  $R$  é uma relação fuzzy e  $\circ$  é um operador composicional SUP-STAR (em geral, max-min).

Nesse sentido, um sistema dinâmico simples, como o mostrado na figura 6.6, pode ser descrito pela seguinte equação relacional:

$$x_{k+1} = x_k u_k \circ R, \quad (6.26)$$



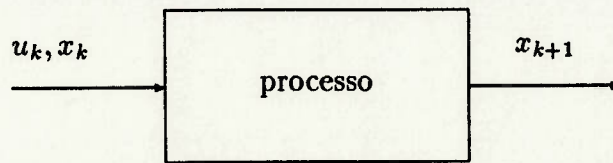


Figura 6.6: sistema dinâmico fuzzy.

onde  $x_k$  e  $x_{k+1}$  são variáveis fuzzy que assumem como valores conjuntos fuzzy definidos sobre o universo de discurso  $X$  e  $u_k$  é uma variável fuzzy no universo de discurso  $U$ .  $x_k$  é o estado corrente,  $u_k$  é a entrada corrente e  $x_{k+1}$  denota o próximo estado do processo.  $x_k u_k$  é uma relação fuzzy definida no espaço do produto cartesiano  $X \times U$  e é definido por,

$$\mu_{x_k u_k}(x, u) = \mu_{x_k}(x) \wedge \mu_{u_k}(u). \tag{6.27}$$

A relação fuzzy discreta  $R$  é definida no espaço  $X \times U \times X$  e é considerada um mapeamento para o próximo estado. O próximo estado  $x_{k+1}$  pode ser calculado como,

$$\mu_{x_{k+1}}(x_{k+1}) = \bigvee_{x_k, u_k} [\mu_{x_k u_k}(x_k, u_k) \wedge \mu_R(x_k, u_k, x_{k+1})]. \tag{6.28}$$

Sistemas MIMO também podem ser representados como em [39,53,109].

**Teorema 6.4** se  $x$  é um conjunto fuzzy discreto em  $X$  e  $Q$  é uma relação fuzzy discreta em  $X \times X$ , então,

$$(x \circ Q) \circ Q = x \circ (Q \circ Q). \tag{6.29}$$

**Teorema 6.5** se  $x$  e  $u$  são conjuntos fuzzy discretos em  $X$  e  $U$ , e  $R$  é uma relação fuzzy definida em  $R \times U \times X$  então,

$$xu \circ R = x \circ (u \circ R) = u \circ (x \circ R). \tag{6.30}$$

Considere uma resposta de 6.26 a uma entrada constante  $u = u_c$  com uma condição inicial  $x = x_0$ . Nesse caso temos,

$$x_1 = x_0 u_c \circ R, \tag{6.31}$$

$$x_2 = x_1 u_c \circ R, \tag{6.32}$$

...

...

$$x_N = x_{N-1} u_c \circ R. \tag{6.33}$$

De 6.31-6.33, temos,

$$x_2 = (x_0 u_c \circ R) \circ (u_0 \circ R), \quad (6.34)$$

$$= (x_0 \circ (u_c \circ R)) \circ (u_0 \circ R), \quad (6.35)$$

$$= (x_0 \circ Q) \circ Q \quad (Q = u_c \circ R). \quad (6.36)$$

$$(6.37)$$

Também, através do teorema 6.5, temos

$$x_2 = x_0 \circ (Q \circ Q), \quad (6.38)$$

$$= x_0 \circ Q^2. \quad (6.39)$$

Seguindo o mesmo procedimento, temos,

$$x_N = x_0 \circ Q^N. \quad (6.40)$$

De 6.40, o comportamento assintótico do sistema depende de  $Q^N$  com  $N \rightarrow \infty$ . Entretanto, foi demonstrado por Thomason [97] que as potências de qualquer matriz relacional fuzzy ou convergem para uma matriz fixa ou se torna cíclica com período finito.

Seguindo tal abordagem, tentativas tem sido realizadas no sentido de se obter uma ferramenta capaz de analisar globalmente um sistema dinâmico fuzzy (controlador + planta) de uma maneira unificada. Analogamente às teorias de controle moderno e controle clássico, conceitos como, controlabilidade, alcançabilidade, estabilidade, compensação direta, compensação realimentada, foram desenvolvidos. Contudo, os avanços nesse sentido ainda não são significativos devido principalmente à dificuldade de manipulação de tais equações.

Os trabalhos fundamentais de equações relacionais fuzzy foram desenvolvidos por Sanchez em [83] e por Pedrycz et alii em [74,75,76], utilizando conjuntos discretos e finitos. A utilização de conjuntos discretos e finitos permite a representação de relações fuzzy através de matrizes.

Existem dois problemas básicos quando se trabalha com equações relacionais [74] do tipo da equação 6.26,

1. dado  $y$  e  $R$  encontrar  $x$  e,
2. dado  $x$  e  $y$  encontrar  $R$ .

Como abordado em [74] tanto o problema (1.) como o problema (2.) admitem múltiplas soluções.

O problema (2.) é fundamentalmente o problema de identificação de sistemas fuzzy. Abordagens distintas podem ser encontradas em [108,106,82,75,76,7].

Deve-se ressaltar aqui os resultados de Xu em [107,109,110] abordando os problemas de desacoplamento através de compensação direta (*feedforward compensation*), desacoplamento através de compensação realimentada (*feedback compensation*) e análise e controle através de compensação direta e realimentação utilizando equações relacionais.



A estabilidade de sistemas fuzzy analisada através de equações relacionais foi abordado por Kania et alii em [46] e Kiszka et alii em [52]. Em particular, Kiszka desenvolveu um método baseado na medida de energia de um sistema dinâmico fuzzy. Nesse caso, o sistema é estável se a energia total decai e um estado de equilíbrio é alcançado. Uma função da medida da energia e um algoritmo heurístico para a determinação da estabilidade são propostos.

Para se discutir a estabilidade de um sistema, é essencial a definição do conceito "instável". Na teoria convencional de sistemas, "não-limitado" é sem dúvida um indicativo de instabilidade. Entretanto, esse conceito de instabilidade se torna ambíguo quando um sistema é representado por conjuntos fuzzy. Como foi observado, o estado final  $x_N$  sempre converge para um valor fixo ou se torna cíclico com período finito, que é claramente diferente das análises convencionais. Se torna óbvio que um tipo diferente de estabilidade, isto é, uma estabilidade definida através de "graus", deve ser utilizada para se analisar sistemas de controle fuzzy. O grau de estabilidade pode ser um número real ou um número fuzzy como: *muito estável*, *mais ou menos estável*. A seguir é apresentado uma medida de estabilidade proposta por Tong [100].

Considere um dado espaço de estados particionado em duas regiões,  $X_I$ , conjunto interior ou região estável,  $X_B$ , conjunto limite ou região instável. O grau de estabilidade de um estado  $x$  é dado por,

$$\sigma(x) = 1 - I(x, X_B), \quad (6.41)$$

onde  $I(x, X_B)$  denota o grau com que o conjunto fuzzy  $x$  está incluído na região instável  $X_B$ . A escolha da função  $I$  não é óbvia, duas possíveis candidatas propostas por Tong são:

$$I_1(x, X_B) = \vee(\mu_x(x) \wedge \mu_{X_B}(x)), \quad (6.42)$$

$$I_2(x, X_B) = \wedge([1 - \mu_x(x)] \vee \mu_{X_B}(x)). \quad (6.43)$$

Exemplos de duas funções de estabilidade são mostradas na figura 6.7. Os graus de estabilidade são:

$$\sigma_1(x_1) = 1.0, \quad \sigma_1(x_2) = 1 - \alpha, \quad \sigma_1(x_3) = 0.0, \quad \sigma_1(x_4) = 0.0, \quad (6.44)$$

$$\sigma_2(x_1) = 1.0, \quad \sigma_2(x_2) = 1.0, \quad \sigma_2(x_3) = \beta, \quad \sigma_2(x_4) = 0.0. \quad (6.45)$$

Deve-se observar que a teoria de sistemas dinâmicos fuzzy baseada em equações relacionais ainda é uma teoria em desenvolvimento. Apesar dos esforços realizados, nenhum resultado efetivo foi observado.

## 6.6 Uma Abordagem Baseada em Energia

O conceito de "energia" é comumente utilizado como uma medida da estabilidade de sistemas dinâmicos. Um exemplo clássico, como pudemos observar é a estabilidade no sentido de Lyapunov que utiliza a função de Lyapunov como uma função de energia generalizada

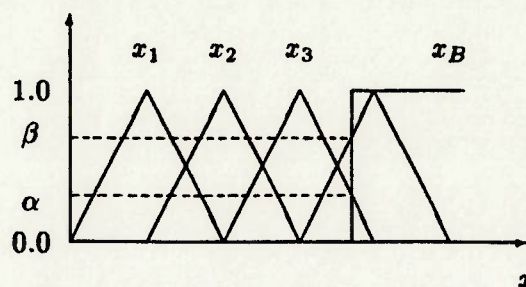


Figura 6.7: estados estáveis e instáveis.

para indicar as propriedades de estabilidade de um sistema dinâmico. A abordagem baseada em energia proposta por Kiszka, Gupta e Nikiforuk [52] é fundamentada no fato que um sistema dinâmico é estável se a energia total decresce monotonicamente até que um estado de equilíbrio é alcançado. Nesse trabalho, procura-se responder perguntas como: Um dado sistema fuzzy é estável, instável, ou oscilatório? O que causa a instabilidade de um sistema fuzzy? Como pode ser removido tal instabilidade?

A equação básica para representação de sistemas dinâmicos fuzzy possui a seguinte forma:

$$x_{k+1} = x_k \circ u_k \circ R, \quad k = 0, 1, \dots \quad (6.46)$$

onde  $x_k$  e  $x_{k+1}$  são variáveis de estado, que assumem conjuntos fuzzy como seus valores, no instante  $k$  e  $k + 1$  respectivamente.

Portanto, como foi visto na seção 6.5 a resposta do sistema a uma entrada constante é dada por,

$$x_n = x_0 \circ Q^N, \quad (6.47)$$

onde  $Q = u_c \circ R$ , e  $u_c$  é uma constante fuzzy.

É intuitivamente verdadeiro que um sistema dinâmico é estável se a energia total decresce monotonicamente até que um estado de equilíbrio é alcançado. Isso também se aplica a um sistema de controle fuzzy já que pertence a classe de sistemas dinâmicos não lineares. Entretanto, o ponto principal consiste em como definir uma "função generalizada de energia" que esteja relacionada a sistemas de controle fuzzy. Através de comparações e observações Kiszka et alii concluíram que a função de energia desejada para um conjunto fuzzy depende da sua forma e do seu conjunto suporte. Dois conjuntos fuzzy denominados *medium* são mostrados na fig. 6.8,  $x$  e  $x'$  tem o mesmo ponto de máximo, mas  $x'$  tem um conjunto suporte maior. Portanto espera-se que,

$$E(x' = \text{medium}) > E(x = \text{medium}) \quad (6.48)$$

onde  $E()$  denota a função de energia para um conjunto fuzzy.



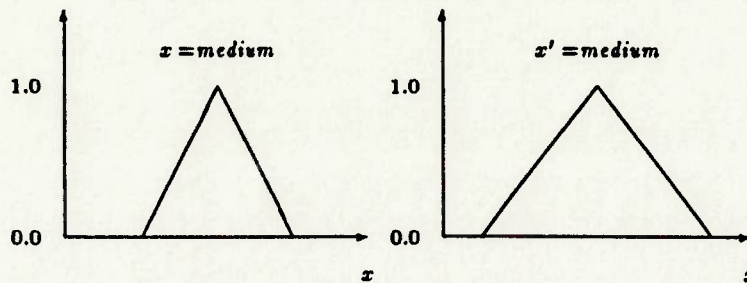


Figura 6.8: dois conjuntos fuzzy denominados *medium*.

Uma função de energia generalizada é formulada, levando-se em conta a forma e o suporte da função de pertinência do conjunto fuzzy,

$$E(x) = \frac{1}{n} \sum_{i=1}^n w(x_i) f(\mu_x(x_i)), \quad (6.49)$$

onde  $w()$  é uma função relacionada com o suporte de  $x$  para a função de energia  $E$ ,  $f()$  é uma função que relaciona a forma da função de pertinência de  $x$  para  $E$ ,  $n$  é a cardinalidade de  $x$ , e  $1/n$  é o fator de normalização.

A definição 6.49 é heurística e não possui nenhuma justificativa teórica. Também não é claro como escolher as funções  $w$  e  $f$ . Kiszka et alii simplificaram 6.49 utilizando as seguintes funções,

$$w(x) = x, \quad (6.50)$$

$$f(x) = x, \quad (6.51)$$

isto é, 6.49 se torna,

$$E(x) = \frac{1}{n} \sum_{i=1}^n x_i \mu_x(x_i), \quad (6.52)$$

Agora considere um sistema dinâmico fuzzy descrito pela seguinte equação relacional,

$$x_{k+1} = x_k \circ Q, \quad (6.53)$$

cuja solução é dada por 6.47 quando o estado inicial  $x_0$  é dado. Pode-se definir agora a função de energia do estado a cada instante  $k$  como,

$$E(x_k) = E(x_0 \circ Q^k), \quad k = 1, 2, \dots \quad (6.54)$$

Uma função característica da energia é também dada por

$$\Delta E_k = E(x_0 \circ Q^k) - E(x_0 \circ Q^{k-1}) \quad k = 1, 2, \dots \quad (6.55)$$

Baseada na intuição e experiência, Kiszka et alii desenvolveram algumas regras para a estabilidade de um sistema dinâmico fuzzy,

1. um sistema dinâmico fuzzy descrito por 6.53 é denominado *estável se*,

$$\Delta E_k \leq 0, \quad k \rightarrow \infty. \quad (6.56)$$

2. O sistema dinâmico fuzzy descrito por 6.53 é denominado *instável se*,

$$\Delta E_k \geq 0, \quad k \rightarrow \infty. \quad (6.57)$$

3. O sistema dinâmico fuzzy descrito por 6.53 é denominado *oscilatório* com um período  $\tau$  se,

$$|\Delta E_k| = |\Delta E_{k+\tau}|, \quad k \rightarrow \infty. \quad (6.58)$$

onde  $|*|$  indica o valor absoluto.

## 6.7 Método de Mapeamento Célula-a-Célula

Considere um sistema dinâmico governado pela seguinte equação:

$$\dot{x} = f(t, x(t)), \quad (6.59)$$

onde  $x$  é um vetor de dimensão  $N$  e  $f$  uma função vetorial real. É possível integrar a equação 6.59 sobre um período  $\tau$  e derivar uma relação entre o estado no instante  $t$  e o estado no instante  $t + \tau$ . Em outras palavras, pode-se representar um sistema dinâmico através de uma nova equação,

$$x(k+1) = G(x(k)), \quad (6.60)$$

onde  $G$  é o mapeamento ponto-a-ponto para o próximo estado. Esse tipo de mapeamento ponto-a-ponto representando um sistema dinâmico é denominado *mapa de pontos* ou *mapa de Poincare*. Hsu [41,42,43] aplicou esse método para analisar uma certa classe de sistemas dinâmicos altamente não lineares e posteriormente desenvolveu uma metodologia, denominada *mapeamento célula-a-célula*. O método de Hsu provou ser bastante efetivo para o tratamento de fenômenos como os de bifurcações em sistemas não lineares.

### 6.7.1 Espaço de Estados Representados por Células

Considere um sistema dinâmico governado pela equação 6.59. O eixo de coordenadas das variáveis de estado  $x_i$ ,  $i = 1, \dots, N$  pode ser particionado em um número de intervalos de tamanho  $h_i$ ,  $i = 1, \dots, N$ . O intervalo  $z_i$  da variável de estado  $x_i$  é definido como sendo aquele que contém todo  $x_i$ , tal que

$$(z_i - \frac{1}{2})h_i \leq x_i \leq (z_i + \frac{1}{2})h_i. \quad (6.61)$$

A  $N$ -upla  $(z_1, \dots, z_N)$  é denominada *célula* e denotada por  $z$ . Cada ponto  $x$  no espaço de estados pertence a alguma célula  $z$ . O conjunto de todas as células, denominado *Espaço de Estados Celular* é normalmente denotado por  $Z$ . Algumas definições básicas são dadas a seguir:

**Definição 6.5** Uma célula  $z'$  é dita ser *contígua* a uma célula  $z$  na direção  $Z_i$  se,

$$z' - z = +e_i, -e_i, \quad (6.62)$$

onde  $e_i$  é o vetor unitário na direção  $Z_i$ .

**Definição 6.6** Uma célula  $z' \neq z$  é dita ser *vizinha* a célula  $z$  se,

$$|z'_i - z_i| \leq 1, \quad \text{para } i = 1, \dots, N. \quad (6.63)$$

Portanto, uma célula contígua a  $z$  também é uma célula vizinha a  $z$ .

### 6.7.2 Mapeamento Célula-a-Célula

Para o sistema dinâmico dado pela equação 6.59, supõe-se que o sistema é autônomo (invariante no tempo), isto é,

$$\dot{x}(t) = f(t, x(t)), \quad (6.64)$$

$$= f(x). \quad (6.65)$$

Portanto, dada uma condição inicial  $x(t_0)$  e um período de tempo  $\tau$ , o estado em  $t = t_0 + \tau$  pode ser calculado como,

$$x = (t_0 + \tau) = x(t_0) + \int_{t_0}^{t_0 + \tau} \dot{x}(t) dt. \quad (6.66)$$

Como o sistema é autônomo, pode-se fazer  $t_0 = 0$  como sendo o tempo inicial. Pode-se também fazer  $x(t_0) = x_0$  e  $x(t_0 + \tau) = x_\tau$  e então reescrever 6.66 como,

$$x_\tau = x_0 + \int_0^\tau \dot{x} dt. \quad (6.67)$$

Assuma que o espaço de estados possa ser particionado em um determinado número de células para a formação de um espaço de estados celular  $Z$ . Para cada célula  $z \in Z$ , representamos  $z$  através do seu ponto central  $x_z^c$ . Dado uma condição inicial  $x_0 = x_z^c$  para uma célula  $z$  e um período de tempo  $\tau$ , o próximo estado  $x_\tau$  é calculado como,

$$x_\tau = x_z^c + \int_0^\tau \dot{x} dt. \quad (6.68)$$

Então, pode-se definir um mapeamento da célula  $z$  para a célula  $z'$  se



$$x_r \in z'. \quad (6.69)$$

O procedimento acima pode ser realizado para cada célula do espaço de estados. O resultado será um mapeamento célula-a-célula para o próximo estado denotado por  $F$ , isto é,

$$z(k+1) = F(z(k)). \quad (6.70)$$

Esse mapeamento é construído para aproximar o comportamento dinâmico do sistema original. Quanto menor é o tamanho das células melhor é a aproximação.

Antes de apresentar o método para extrair o comportamento dinâmico do mapeamento célula-a-célula, algumas definições relacionadas às propriedades dinâmicas serão enunciadas.

**Definição 6.7** Uma célula  $z^*$  que satisfaz

$$z^* = F(z^*), \quad (6.71)$$

é dita célula de equilíbrio no mapeamento  $F$ .

**Definição 6.8** Uma seqüência de  $k$  distintas células  $z^*(1), \dots, z^*(k)$  que satisfaz,

$$z^*(m+1) = F^m(z^*(1)), \quad m = 1, \dots, k-1 \quad (6.72)$$

$$z^*(1) = F^k(z^*(1)), \quad (6.73)$$

é dita constituir um movimento periódico de período  $k$  no mapeamento  $F$ . O movimento é denominado movimento  $p$ - $k$  e as células são denominadas células  $p$ - $k$ .

**Definição 6.9** Uma célula  $z$  é dita removida com passo- $r$  de um movimento  $p$ - $k$  se  $r$  é o mínimo inteiro tal que

$$F^r(z) = z^*(j), \quad (6.74)$$

$z^*(j)$  é uma das células  $p$ - $k$  num movimento  $p$ - $k$ .

**Definição 6.10** Para um movimento  $p$ - $k$ , o conjunto de células que são passo- $q$ ,  $q \leq r$ , é denominado domínio de atração passo- $r$  de um movimento  $p$ - $k$ . O domínio de atração total ou simplesmente o domínio de atração de um movimento  $p$ - $k$  é o domínio de atração passo- $r$  com  $r$  tendendo a infinito.

**Definição 6.11** Para um mapeamento célula-a-célula  $F$ , assumamos que o domínio de interesse no sub-espaço  $X_j$  esteja limitado a  $[lb_j, ub_j]$ ,  $j = 1, \dots, n$ . As células que pertencem ao domínio de interesse são denominadas células regulares. A região que não pertence ao domínio de interesse é considerada como uma única célula e é denominada célula exterior.

Após a partição do espaço de estados, o espaço de estados celular consiste de um número de células regulares e uma célula exterior. As células regulares são rotuladas através de números inteiros positivos  $1, \dots, N_c$  e a célula exterior é numerada com  $N_c + 1$ . Já que não há interesse em saber informações a respeito do estado do sistema dinâmico fora



do domínio especificado, o mapeamento célula-a-célula nessa região pode ser representado por,

$$F(N_c + 1) = N_c + 1. \quad (6.75)$$

Isso significa que uma célula exterior é uma célula p-1 e o sistema permanecerá na célula exterior após a sua primeira entrada.

A evolução do mapeamento iniciando com uma célula regular  $z$  pode conduzir a três situações:

1. a célula  $z$  pertence a um movimento p- $k$ , isto é, a célula  $z$  pode ser novamente atingida em  $k$  passos,
2. a célula  $z$  é mapeada num movimento p- $k$  em  $r$  passos, isto é, a célula  $z$  pertence ao domínio de atração de passo- $r$  de um movimento periódico,
3. a célula  $z$  é mapeada na célula exterior em  $r$  passos, isto é, a célula  $z$  pertence ao domínio de atração de passo- $r$  da célula exterior.

### 6.7.3 Algoritmo de Agrupamento

Deve-se processar o espaço de estados celular para se encontrar os movimentos periódicos e os domínios de atração através de um algoritmo de agrupamento. O algoritmo se utiliza de buscas exaustivas. Cada célula  $z$  deve ser assinalada com três números característicos. Um número de grupo  $G(z)$  é assinalado a cada célula em um movimento periódico e a cada célula em seu domínio de atração. O número de passo  $S(z)$  indica o número de passos para que  $z$  seja mapeado em um movimento periódico. Se  $z$  é uma célula periódica então  $S(z) = 0$ . O período de um movimento periódico é denotado por um número de periodicidade. O mesmo número de periodicidade  $P(z)$  é assinalado a todas as células de um grupo.

Para distinguir as células em diferentes estágios, elas são divididas em *células virgens*, *células em processamento* e *células processadas*. O número de grupo de todas as células virgens são setadas para zero no início do algoritmo. Quando uma célula virgem tem seu processamento iniciado o seu número de grupo deve ser trocado para  $-1$ , indicando ser uma célula em processamento. Após o processamento, um número de grupo deve ser assinalado e então a célula passa a ser denominada *célula processada*.

No início, já que a célula exterior é uma célula p-1, ela é considerada o primeiro movimento periódico e faz-se  $G(N_c + 1) = 1$ ,  $S(N_c + 1) = 0$  e  $P(N_c + 1) = 1$ . Então, pode-se processar as células regulares  $z = 1, \dots, N_c$  seqüencialmente.

Vamos considerar uma célula virgem  $z$  e a seqüência de processamento como:

$$z \rightarrow F(z) \rightarrow F^2(z) \rightarrow \dots \rightarrow F^i(z). \quad (6.76)$$

A cada passo, existem três condições possíveis:

1. a nova célula gerada  $F^i(z)$  é uma célula virgem. Nesse caso, o número de grupo da célula deve ser trocado de 0 para  $-1$ , o que indica que essa célula se encontra sob processamento, e prossegue-se para a próxima célula  $F^{i+1}(z)$ .

2. a nova célula gerada  $F^i(z)$  é uma célula processada com um número de grupo positivo. Nesse caso, a corrente seqüência em processamento está mapeada em uma célula cujas propriedades globais são conhecidas (número de grupo, número de passo e número de periodicidade). É claro que a seqüência obtida deve ser assinalada com o mesmo número de grupo e o mesmo número de periodicidade que  $F^i(z)$ . O número de passo para as células na seqüência pode ser calculada como:

$$S(F^j(z)) = S(F^i(z)) + i - j, \quad j = 0, 1, \dots, i - 1. \quad (6.77)$$

A seguir deve-se escolher uma nova célula virgem e iniciar uma nova seqüência de processamento.

3. a nova célula gerada  $F^i(z)$  é uma célula em processamento. Nesse caso, um novo movimento periódico foi descoberto. Todas as células da seqüência são assinaladas com um novo número de grupo. Assuma que a célula seja a  $(j + 1)$ -ésima célula da seqüência 6.76, isto é,  $F^i(z) = F^j(z), j < i$ . Então o número de periodicidade é  $(i - j)$  para esse movimento periódico é assinalado a cada célula da seqüência. O número de passo também pode ser determinado por:

$$S(F^k(z)) = j - k, \quad k = 0, 1, \dots, j - 1, \quad (6.78)$$

$$S(F^k(z)) = 0, \quad k = j, j + 1, \dots, i - 1. \quad (6.79)$$

Ao final, deve-se escolher uma nova célula virgem e iniciar um novo processamento. O processo deve ser repetido até que todas as células sejam processadas.

Em resumo, o algoritmo de agrupamento divide o espaço de estados celular em diferentes grupos consistindo de movimentos periódicos e seus domínios de atração. Cada célula no espaço de estados pertence a um grupo. O número de passo de cada célula indica a distância de cada célula do movimento periódico que a atrai.

#### 6.7.4 Análise Global de Sistemas Dinâmicos Fuzzy

O método de mapeamento célula-a-célula pode ser aplicado a um sistema dinâmico fuzzy através da aplicação dos operadores de transformação escalar-fuzzy e de transformação fuzzy-escalar para obtenção de um mapeamento real do sistema.

Dado um estado inicial real,

$$x = x_0, \quad (6.80)$$

onde  $x$  é a variável de estado e  $x_0 \in X$  é o valor inicial para  $x$ , identifica-se a célula  $z$  tal que  $x_0 \in z$ . O comportamento do sistema a partir do estado inicial  $x_0$  pode então ser entendido através do número de grupo, número de passo e do número de periodicidade associado a célula correspondente. Além disso, uma análise global pode ser realizada através da análise dos domínios de atração dos movimentos periódicos com diferentes números de passo.

As figuras 6.9 e 6.10 são os resultados obtidos por Chen [23,24] para o problema de controle de pêndulo invertido apresentado no capítulo 5 considerando a posição angular

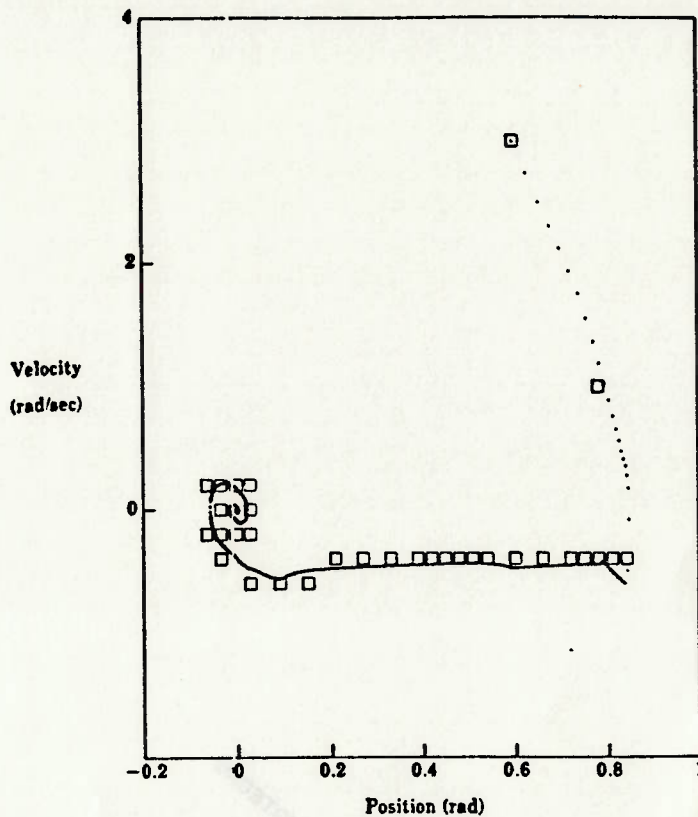


Figura 6.9: trajetórias para o pêndulo invertido e o seu mapeamento célula-a-célula com estados iniciais  $x_1 = 0.6$  e  $x_2 = 3.0$  (quadrados para o mapeamento célula-a-célula e pontos para a trajetória real). A trajetória real converge para a origem enquanto que o mapeamento célula-a-célula converge para um movimento p-8 em torno da origem [23,24].

( $x_1$ ) e a velocidade angular ( $x_2$ ) como variáveis de estado. Foram utilizados  $101 \times 101$  blocos resultando num total de 10201 células cobrindo de  $-1.5$  a  $+1.5$  rad para a posição angular e de  $-10.0$  a  $+10.0$  rad/s para a velocidade angular.

O método de mapeamento célula-a-célula como descrito até aqui prevê o estado inicial como um número real. Chen [23,24] desenvolveu um método que permite a adoção de números fuzzy como estados iniciais. Nesse caso, o comportamento do sistema é descrito através de números de grupo fuzzy. A seguir descreve-se o método desenvolvido por Chen.

Por simplicidade, vamos considerar apenas o espaço bidimensional  $X = X_1 \times X_2$ . Suponha que os estados iniciais fuzzy sejam,

$$\bar{x}_1 = \bar{A}_1, \quad (6.81)$$

$$\bar{x}_2 = \bar{A}_2, \quad (6.82)$$

onde  $\bar{x}_1$  e  $\bar{x}_2$  são variáveis fuzzy e  $\bar{A}_1$  e  $\bar{A}_2$  são conjuntos fuzzy nos sub-espacos  $X_1$  e  $X_2$ . Inicialmente definimos os conjuntos nível- $\alpha$  de  $\bar{x}_1$  e  $\bar{x}_2$  como,



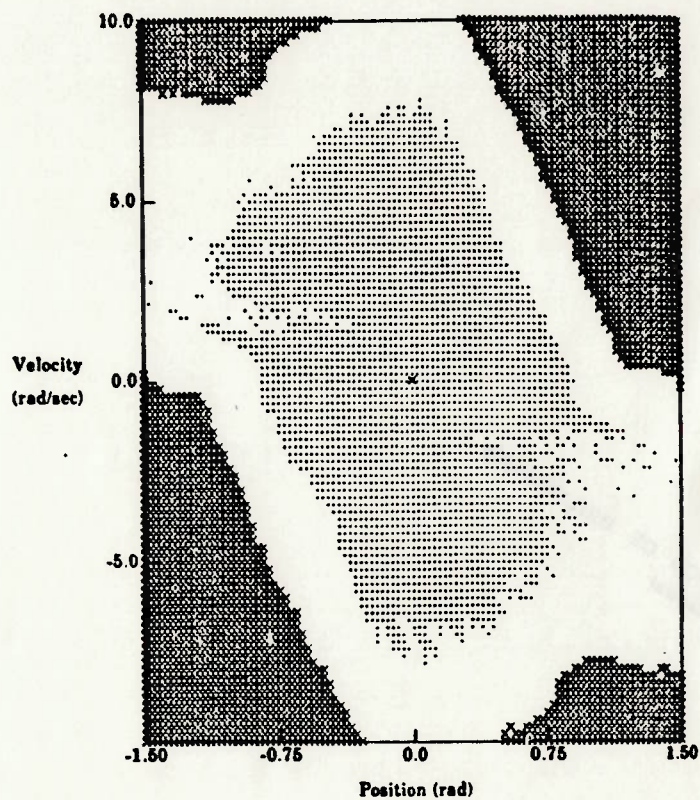


Figura 6.10: domínios de atração de passo-20 para o movimento p-8. As áreas mais escuras se referem aos domínios de atração da célula externa [23,24].



$$A_1^\alpha = \{\bar{x}_1 \in X_1 \mid \mu_{\bar{x}_1}(x_1) \geq \alpha\}, \quad (6.83)$$

$$A_2^\alpha = \{\bar{x}_2 \in X_2 \mid \mu_{\bar{x}_2}(x_2) \geq \alpha\}, \quad (6.84)$$

onde  $A_1^\alpha \subset X_1$  e  $A_2^\alpha \subset X_2$  são conjuntos reais e não conjuntos fuzzy.

Então, denota-se a região nível- $\alpha$  delimitada pelos conjuntos nível- $\alpha$   $A_1^\alpha$  e  $A_2^\alpha$  como,

$$R^\alpha(x_1, x_2) = \{(x_1, x_2) \mid x_1 \in A_1^\alpha, x_2 \in A_2^\alpha\}. \quad (6.85)$$

Cada ponto em  $R^\alpha$  tem um valor de pertinência maior que  $\alpha$  tanto na direção  $X_1$  quanto na direção  $X_2$ . Entretanto, como a menor unidade no espaço de estados celular é uma célula,  $R^\alpha$  deve ser aproximado através de um conjunto de células. Dessa forma, define-se o conjunto mínimo de células de  $R^\alpha$  como,

$$R_m^\alpha(x_1, x_2) = \min\{z \in Z \mid \forall x \in R^\alpha(x_1, x_2), \exists z, \mid x \in z\}. \quad (6.86)$$

Em outras palavras,  $R_m^\alpha$  é o menor conjunto de células que contém  $R^\alpha$ .

Agora, considere uma seqüência decrescente positiva  $\alpha$

$$\alpha_1, \alpha_2, \dots, \alpha_k \quad \text{geralmente } \alpha_1 = 1. \quad (6.87)$$

Pode-se derivar facilmente  $R^{\alpha_i}$  e  $R_m^{\alpha_i}$ ,  $i = 1, \dots, k$  de 6.85 e 6.86. O problema será como no domínio real se todas as células em  $R_m^{\alpha_i}$  pertencerem a um único grupo. Entretanto, esse não é necessariamente o caso.

Assuma que existam  $1, 2, \dots, g$  movimentos periódicos como resultado do algoritmo de agrupamento. Cada célula no nosso domínio de interesse pertence a um dos domínios de atração de movimentos periódicos. Portanto, cada célula em  $R_m^{\alpha_i}$  também pertence a um dos grupos. Também definimos  $N_{R_m^{\alpha_i}}(i) =$  número de células em  $R_m^{\alpha_i}$  que pertence ao  $i$ -ésimo grupo,  $i = 1, \dots, g$ .

A tabela 6.1 mostra o número de células que pertencem a diferentes grupos nos conjuntos de células mínimos  $R_m^{\alpha_i}$ ,  $i = 1, 2, \dots, K$ . A soma dos elementos em cada linha são o número total de células em  $R_m^{\alpha_i}$  para  $i = 1, \dots, g$ .

A tabela 6.2 mostra as proporções de células em diferentes grupos em cada conjunto de células mínimo. A soma dos elementos em cada linha é igual a um. O número de proporções é definido como,

$$P_{ij} = \frac{N_{R_m^{\alpha_i}}(j)}{\sum_{k=1, \dots, g} N_{R_m^{\alpha_i}}(k)}, \quad i = 1, \dots, k, \quad j = 1, \dots, j. \quad (6.88)$$

Finalmente define-se o número de grupo fuzzy  $\bar{G}$  dos estados iniciais  $\bar{x}_1$  e  $\bar{x}_2$  como,

$$\bar{G}(\bar{x}_1, \bar{x}_2) = 1/m_1 + 1/m_2 + \dots + g/m_g, \quad (6.89)$$

$$m_j = \max_{i=1, \dots, k} \min(\alpha_i, P_{ij}), \quad (6.90)$$

onde  $\alpha_i$  e  $P_{ij}$  são definidos em 6.87 e 6.88. Com os estados iniciais fuzzy  $\bar{x}_1$  e  $\bar{x}_2$ , o número de grupo fuzzy  $\bar{G}$  denota a distribuição de possibilidade do estado final do sistema entre diferentes grupos.

Número do Grupo				
$\alpha$	1	2	...	$g$
$\alpha_1$	$N_{R_m^{\alpha_1}}(1)$	$N_{R_m^{\alpha_1}}(2)$	...	$N_{R_m^{\alpha_1}}(g)$
$\alpha_2$	$N_{R_m^{\alpha_2}}(1)$	$N_{R_m^{\alpha_2}}(2)$	...	$N_{R_m^{\alpha_2}}(g)$
$\vdots$	$\vdots$	$\vdots$	...	$\vdots$
$\alpha_k$	$N_{R_m^{\alpha_k}}(1)$	$N_{R_m^{\alpha_k}}(2)$	...	$N_{R_m^{\alpha_k}}(g)$

Tabela 6.1: número de células dos grupos em diferentes conjuntos de células mínimo.

Número do Grupo				
$\alpha$	1	2	...	$g$
$\alpha_1$	$P_{11}$	$P_{12}$	...	$P_{1g}$
$\alpha_2$	$P_{21}$	$P_{22}$	...	$P_{2g}$
$\vdots$	$\vdots$	$\vdots$	...	$\vdots$
$\alpha_k$	$P_{k1}$	$P_{k2}$	...	$P_{kg}$

Tabela 6.2: Proporções de células dos grupos em diferentes conjuntos de células mínimo.

## 6.8 Conclusão

Esse capítulo apresentou algumas das técnicas propostas para a análise de controladores fuzzy.

O problema da análise formal de controladores fuzzy, que se refere basicamente aos conceitos de estabilidade e robustez, tem atraído a atenção de muitos pesquisadores. O grande desafio, é o estabelecimento de uma teoria de sistemas dinâmicos fuzzy, semelhante a teoria de controle moderno, e que permita analisar o comportamento global de sistemas de controle fuzzy.

Em geral, as técnicas propostas são adaptações de ferramentas utilizadas em análise de sistemas não lineares. A técnica de *Funções Descritivas* e o *Critério de Estabilidade do Círculo* foram duas das primeiras técnicas propostas. Ambas as técnicas restringem sua utilização a controladores fuzzy do tipo *Multilevel-Relay* associados plantas lineares.

O método da *Função de Lyapunov* utilizada no projeto do controle do pêndulo invertido é bastante difícil de ser aplicada como pôde-se observar.

As técnicas relacionadas às *Equações relacionais* parecem ser as mais promissoras, já que permitem uma análise global do sistema como um sistema dinâmico fuzzy de maneira semelhante à teoria de controle moderno. Conceitos como estabilidade, alcançabilidade, controlabilidade, compensação direta, etc. tem sido desenvolvidos. Contudo, os avanços nesse sentido ainda não permitem utilização prática.

No presente momento o método de *Mapeamento célula-a-célula* desenvolvido por Hsu [41] e adaptado por Chen [23] para sistemas de controle fuzzy parece ser o mais efetivo. O trabalho de Chen apresentado nesse capítulo foi direcionado para a análise e estabilização

entretanto, essa mesma técnica permite realizar a otimização de controladores fuzzy como pode ser observado no trabalho de Smith e Comer [87].



# Capítulo 7

## Observações Finais

Esse trabalho teve como objetivo organizar alguns resultados importantes da técnica de controle fuzzy relacionados ao projeto e análise de controladores.

Os conceitos básicos da teoria de conjuntos e lógica fuzzy relevantes para o desenvolvimento de controladores fuzzy, a arquitetura de um controlador fuzzy e os parâmetros envolvidos no seu projeto foram apresentados.

Foi estabelecido uma relação entre controladores fuzzy e controladores lineares do tipo PID (P,PI,PD). Como pôde ser observado uma manipulação adequada das partes que compõem o controlador (transformação escalar-fuzzy, base de conhecimento, mecanismo de inferência e transformação fuzzy-escalar) permitem a obtenção de um controlador aproximadamente linear ou linear num caso limite. Os resultados apresentados permitem explicar o bom desempenho de controladores fuzzy do tipo PI.

O desempenho de controladores fuzzy como elementos não lineares de controle foram apresentados através de dois estudos de caso.

No controlador fuzzy projetado para o sistema de primeira-ordem com tempo morto foi utilizado uma base de regras desenvolvida por Yamazaki e Sugeno [115]. Para uma mesma base de conhecimento foram utilizados algoritmos distintos, citados em literatura, envolvendo escolhas diferentes para o conectivo e , o conectivo também e a função de implicação fuzzy. No segundo caso, projetou-se um controlador fuzzy para um pêndulo invertido, sistema altamente instável, utilizando o auxílio de uma função de Lyapunov. Os desempenhos obtidos foram bastante satisfatórios.

Técnicas propostas na literatura para a análise de controladores fuzzy foram investigadas. As técnicas apresentadas foram as seguintes: *Funções Descritivas*, *Critério de Estabilidade do Círculo*, *Funções de Lyapunov*, *Equações Relacionais Fuzzy* e *Método de Mapeamento Célula-a-Célula*.

Os resultados aqui apresentados, permitem determinar algumas diretrizes gerais de projeto ligados principalmente à construção da base de conhecimento e a adoção de mecanismos de inferência adequados. Algumas técnicas de análise relacionadas à determinação de estabilidade e robustez de sistemas fuzzy também foram discutidas.

Sem dúvida, o desenvolvimento de novas abordagens que levem a uma *metodologia* de projeto de controladores fuzzy se faz necessária, dessa forma vamos concluir esse trabalho apresentando duas abordagens distintas e que representam direções importantes para o desenvolvimento de trabalhos futuros:



1. O trabalho recente de Smith e Comer [87] utilizando mapeamento célula-a-célula para a otimização do controle de posição de um motor DC sob o ponto de vista do tempo (*time optimal control*), vem confirmar a eficiência dessa técnica desenvolvida por Hsu [41] para a análise topológica de sistemas não lineares e aplicado inicialmente por Chen [23] para a análise global de sistemas fuzzy. Smith e Comer adaptaram o algoritmo de controle ótimo desenvolvido por Hsu. Os dados gerados pelo algoritmo de Hsu são utilizados para a escolha dos parâmetros  $(c_{j,0}, c_{j,1}, c_{j,2})$  que definem regras do tipo,

$$R_j : \text{se } x_1 = A_{j,1} \text{ e } x_2 = A_{j,2} \text{ então } y_j = c_{j,0} + c_{j,1}x_1 + c_{j,2}x_2.$$

Para a escolha de tais parâmetros utiliza-se o algoritmo de Widrow-Hopf de minimização de erro quadrático.

A utilização da técnica de mapeamento célula-a-célula se encontra ainda no seu início, mas certamente deve atrair maior atenção dos pesquisadores pelo fato de ser a única técnica que permite simultaneamente o projeto e a análise de controladores fuzzy.

2. Um dos trabalhos mais significativos na área de sistemas fuzzy foi desenvolvido por Kosko [54] ao combinar redes neurais e sistemas fuzzy. Uma nova abordagem geométrica para sistemas fuzzy foi desenvolvida. Um conjunto fuzzy pode ser entendido como um ponto dentro de um hipercubo. A geometria de conjuntos fuzzy envolve o domínio  $X = \{x_1, \dots, x_n\}$  e o intervalo  $[0, 1]$  de mapeamentos  $\mu_A : X \rightarrow [0, 1]$ . O conjunto de todos os conjuntos fuzzy corresponde ao hipercubo unitário  $I^n = [0, 1]^n$ .

Sistemas fuzzy são representados por mapeamentos  $S : I^n \rightarrow I^p$  e são denominados, FAMs (*Fuzzy Associative memories*). Uma regra do tipo se  $A_i$  então  $B_i$  é denominada regra FAM e é representada por  $(A_i, B_i)$ .

Um sistema Adaptativo, AFAM (*Adaptive Fuzzy Associative Memory*) pode ser obtido utilizando algoritmos de treinamento de redes neurais do tipo DCL (*Differential Competitive Learning*). Dessa forma, pode-se estimar as regras  $(A_i, B_i)$  e o número  $m$  de regras capazes de controlar uma planta.

# Apêndice A

## Programas de Simulação

### A.1 Introdução

Para a realização das simulações que constam nesse trabalho foram implementados 4 versões de programas em linguagem C,

1. *Simulador Básico,*
2. *Simulador para o Método Simplificado,*
3. *Simulador para o Método de Buckley e Ying, e*
4. *Simulador para o Pêndulo Invertido.*

As versões 2,3 e 4 são adaptações da versão 1. Uma descrição detalhada será aqui realizada apenas para o Simulador Básico.

### A.2 Simulador Básico

#### A.2.1 Características Gerais

O Simulador Básico foi construído para simular a dinâmica de um sistema de controle de malha fechada como o da figura A.1.

Esse programa, permite simular controladores fuzzy do tipo Proporcional-Derivativo e também controladores PI lineares. O gráfico tridimensional que caracteriza o controlador fuzzy pode ser gerado também através desse programa.

Os dados relativos à simulação são lidos de um arquivo fornecido pelo usuário. Os resultados são gravados em arquivos.

A principal característica desse simulador é a possibilidade de escolha de diversos algoritmos de raciocínio fuzzy combinando-se as funções de implicação fuzzy e o conectivo também. O conectivo é sempre feito através da intersecção.

O programa pode ser facilmente expandido adicionando-se novas plantas e novos algoritmos de raciocínio fuzzy.

O programa é composto por 4 módulos:

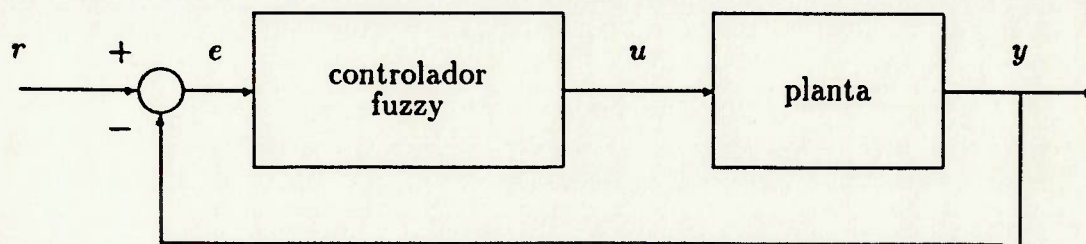


Figura A.1: sistema de controle (controlador fuzzy + planta)

1. Módulo Principal: é responsável pelo tratamento do arquivo de dados de simulação e pela inicialização das estruturas de dados.
2. Módulo de Simulação : gerencia a simulação, contém a definição dos procedimentos que simulam as plantas.
3. Máquina de Inferência Fuzzy : executa os algoritmos de raciocínio fuzzy,
4. Algoritmos de Integração Numérica : contém algoritmos de integração numérica.

os módulos e sua hierarquia estão representados na figura A.2

Nas seções a seguir são detalhados tais módulos.



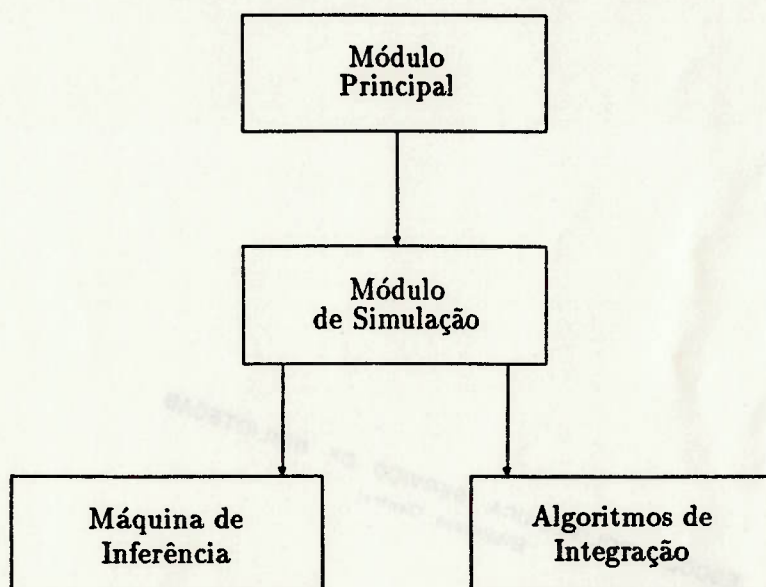


Figura A.2: módulos do Simulador Básico.



## A.2.2 Módulo Principal

Este módulo é implementado como o programa principal (main) e sua função é realizar o tratamento dos dados do arquivo de entrada e inicializar adequadamente as estruturas de dados.

### Estruturas de Dados

As principais estruturas de dados desse módulo e que são também utilizadas nos demais são detalhadas a seguir:

1. **struct rac\_fuzzy prf**: contém os dados relativos ao algoritmo de raciocínio fuzzy escolhido e o tipo de função de pertinência. O tipo **structure rac\_fuzzy** é definido como:

```
struct rac_fuzzy {
    double (*funcao_de_pertinencia[2])();
    int indica_tipo_funcao;
    void (*discretizacao[2])();
    int indica_discretizacao;
    void (*algoritmo_de_raciocinio[10])();
    int indica_alg_rac;
    void (*conectivo_tambem[10])();
    int indica_conec;
};
```

- (a) **double (\*funcao\_de\_pertinencia[2])()**: vetor de ponteiros para as funções que calculam o valor para uma dada função de pertinência. A inicialização desse vetor é mostrada a seguir,

```
prf.funcao_de_pertinencia[0] = Gaussiana;
prf.funcao_de_pertinencia[1] = Triangular;
```

onde **Gaussiana** é o nome da função que calcula o valor de uma função do tipo gaussiana em um dado ponto e o mesmo para **Triangular** em relação a função do tipo triangulares.

- (b) **int indica\_tipo\_funcao**: o valor 0 se refere a escolha de funções de pertinência do tipo gaussiana e 1 do tipo triangular.
- (c) **void (\*discretizacao[2])()**: vetor de ponteiros para funções que discretizam uma dada função de pertinência no espaço normalizado  $[-1, +1]$ . A inicialização desse vetor é mostrada a seguir,

```
prf.discretizacao[0] = Discretizacao_gauss;
prf.discretizacao[1] = Discretizacao_triang;
```

- (d) `int indica_discretizacao`: o valor 0 se refere a discretização de funções do tipo gaussiana e 1 do tipo triangular.
- (e) `void (*algoritmo_de_raciocinio[10])()`: vetor de ponteiros para as funções que implementam algoritmos de raciocínio fuzzy. A inicialização desse vetor é mostrada a seguir,

```
prf.algoritmo_de_raciocinio[0] = Reasoning_Mandani;
prf.algoritmo_de_raciocinio[1] = Reasoning_Ra;
prf.algoritmo_de_raciocinio[2] = Reasoning_Rbp;
prf.algoritmo_de_raciocinio[3] = Reasoning_Rdp;
prf.algoritmo_de_raciocinio[4] = Reasoning_Larsen;
```

- (f) `int indica_alg_rac`: indica qual o algoritmo de raciocínio fuzzy desejado.

0 = algoritmo de Mandani  
 1 = regra aritmética como função de implicação  
 2 = produto limitado como função de implicação  
 3 = produto drástico como função de implicação  
 4 = algoritmo de Larsen

- (g) `void (*conectivo_tambem[10])()`: vetor de ponteiros para funções que implementam algoritmos que definem o conectivo também. A inicialização desse vetor é mostrada a seguir:

```
prf.conectivo_tambem[0]= uniao;
prf.conectivo_tambem[1]= soma_algebrica;
prf.conectivo_tambem[2]= soma_limitada;
prf.conectivo_tambem[3]= soma_drastica;
prf.conectivo_tambem[4]= soma_disjunta;
prf.conectivo_tambem[5]= interseccao;
prf.conectivo_tambem[6]= produto_algebrico;
prf.conectivo_tambem[7]= produto_limitado;
prf.conectivo_tambem[8]= produto_drastico;
```

- (h) `int indica_conec`: indica qual o algoritmo para o conectivo também desejado.

0 = união  
 1 = soma algébrica  
 2 = soma limitada  
 3 = soma drástica  
 4 = soma disjunta  
 5 = intersecção  
 6 = produto algébrico  
 7 = produto limitado  
 8 = produto drástico

2. `struct data_rule_base pdata`: contém as informações que compõem a base de dados e a base de regras fuzzy. O tipo `struct data_rule_base` é definido como:

```

struct data_rule_base {
    int M_inferencia[15][15];
    double vetore[15][2];
    int ne;
    double vetorce[15][2];
    int nce;
    double vetorcu[30][2];
    int ncu;
    double KE, KCE, KCU;
};

```

- (a) `int M_inferencia[15][15]`: matriz de inferência fuzzy (base de regras) para duas variáveis de entrada e uma variável de saída.
  - (b) `double vetore[15][2]`: vetor contendo os parâmetros que definem as funções de pertinência da variável *e* (erro).
  - (c) `int ne`: número de funções de pertinência para a variável *e*.
  - (d) `double vetorce[15][2]`: idem para a variável *ce* (variação do erro).
  - (e) `int nce`: idem para a variável *ce*.
  - (f) `double vetorcu`: idem para a variável *cu* (variação do esforço de controle).
  - (g) `int ncu`: idem para a variável *cu*.
  - (h) `KE`: fator multiplicativo para a variável *e*.
  - (i) `KCE`: fator multiplicativo para a variável *ce*.
  - (j) `KCU`: fator multiplicativo para a variável *cu*.
3. `struct seg_real_normalizado pseg`: contêm as estruturas de dados relativas a normalização dos espaços. O tipo `struct seg_real_normalizado` é definido como:

```

struct seg_real_normalizado {
    double vetore_real[10];
    double vetore_normalizado[10];
    int nsege;
    double vetorce_real[10];
    double vetorce_normalizado[10];
    int nsegece;
    double vetorcu_real[10];
    double vetorcu_normalizado[10];
    int nsegecu;
};

```

- (a) `double vetore_real[10]`: contêm os valores que correspondem às extremidades dos segmentos no espaço real da variável *e*. [`vetore_real[0]`, `vetore_real[1]`] corresponde ao primeiro segmento, [`vetore_real[1]`, `vetore_real[2]`] corresponde ao segundo segmento, e assim por diante.
- (b) `double vetore_normalizado[10]`: contêm os valores que correspondem às extremidades dos segmentos no espaço normalizado da variável *e*.
- (c) `int nsege`: número de segmentos no espaço *e*.
- (d) `double vetorce_real[10]`: idem para a variável *ce*.



- (e) double `vetorce_normalizado[10]`: idem para a variável `ce`.
- (f) int `nsegce`: idem para a variável `ce`.
- (g) double `vetorcu_real[10]`: idem para a variável `cu`.
- (h) double `vetorcu_normalizado[10]`: idem para a variável `cu`.
- (i) int `nsegcu`: idem para a variável `cu`.

4. struct `parametros_planta pp`: contém os parâmetros que definem as plantas a serem simuladas. O tipo struct `parametros_planta` é definido como:

```
struct parametros_planta
{
    double a, b, C;
    double KD, TID; /* constantes do controlador PI */
    int delay;
    double x1, x2; /* valores iniciais p/ o pendulo invertido */
};
```

(a) double `a, b, C`: parâmetros que definem as plantas de primeira e segunda ordem.

- i.  $G(s) = \frac{C}{(s+a)}$
- ii.  $G(s) = \frac{C}{(s+a)(s+b)}$
- iii.  $G(s) = \frac{\exp(-ds)C}{(s+a)}$
- iv.  $G(s) = \frac{\exp(-ds)C}{(s+a)(s+b)}$

(b) double `KD`: ganho proporcional do controlador PI.

(c) double `TID`: constante de tempo integral do controlador PI.

(d) int `delay`: numero de intervalos de amostragem de atraso para sistemas com tempo morto.

(e) double `x1, x2`: valores iniciais para as variáveis  $\theta$  e  $\dot{\theta}$  do pêndulo invertido.

5. struct `para_simulacao l`: contém parâmetros de simulação setados pelo usuário. A definição do tipo struct `para_simulacao` é mostrado a seguir:

```
struct para_simulacao {
    boolean indfuzzy; /* indica se a simulacao com
                       controlador fuzzy deve ser
                       realizada ou nao */
    boolean ind3d; /* idem para grafico 3d */
    boolean indpi; /* idem para controle PI */
    int ind_planta; /* indica qual a planta a ser
                    simulada */
    int ind_fent; /* indica qual a funcao de
                  entrada */
    double timemax; /* tempo maximo de simulacao */
    double T; /* tempo de amostragem */
    double alfa; /* fator multiplicativo da funcao
                  de entrada */
};
```

- (a) **boolean** *indfuzzy*: indica se a simulação do controlador fuzzy deve ser realizada ou não.
  - (b) **boolean** *ind3d*: indica se os valores para o gráfico tridimensional que caracteriza o controlador fuzzy deve ser gerado.
  - (c) **boolean** *indpi*: indica se a simulação para o controlador PI deve ser realizada ou não.
  - (d) **int** *ind\_planta*: indica qual a planta que deve ser simulada.
    - 0 = primeira-ordem
    - 1 = segunda-ordem
    - 2 = primeira-ordem com tempo morto
    - 3 = segunda-ordem com tempo morto
    - 4 = pêndulo invertido
  - (e) **int** *ind\_fent*: indica qual a função escolhida para gerar os valores de referência.
    - 0 = Degrau
    - 1 = Rampa
  - (f) **double** *timemax*: tempo máximo de simulação.
  - (g) **double** *T*: tempo de amostragem em segundos.
  - (h) **double** *alfa*: fator multiplicativo da função de referência.
6. **nc**: número de pontos para a discretização das funções de pertinência.
7. **n3d**: número de pontos para a geração do gráfico tridimensional.

### Algoritmo

O algoritmo para o programa principal é apresentado a seguir:

```
<inicialização das estrutura prf>;
<imprime 'nome do arquivo de simulação ?'>;
<leia teclado>;
<abre arquivo de dados>;
<leia o nome base para os arquivos na variável arq>;
<leia 1.indfuzzy, 1.indpi, 1.ind3d>;
se 1.indfuzzy = TRUE ou 1.indpi = TRUE então
  inicio
  <leia as variáveis que fazem parte da estrutura 1>;
  <leia os parâmetros para a planta escolhida>;
  fim
se 1.indfuzzy = TRUE ou 1.ind3d = TRUE então
  inicio
  <leia nc, n3d>;
  <leia as variáveis que fazem parte das estruturas pdata, prf,pseg>;
```

```

fim
se l.indpi= TRUE então
  inicio
  <leia pp.KD, pp.TID>;
  fim
<fecha o arquivo>
< chamada do procedimento de controle da simulação >;

```

### A.2.3 Módulo de Simulação

O módulo de simulação gerencia a simulação e possui a definição dos procedimentos que simulam as plantas.

#### Estruturas de Dados

As principais estruturas de dados definidas nesse módulo são:

1. `double (*funcao_de_entrada[2])()`: vetor de ponteiros para funções que geram valores de referência. A inicialização desse vetor é mostrada a seguir:

```

funcao_de_entrada[0] = Degrau;
funcao_de_entrada[1] = Rampa;

```

2. `double (*planta[10])()`: vetor de ponteiros para funções que simulam as plantas. A inicialização desse vetor é mostrada a seguir:

```

planta[0] = Primeira_ordem;
planta[1] = Segunda_ordem;
planta[2] = Primeira_ordem_delay;
planta[3] = Segunda_ordem_delay;
planta[4] = pendulo_invertido;

```

#### Algoritmos

O procedimento `void Controle.de.Simulação()` realiza o gerenciamento da simulação. Um algoritmo simplificado para esse procedimento é mostrado a seguir:

```

< inicialização de funcao_de_entrada e planta >;
se l->indfuzzy = TRUE então
  < realize a simulação do controlador fuzzy >;
se l->indpi = TRUE então
  < realize a simulação do controlador PI >;
se l->ind3d = TRUE então
  < realize a geração do gráfico tridimensional >;

```



Para cada simulação e geração de gráfico 3d é gerado um arquivo contendo a informações desejadas. O nome de tais arquivos é gerado a partir da cadeia de caracteres armazenada na variável `arq` mais as extensões '.fz' para a simulação do controlador fuzzy, '.pi' para a simulação do controlador PI e '.3d' para o gráfico tridimensional.

O algoritmo para a simulação do controlador fuzzy + planta é mostrado a seguir:

$$\begin{aligned}e(k) &\leftarrow y(k) - \alpha s(k); \\ce(k) &\leftarrow e(k-1) - e(k-2); \\cu(k) &\leftarrow g_c(K_e e(k), K_{ce} ce(k)); \\u(k) &\leftarrow K_{cu} + u(k-1); \\y(k+1) &\leftarrow g_p(u(k)); \\k &= 0, 1, \dots\end{aligned}$$

onde  $g_c()$  representa o controlador fuzzy e  $g_p()$  representa a planta a ser controlada;  $e(k)$  é o erro do sistema,  $ce(k)$  é a variação do erro,  $cu(k)$  é a variação do esforço de controle e  $u(k)$  é o esforço de controle;  $\alpha, K_e, K_{ce}$  e  $K_{cu}$  são fatores multiplicativos. Note que  $cu(k)$  depende de  $e(k-2)$  o que provoca um atraso na resposta do controlador.

A seguir é apresentado a implementação completa em linguagem C do procedimento `void Controle_de_Simulacao()`:

```

void Controle_de_Simulacao(prf, pdata, pseg, pp, l, nc, n3d, arq)
struct rac_fuzzy *prf;
struct data_rule_base *pdata;
struct seg_real_normalizado *pseg;
struct parametros_planta *pp;
struct para_simulacao *l;
int nc; /* numero de pontos de discretizacao de funcoes de pertinencia */
int n3d; /* numero de pontos no espaco eo e ceo p/ construcao do grafico 3d */
char arq[]; /* nome base para arquivos */
{
double (*planta[10])(struct parametros_planta*, double,int,double);
/* vetor de ponteiros para as funcoes que simulam plantas */
double (*funcao_de_entrada[2])(int,double); /* 0 - degrau
1 - rampa */
double pendulo_invertido(struct parametros_planta*,double,int,double);
double t, e_k, e_k_menos_1, e_k_menos_2, cu_k, setp_k, y_k, y_k_mais_1;
double u_k, stepco, stepceo, eo, ceo, ceoi, uo, au;
int k, i, j, cont;
FILE *fp1, *fp2, *fp3;
char aux[40];

/* inicializacao do vetor de ponteiros para funcoes de entrada */
funcao_de_entrada[0] = Degrau;
funcao_de_entrada[1] = Rampa;

/* inicializacao do vetor de ponteiros para plantas */
planta[0] = Primeira_ordem;
planta[1] = Segunda_ordem;
planta[2] = Primeira_ordem_delay;
planta[3] = Segunda_ordem_delay;
planta[4] = pendulo_invertido;

if ( l->indfuzzy == TRUE ) /* simulacao de controle fuzzy ? */
{
t=0;
e_k = 0;
u_k = 0;
cu_k = 0;
setp_k = 0;
e_k_menos_1 = 0;
e_k_menos_2 = 0;
y_k = 0;
y_k_mais_1 = 0;
k=0;
strcpy(aux,arq); /* copia arq em aux */
strcat(aux, ".fzy"); /* concatena strings */
fp1 = fopen(aux, "w+");
fprintf(fp1, "t y e ce u cu\n");
while ( t < l->timemax )
{
setp_k = l->alfa * (*funcao_de_entrada[l->ind_fent])(k,l->T); /* calculo do setpoint */
e_k = y_k - setp_k;
cu_k = Maquina_de_inferencia(prf,pdata,pseg,pdata->IE*e_k_menos_1,
pdata->KCE*(e_k_menos_1-e_k_menos_2),nc);
u_k = pdata->KCU * cu_k + u_k;
y_k_mais_1 = (*planta[l->ind_planta])(pp,u_k,k,l->T);
/* rotinas de arquivos */
au = e_k - e_k_menos_1;
fprintf(fp1, "%f %f %f %f %f %f\n", t, y_k, e_k, au, u_k, cu_k);
printf("t=%f y=%f e=%f ce=%f u=%f cu=%f\n", t, y_k, e_k, au, u_k, cu_k);

k = k + 1;
t = t + l->T;
/* atualizacao */
e_k_menos_2 = e_k_menos_1;
e_k_menos_1 = e_k;
y_k = y_k_mais_1;
}
}

```

```

setp_k = l->alfa * (*funcao_de_entrada[l->ind_fent])(k,l->T);
e_k = y_k - setp_k;
cu_k = Maquina_de_inferencia(prf, pdata, pseg, pdata->KE*e_k_menos_1,
                             pdata->KCE*(e_k_menos_1-e_k_menos_2),nc);
u_k = pdata->KCU * cu_k + u_k;
y_k_mais_1 = (*planta[l->ind_planta])(pp,u_k,k,l->T);
/* rotinas de arquivos */
au = e_k - e_k_menos_1;
fprintf(fp1,"%f %f %f %f %f\n",t,y_k,e_k,au,u_k,cu_k);
fclose(fp1); /* fecha o arquivo */
}

if ( l->indpi == TRUE ) /* execucao do controle PI */
{
t=0;
e_k = 0;
setp_k = 0;
y_k = 0;
y_k_mais_1 = 0;
k=0;
strcpy(aux,arq); /* copia arq em aux */
strcat(aux,".pi"); /* concatena strings */
fp2 = fopen(aux,"w+");
/* arquivos */
fprintf(fp2,"t y e\n");

while ( t < l->timemax )
{
setp_k = l->alfa * (*funcao_de_entrada[l->ind_fent])(k,l->T);
e_k = setp_k - y_k;
u_k = Controlador_PI_analog(e_k,pp->KD,pp->TID,k, ->T);
y_k_mais_1 = (*planta[l->ind_planta])(pp,u_k,k,l->T);
/* rotinas de arquivos */
fprintf(fp2, "%f %f %f \n", t, y_k, e_k);
printf("t=%f y=%f e= %f\n", t, y_k, e_k);

k = k + 1;
t = t + l->T;
y_k = y_k_mais_1;
}

setp_k = l->alfa * (*funcao_de_entrada[l->ind_fent])(k,l->T);
e_k = setp_k - y_k;
u_k = Controlador_PI_analog(e_k, pp->KD, pp->TID, k, l->T);
y_k_mais_1 = planta[l->ind_planta](pp, u_k, k, l->T);

/* rotinas de arquivos */
fprintf(fp2,"%f %f %f \n", t, y_k, e_k);
fclose(fp2);
}

if ( l->ind3d == TRUE ) /* grafico 3d ? */
{
cont = 1;
strcpy(aux,arq);
strcat(aux,".3d");
fp3 = fopen(aux,"w+");

stepco=(pseg->vetore_real[pseg->nsege]- \
         pseg->vetore_real[0])/(n3d-1);
/* calculo do passo no eixo eo */
stepceo = ( pseg->vetorce_real[pseg->nsegce] - \
            pseg->vetorce_real[0] ) / (n3d-1);
/* calculo do passo no eixo ceo */
eo = pseg->vetore_real[0];
ceoi = pseg->vetorce_real[0];
for ( i=1; i<=n3d; i++ )
{
ceo = ceoi;

```



```

for ( j=1; j<=n3d; j++ )
{
  uo = Maquina_de_inferencia(prf, pdata, pseg, eo, ceo, nc);
  printf("eo=%f ceo =%f\n", eo, ceo);
  printf("cont=%d\n",cont);
  cont++;
  printf("%f\n",uo);
  fprintf(fp3,"%f\n",uo);
  ceo = ceo + stepceo;
} /* end for ceo */
eo = eo + stepceo;
} /* end for eo */
fclose(fp3); /* fecha arquivo */
} /* end if ind3d */
} /* end Controle_de_simulacao */

```

Nessa versão foram implementadas funções que simulam as seguintes plantas:

- 0 = primeira-ordem,
- 1 = segunda-ordem,
- 2 = primeira-ordem com tempo morto,
- 3 = segunda-ordem com tempo morto,
- 4 = pêndulo invertido.

Os algoritmos das plantas lineares são realizados através de equações de diferenças. Como exemplo, a seguir é mostrado o algoritmo para a planta linear de primeira-ordem.

```

double Primeira_ordem(struct parametros_planta *p,double u_k,int k,double T)
{
  static double y_k;
  double y_k_mais_1;
  if ( k == 0 )
  {
    y_k = 0.0;
    y_k_mais_1 = 0.0;
  }
  y_k_mais_1 = exp(-1.0*p->a*T)*y_k + (p->C/p->a)*(1-exp(-1.0*p->a*T))*u_k;
  /* atualizacao */
  y_k = y_k_mais_1;
  return(y_k_mais_1);
}

```

Os algoritmos de plantas não lineares são simulados através de algoritmos de integração do módulo *Algoritmos de Integração Numérica* levando em consideração o tempo de amostragem  $T$ , durante o qual a entrada permanece constante para a planta. Durante o intervalo de amostragem  $T$  as equações diferenciais são integradas com um passo de integração  $h = T/10$ .

A utilização do vetor de ponteiros double (*\*planta[10]*)( ) permite que novas plantas sejam facilmente adicionadas ao sistema.

#### A.2.4 Máquina de Inferência Fuzzy

Este módulo é responsável pelos algoritmos de raciocínio fuzzy para base de regras com duas variáveis de entrada e uma variável de saída. A função double *Maquina\_de\_inferencia* é a função principal desse módulo e o seu código é mostrado a seguir.

```

double Maquina_de_inferencia(struct rac_fuzzy *prf, struct data_rule_base *pdata,
                             struct seg_real_normalizado *pseg, double eo, double ceo, int nc)
{
double Real_para_normalizado(double valor_real, double vetor_real[],
                              double vetor_normalizado[], int nseg);
double Normalizado_para_real(double valor_normalizado, double vetor_real[],
                              double vetor_normalizado[], int nseg);
double eno, ceno, cuno, cuo;

/* calcula o valor normalizado de eo */
eno = Real_para_normalizado(eo, pseg->vetore_real, pseg->vetore_normalizado, pseg->nsego);
/* calcula o valor normalizado de ceo */
ceno = Real_para_normalizado(ceo, pseg->vetorce_real, pseg->vetorce_normalizado, pseg->nsegce);
/* realiza a inferencia fuzzy com valores normalizados */
cuno = Infere(prf, pdata, eno, ceno, nc);
/* transforma o valor normalizado da variavel de saida cuno para o valor real */
cuo = Normalizado_para_real(cuno, pseg->vetorcu_real, pseg->vetorcu_normalizado, pseg->nsegcu);
return(cuo);
}

```

Todas as operações que envolvem os algoritmos são realizadas com valores normalizados no intervalo  $[-1, +1]$ . O primeiro passo portanto é descobrir quais são os valores normalizados para a variável *eo* e *ceo*. A operação de normalização é realizada pela função `double Real_para_normalizado` cujo código de encontra a seguir.

```

double Real_para_normalizado(double valor_real, double vetor_real[],
                              double vetor_normalizado[], int nseg)
{
int Acha_posicao(double valor, double vetor[]);
double valor_normalizado;
int posicao;

if ( (valor_real >= vetor_real[0]) && (valor_real <= vetor_real[nseg]) )
{
posicao = Acha_posicao( valor_real, vetor_real );
valor_normalizado = ( (valor_real - vetor_real[posicao])/
                      (vetor_real[posicao+1] - vetor_real[posicao]) ) *
                    ( vetor_normalizado[posicao+1] -
                      vetor_normalizado[posicao] ) +
                      vetor_normalizado[posicao];
}
else
{
printf("fora dos limites reais\n");
}
return(valor_normalizado);
}

```

Os vetores `double vetor_real[]` e `double vetor_normalizado` contêm os valores extremos para cada segmento. O valor real `vetor_real[0]` corresponde ao valor normalizado `vetor_normalizado[0]`; o valor real `vetor_real[1]` corresponde ao valor normalizado `vetor_normalizado[1]` e assim por diante.

A função `int Acha_posicao(double valor, double vetor[])` devolve um inteiro *i* tal que  $\text{vetor}[i] \leq \text{valor} \leq \text{vetor}[i+1]$ .

A função `double Normalizado_para_real()` realiza o oposto.

A função `double Infere()` é que efetivamente realiza a inferência fuzzy através de valores normalizados. A implementação na linguagem C dessa função é mostrada a seguir.

```

double Infere(struct rac_fuzzy *prf, struct data_rule_base *pdata, double uo, double vo, int nc)
{
double miCI[50], miC[50], miCR[50], miA_uo, miB_vo, cuo;

```



```

int i, j, k, w;

for(i=0; i < nc; i++) /* inicializa a matriz com zeros */
  miCT[i]=0.0;
for(i=0; i < pdata->ne; i++) /* varre a matriz de inferencia */
{
  for(j=0 ;j < pdata->ncc; j++)
  {
    k = pdata->H_inferencia[i][j]; /* consulta a matriz de inferencia
                                   k e' o indice do conjunto fuzzy para
                                   a variavel de saida */
    /* grau de pertinencia para o valor uo no conjunto i da variavel u */
    miA_uo = (prf->funcao_de_pertinencia[prf->indica_tipo_funcao])(uo, pdata->vetore[i][0],
                                                                pdata->vetore[i][1], i, pdata->ne);
    /* grau de pertinencia para o valor vo no conjunto j da variavel v */
    miB_vo = (prf->funcao_de_pertinencia[prf->indica_tipo_funcao])(vo, pdata->vetorce[j][0],
                                                                pdata->vetorce[j][1], j, pdata->ncc);
    /* discretizacao do conjunto fuzzy k da variavel de saida cu */
    (prf->discretizacao[prf->indica_discretizacao])(pdata->vetorcu[k][0], pdata->vetorcu[k][1],
                                                  nc, miC, k, pdata->ncu);
    /* executa o algoritmo de raciocinio fuzzy */
    (prf->algoritmo_de_raciocinio[prf->indica_alg_rac])(miA_uo, miB_vo, miC, nc, miCR);

    if ( i == 0 && j == 0) /* primeira execucao */
    {
      for (w=0; w<nc; w++)
      {
        miCT[w]=miCR[w];
      }
    }
    else /* operacao correspondente ao conectivo tambem */
    {
      (prf->conectivo_tambem)[prf->indica_conec](miCT, miCR, miCT, nc);
    }
  } /* end for j */
} /* end for i */

alfa_cut(0.0, miCT, nc); /* operacao alfa cut com nivel 0 */
cuo = Centro_de_gravidade(miCT, nc); /* calcula o centro de gravidade */
return(cuo);
}

```

Deve ser observado que a forma como foi implementado o algoritmo acima, utilizando vetor de ponteiros para funções ou procedimentos permite que, para cada operação fundamental, diversos algoritmos diferentes possam ser selecionados.

- *tipos de função de pertinência:* podem ser escolhidas funções do tipo triangular ou do tipo gaussiana. O procedimento para discretização também deve ser de mesmo tipo, ou seja, se forem escolhidas funções triangulares deve ser utilizado o procedimento de discretização de funções triangulares. O protótipo de tais funções é mostrado a seguir.

```

double Gaussiana(double x, double u_f, double sigma_f, int i, int nmax);
double Triangular(double x, double xmax, double comp, int i, int nmax);
void Discretizacao_triangular(double xmax, double comp, int n, double f[], int i, int nmax);
void Discretizacao_gauss(double u_f, double sigma_f, int n, double f[], int i, int nmax);

```

Os parâmetros  $u_f$  e  $\sigma_f$  em `double Gaussiana` é que definem a função gaussiana  $\mu_f(x) = \exp(-(x - u_f)^2 / 2\sigma_f^2)$ . ,  $i$  indica a posição relativa dessa função em relação às demais no espaço normalizado.



As funções do tipo triangular são definidas através do ponto de máximo `double xmax` e do comprimento da base `double comp`. Os triângulos definidos são sempre isósceles.

- *algoritmos de raciocínio*: os algoritmos de raciocínio disponíveis utilizam as seguintes funções de implicação:  $R_c$  (Mandani),  $R_p$  (Larsen),  $R_a$  (Regra Aritmética),  $R_{dp}$  (Produto Drástico),  $R_{bp}$  (Produto Limitado). A interpretação do conectivo e é realizado através do operador intersecção ( $\wedge$ ).

```
void Reasoning_Mandani(double miA_uo,double miB_vo,double miC[],int nc,double miCR[]);
void Reasoning_Ra(double miA_uo,double miB_vo,double miC[],int nc,double miCR[]);
void Reasoning_Rbp(double miA_uo,double miB_vo,double miC[],int nc,double miCR[]);
void Reasoning_Rdp(double miA_uo,ouble miB_vo,double miC[],int nc,double miCR[]);
void Reasoning_Larsen(double miA_uo,double miB_vo,double miC[],int nc,double miCR[]);
```

Como exemplo o código da função para o método de Mandani é mostrado a seguir:

```
void Reasoning_Mandani(double miA_uo,double miB_vo,double miC[],int nc,double miCR[])
{
double min;
int i;
min=min2(miA_uo, miB_vo);
for (i = 0; i < nc; i++)
{
miCR[i] = min2(min, miC[i]);
}
}
```

- *conectivos também*: as operações que podem ser utilizadas são as seguintes normas ou co-normas triangulares: intersecção, produto algébrico, produto limitado, produto drástico, união, soma algébrica, soma limitada, soma drástica e soma disjunta. Os protótipos dessas funções são mostradas a seguir.

```
void uniao(double miA[],double miB[],double miC[],int np);
void soma_algebrica(double miA[],double miB[],double miC[],int np);
void soma_limitada(double miA[],double miB[],double miC[],int np);
void soma_drastica(double miA[],double miB[],double miC[],int np);
void soma_disjunta(double miA[],double miB[],double miC[],int np);
void interseccao(double miA[],double miB[],double miC[],int np);
void produto_algebrico(double miA[],double miB[],double miC[],int np);
void produto_limitado(double miA[],double miB[],double miC[],int np);
void produto_drastico(double miA[],double miB[],double miC[],int np);
```

Como exemplo é mostrado a seguir o código para a função que realiza a operação de união.

```
void uniao(double miA[], double miB[], double miC[], int np)
{
int i;
for(i=0; i<np; i++)
miC[i] = max2(miA[i], miB[i]);
}
```

### A.2.5 Algoritmos de Integração Numérica

Neste módulo podem ser utilizados os seguintes algoritmos de integração numérica:

1. Euler:
2. Runge-Kutta de 4ª ordem:
3. Múltiplos passos:

### A.3 Simulador para o Método Simplificado

Esse programa realiza simulações de controladores fuzzy utilizando o método de raciocínio fuzzy simplificado cujo algoritmo pode ser encontrado no capítulo 2

### A.4 Simulador para o Método de Buckley e Ying

Esse programa realiza simulações de controladores fuzzy utilizando o método de Buckley e Ying cujo algoritmo pode ser encontrado no capítulo 4. As regras utilizadas devem ser regras lineares segundo o conceito de Buckley e Ying.

### A.5 Simulador para o Pêndulo Invertido

Esse programa permite simular o controle de um pêndulo invertido. Essa versão foi alterada de tal forma que permitisse:

1. 4 variáveis de entrada ( $\theta$ ,  $\dot{\theta}$ ,  $x$ ,  $\dot{x}$ ) e uma variável de saída ( $f$ ),
2. comportamento Proporcional-Derivativo para o controlador fuzzy.

# Bibliografia

- [1] ALIEV, R.A.; MAMEDOVA, G.A. Analysis of fuzzy models industrial processes. **Fuzzy Sets and Systems**, Amsterdam, v.37,n.1, p.13-21, Aug. 1990.
- [2] ANDERSEN, T.R.; NIELSEN, S. B. An efficient single output fuzzy control algorithm for adaptive applications, **Automatica**, Oxford, v.21, n.5, p.539-45, Sept. 1985.
- [3] ARACYL, J.; OLLERO, A.; CEREZO, A.G. Stability indices for the global analysis of expert control systems. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.19, n.5, p.998-1007, Sept./Oct. 1989.
- [4] ÅSTRÖM, K.J.; WITTENMARK, B. **Computer-controlled systems: theory and design**. Englewood Cliffs, N. J., Prentice-Hall, 1984.
- [5] ÅSTRÖM, K.J.; ANTON, J.J.; ARZEN K.E. Expert control. **Automatica**, Oxford, v.22, n.3. p.277-86, March 1986.
- [6] ÅSTRÖM, K.J. Toward intelligent control. **IEEE Control Systems Magazine**, New York, v.9, n.3, p.60-4, Apr. 1989.
- [7] BABOSHIN, N.P.; NARYSHKIN, D.G. On identification of multidimensional fuzzy systems. **Fuzzy Sets and Systems**, Amsterdam, v.35, n.3, p.325-31, May 1990.
- [8] BALDWIN, J.F. A new approach to approximate reasoning using a fuzzy logic. **Fuzzy Sets and Systems**, Amsterdam, v.2, n.4, p.309-25, Oct. 1979.
- [9] BALDWIN, J.F.; PILSWORTH, B.W. Axiomatic approach to implication for approximate reasoning with fuzzy logic. **Fuzzy Sets and Systems**, Amsterdam, v.3, n.2, p.193-219, March 1980.
- [10] BALDWIN, J.F. Fuzzy sets and expert systems. **Information Sciences**, New York, v.36, n.1-2, p.123-56, 1985.
- [11] BARE W.; MULHOLLAND, R.J.; SOFER S.S. Design of a self-tuning rule based controller for a gasoline refinery catalytic reformer. **IEEE Transactions on Automatic Control**, New York, v.35, n.2, p.156-64, Feb. 1990.
- [12] BARTO, A.G.; SUTTON, R.S.; ANDERSON, C.W. Neuronlike adaptive elements that can solve difficult learning control problems. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.13, n.5, p.834-46, Sept./Oct. 1983.



- [13] BELLMAN, R.E.; ZADEH, L.A. Decision-making in a fuzzy environment. **Management Science**, Providence, v.17, n.4, p.141-63, Dec. 1970.
- [14] BENNET, M.E. Real-time continuous AI systems. **IEE Proceedings**, London, Part D, v.134, n.4, p.272-7, July 1987.
- [15] BERNARD, J.A. Use of a rule-based system for process control. **IEEE Control Systems Magazine**, New York, v.8, n.5, p.3-13, Oct. 1988.
- [16] BOYD, S.; BARRAT, C.; NORMAN S. Linear controller design: limits of performance via convex optimization, **Proceedings of the IEEE**, New York, v.78, n.3, March 1990.
- [17] BRAAE, M.; RUTHERFORD D.A. Theoretical and linguistics aspects of the fuzzy controller. **Automatica**, Oxford, v.15, n.5, p.553-77, Sept. 1979.
- [18] BRAAE M.; RUTHERFORD, D.A. Selection of parameters for a fuzzy logic controller. **Fuzzy Sets and Systems**, Amsterdam, v.2, n.3, p.185-99, July 1979.
- [19] BUCKLEY, J.J. Fuzzy controller theory: limit theorems for linear fuzzy control rules. **Automatica**, Oxford, v.25, n.3, p.469-72, May 1989.
- [20] BUCKLEY, J.J. Linear fuzzy controller: It is a non-linear nonfuzzy controller. **Information Sciences**, New York, v.51, n.2, p.183-92, 1990.
- [21] CHANG, S.S.L.; ZADEH, L.A. On fuzzy mapping and control, **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.2, n.1, p.30-35, Jan. 1972.
- [22] CHEN, Y.Y. Stability analysis of fuzzy control - a Lyapunov approach. In: **IEEE SYSTEMS, MAN, AND CYBERNETICS ANNUAL CONFERENCE**, Alexandria, 1987. **Proceedings**. New York, 1987. v.3, p.1027-31.
- [23] CHEN, Y.Y. **The global analysis of fuzzy dynamical systems**. 125p. Berkeley, 1989. Thesis (PhD) - Department of Electrical Engineering and Computer Sciences, University of California.
- [24] CHEN, Y.Y.; TSAO, T.C. A description of the dynamical behavior of fuzzy systems. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.19, n.4, p.745-55, July/Aug. 1989.
- [25] CHIU, S.; CHAND, S.; MOORE, D.; CHAUDHARY, A. Fuzzy logic for control of roll and moment for a flexible wing aircraft. **IEEE Control Systems Magazine**, New York, v.11, n.4, p.42-48, June 1991.
- [26] CZOGALA, E.; PEDRYCZ W. On identification in fuzzy systems and its applications in control problems. **Fuzzy Sets and Systems**, Amsterdam, v.6, n.1, p.73-83, July 1981.
- [27] CZOGALA, E.; DREWNIAK, J.; PEDRYCZ, W. Fuzzy relation equations on a finite set. **Fuzzy Sets and Systems**, Amsterdam, v.7, n.1, p.89-101, Jan. 1982.

- [28] CZOGALA, E.; PEDRYCZ, W. Control problems in fuzzy systems. **Fuzzy Sets and Systems**, Amsterdam, v.7, n.3, p.257-73, May 1982.
- [29] CZOGALA, E.; RAWLIK, T. Modelling of a fuzzy controller with application to the control of biological processes. **Fuzzy Sets and Systems**, Amsterdam, v.31, n.1, p.13-22, May 1989.
- [30] DUBOIS, D.; PRADE, H. **Fuzzy sets and systems**. New York, Academic Press, 1981.
- [31] DUBOIS, D.; PRADE, H. Unfair coins and necessity measure: toward a possibilistic interpretations of histograms. **Fuzzy Sets and Systems**, Amsterdam, v.10, n.1, p.15-20, April 1983.
- [32] DALEY, S. Comparison of a fuzzy logic controller with a P+D control law. **Transactions of the ASME: Journal of Dynamics Systems, Measurement and Control**, serie G, New York, v.111, n.2, p.128-37, June 1989.
- [33] FUKAMI, S.; MIZUMOTO, M.; TANAKA, K. Some considerations on fuzzy conditional inference. **Fuzzy Sets and Systems**, Amsterdam, v.4, n.3, p.243-73, Nov. 1980.
- [34] GAINES, B.R.; SHAW, M.L.G. From fuzzy logic to expert systems. **Information Sciences**, New York, v.36, n.1-2, p.5-16, July/Aug. 1985.
- [35] GORZALCZANY, M.B. A method of inference in approximate reasoning based on interval-valued fuzzy sets. **Fuzzy Sets and Systems**, Amsterdam, v.21, n.1, p.1-17, Jan. 1987.
- [36] GORZALCZANY, M.B. Interval-valued fuzzy controller based on verbal model of object. **Fuzzy Sets and Systems**, Amsterdam, v.28, n.1, p.45-53, Oct. 1988.
- [37] GRAHAM, B.P.; NEWELL, R.B. Fuzzy identification and control of a liquid level rig. **Fuzzy Sets and Systems**, Amsterdam, v.26, n.3, p.255-73, June 1988.
- [38] GRAHAM, B.P.; NEWELL, R.B. Fuzzy adaptive control of a first-order process. **Fuzzy Sets and Systems**, Amsterdam, v.31, n.1, p.47-65, May 1989.
- [39] GUPTA, M. M.; NIKIFORUK P.N. Modelling of multivariable fuzzy systems. In: **IFAC TRIENNIAL WORLD CONGRESS, 10.**, Munich, 1987. **Automatic Control**. Oxford, Pergamon Press, 1988. v.6, p.349-54.
- [40] HIROTA, K.; ARAI, Y.; HACHISU, S. Fuzzy controlled robot arm playing two-dimensional ping-pong game. **Fuzzy Sets and Systems**, Amsterdam, v.32, n.2, p.149-59, Sept. 1989.
- [41] HSU, C. S. A theory of cell-to-cell mapping dynamical systems. **ASME J. of Applied Mechanics**, serie Z, New York, v.47, n., p.931-9, 1980.



- [42] HSU, C.S.; GUTTALU, R.S. An unravelling algorithm for global analysis of dynamical systems: an application of cell-to-cell mappings. **ASME J. of Applied Mechanics**, serie Z New York, v.47, n., p.940-8, 1980.
- [43] HSU, C.S. A generalized theory of cell-to-cell mapping for nonlinear dynamical systems. **ASME J. of Applied Mechanics**, serie Z, New York, v.48, n.3, p.634-42, Sept. 1981.
- [44] HUANG, L.J. A self-paced fuzzy tracking controller for two-dimensional motion control. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.20, n.5, p.1115-24, Sept./Oct. 1990.
- [45] KAUFMANN, A.; GUPTA, M.M. **Introduction to fuzzy arithmetic**. New York, Van Nostrand, 1985.
- [46] KANIA, A.A. et alii. On stability of formal fuzziness systems, **Information Sciences**, New York v.22, n.1, p.51-68, Oct. 1980.
- [47] KICKERT, W.J.M.; LEMKE, H.R. Van Nauta. Application of a fuzzy controller. **Automatica**, Oxford, v.12, n.4, p.301-8, July 1976.
- [48] KICKERT, W.J.M.; MANDANI E.H. Analysis of a fuzzy logic controller. **Fuzzy Sets and Systems**, Amsterdam, v.1, n.1, p.29-44, 1978.
- [49] KING, P.J.; MANDANI, E.H. The application of fuzzy control systems to industrial processes. **Automatica**, Oxford, v.13, n.3, p.235-42, May 1977.
- [50] KISZKA, J.B.; KOCHANSKA, M.E.; SLIWINSKA, D.S. The influence of some fuzzy implication operators on the accuracy of a fuzzy model - part I. **Fuzzy Sets and Systems**, Amsterdam, v.15, n.2, p.111-28, March 1985.
- [51] KISZKA, J.B.; KOCHANSKA, M.E.; SLIWINSKA, D.S. The influence of some fuzzy implication operators on the accuracy of a fuzzy model - part II. **Fuzzy Sets and Systems**, Amsterdam, v.15, n.3, p.223-40, April 1985.
- [52] KISZKA, J.B.; GUPTA, M.M.; NIKIFORUK, P.N. Energistic Stability of fuzzy dynamic systems. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.15, n.6, p.783-92, Nov./Dec. 1985.
- [53] KISZKA, J.B.; GUPTA, M.M.; TROJAN, G.M. Multivariable fuzzy controller under Gödel's implication. **Fuzzy Sets and Systems**, Amsterdam, v.34, n.3, p.301-321, Feb. 1990
- [54] KOSKO, B. **Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence**, Prentice-Hall, Englewoods Cliffs, New Jersey, 1990.
- [55] LEE, C.C.; BERENJI H.R. An intelligent controller based on approximate reasoning and reinforcement learning. In: **IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL**, 4., Albany, 1989. **Proceedings**. New York. p.200-5.



- [56] LEE, C.C. Fuzzy logic in control systems: fuzzy logic controller - part I. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.20, n.2, March/April 1990.
- [57] LEE, C.C. Fuzzy logic in control systems: fuzzy logic controller - part II. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.20, n.2, March/April 1990.
- [58] LEUNG, K.S.; LAM, W. Fuzzy concepts in expert systems. **IEEE Computer Magazine**, Los Alamitos, v.21, n.9, p.43-56, Sept. 1988.
- [59] LI, Y.F.; LAU, C.C. Development of fuzzy algorithms for servo systems. **IEEE Control Systems Magazine**, New York, v.9, n.3, p.65-72, Apr. 1989.
- [60] MAEDA, M.; MURAKAMI, S. A design for a fuzzy logic controller. **Information Sciences**, New York, v.45, n.2, p.315-30, July, 1988.
- [61] MAEDA, M.; MAEDA Y.; MURAKAMI S. Fuzzy drive control of a autonomous mobile robot. **Fuzzy Sets and Systems**, Amsterdam, v.39, n.2, p.195-204, Jan. 1991.
- [62] MANDANI, E.H. Application of fuzzy algorithm for control of simple dynamic plant. **IEE Proceedings**, London, Part D, v.121, n.12, p., Dec. 1974.
- [63] MANDIC, N.J.; SCHARF, E.M.; MANDANI, E.H. Practical application of a heuristic fuzzy rule-based controller to the dynamic control of a robot arm. **IEE Proceedings**, London, Part D, v.132, n.4, p.190-203, July 1985.
- [64] MAIERS, J.; SHERIF, Y.S. Applications of fuzzy set theory. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.15, n.1, p.175-189, Jan./Feb. 1985.
- [65] MACVICAR-WHELAN, P.J. Fuzzy sets for man-machine interaction. **Int. J. Man-Machine Studies**, London, v.8, n., p.687-97, Nov. 1976.
- [66] MARTIN, T.P.; BALDWIN J.F.; PILSWORTH B.W. The implementation of fprolog - a fuzzy prolog interpreter. **Fuzzy Sets and Systems**, Amsterdam, v.23, n.1, p.119-29, July 1987.
- [67] MIZUMOTO, M.; ZIMMERMANN, Hans-Jürgen. Comparison of fuzzy reasoning methods. **Fuzzy Sets and Systems**, Amsterdam, v.8, n.3, p.253-83, Sept. 1982.
- [68] MIZUMOTO, M. Fuzzy controls under various reasoning methods. **Information Sciences**, New York, v.45, n.2, p.129-151, July 1988.
- [69] MURAYAMA, Y.; TERANO T. Optimizing control of a diesel engine. In: SUGENO, M. ed. **Industrial applications of fuzzy control**, Amsterdam, North-Holland, 1985, p.63-72.

- [70] OLLERO, A.; JIMÉNEZ, J.G.; CERESO, A.G.; SILVA, C.F. Expert tuning of automatic controllers. In: IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL, 4., Albany, 1989. **Proceedings**. New York. p.207-11.
- [71] OLLERO, A.; GARCÍA-CERESO, A.J. Direct digital control, auto-tuning and supervision using fuzzy logic. **Fuzzy Sets and Systems**, Amsterdam, v.30, n.2, p.135-53, April 1989.
- [72] OSTERGAARD, J.J. **Fuzzy logic control of a heat exchanger process**. Publication 7601, Electric Power Engineering Department, Technical University of Denmark, Jan. 1976.
- [73] PALM, R. Fuzzy controller for a sensor guided robot manipulator. **Fuzzy Sets and Systems**, Amsterdam, v.31, n.2, p.133-49, June 1989.
- [74] PEDRYCZ, W. Numerical and applicational aspects of fuzzy relational equations. **Fuzzy Sets and Systems**, Amsterdam, v.11, n.1, p.1-18, Aug. 1983.
- [75] PEDRYCZ, W. An identification algorithm in fuzzy relational systems. **Fuzzy Sets and Systems**, Amsterdam, v.13, n.2, p.153-67, July 1984.
- [76] PEDRYCZ, W.; CZOGALA, E.; HIROTA K.. Some Remarks on fuzzy identification problem. **Fuzzy Sets and Systems**, Amsterdam, v.12, n.2, p.185-9, Feb. 1984.
- [77] PENG, X.T. Generating rules for fuzzy logic controllers by functions. **Fuzzy Sets and Systems**, Amsterdam, v.36, n.1, p.83-9, May 1990.
- [78] PROCYK, T.J.; MANDANI, E.H. A linguistic self-organizing process controller. **Automatica**, Oxford, v.15, n.1, p.15-30, Jan. 1979.
- [79] RAY, K.S.; MAJUNDER, D.D. Application of circle criteria stability analysis of linear SISO and MIMO systems associated with fuzzy logic controller. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.14, n.2, p.345-9, Mar./Apr. 1984.
- [80] RAY, K.S.; GHOSH, A.M.; MAJUNDER, D.D.  $L_2$ -Stability and related design concept for SISO linear system associated with fuzzy logic controller. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.14, n.6, p.932-39, Nov./Dec. 1984.
- [81] RAY, K.S.; MAJUNDER, D.D. Fuzzy logic control of a nonlinear multivariable steam generating unit using decoupling theory. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.15, n.4, p.539-63, July/Aug. 1985.
- [82] RHEE, F.V.D.; LEMKE; H.R.V.N.; DIJKIMAN, J. G. Applying fuzzy set theory to modelling processes. In: IFAC TRIENNIAL WORLD CONGRESS, 10., Munich, 1987. **Automatic Control**. Oxford, Pergamon Press, 1988. p. 343-8.
- [83] SANCHEZ, E. Resolution of composite fuzzy relation equations. **Information and Control**, New York, v.30, n.1, p.38-48, Jan. 1976.



- [84] SHAO, S. Fuzzy self-organizing controller and its application for dynamic processes. **Fuzzy Sets and Systems**, Amsterdam, v.26, n.2, p.151-64, May 1988.
- [85] SILER, W.; YING, H. Fuzzy control theory: the Linear case, **Fuzzy Sets and Systems**, Amsterdam, v.33, n.3, p.275-90, Dec. 1989.
- [86] SILVA, L.E.B.; TORRES, G.L.. Usando regras nebulosas (Fuzzy Rules) em sistemas de controle em tempo real. In: ENCONTRO REGIONAL DE AUTOMAÇÃO E INSTRUMENTAÇÃO, 1., Vitória, 1989. **Anais. Vitória, UFES, 1989. v.1., p.IA01-IA07.**
- [87] SMITH, S.M.; COMER D.J. Automated calibration of a fuzzy logic controller using a cell state space algorithm, **IEEE Control Systems Magazine**, New York, v.11, n.5, p.18-28, Aug.1991.
- [88] SUGENO, M.; MURAKAMI, K. Fuzzy parking control of model car. In: IEEE CONF. ON DECISION AND CONTROL, 23., Las Vegas, 1984. **Proceedings. New York, 1984.**
- [89] SUGENO, M. An introductory survey of fuzzy control. **Information Sciences**, New York, v.36, n.1-2, p.59-83, July/Aug. 1985.
- [90] SUGENO, M.; KANG G.T. Structure identification of fuzzy model, **Fuzzy Sets and Systems**, Amsterdam, v.28, n.1, p.15-33, Oct. 1988.
- [91] SUGENO, M.; MUROFUSHI, T.; MORI, T.; TATEMATSU, T.; TANAKA, J. **Fuzzy Sets and Systems**, Amsterdam, v.32, n.2, p.207-19, Sept. 1989.
- [92] TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modelling and control. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.15, n.1, p.116-33, Jan./Feb. 1985.
- [93] TAKEUCHI, T.; NAGAI, Y.; ENOMOTO, N. Fuzzy control of a mobile robot for obstacle avoidance. **Information Sciences**, New York, v.45, n.2, p.231-48, July 1988.
- [94] TANAKA, H.; ASAI, K. Fuzzy solution in fuzzy linear programming problems. **IEEE Transactions on Systems, Man, and Cybernetics**, v.14, n.2, p.325-8, Mar./Apr. 1984.
- [95] TANG, K.L.; MULHOLLAND R.J. Comparing fuzzy logic with classical controller designs, **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.17, n.6, p.1085-7, Nov./Dec. 1987.
- [96] TANSCHAIT, R.; SCHARF, E.M. Experiments with the use of a rule-based self-organizing controller for robotics applications. **Fuzzy Sets and Systems**, Amsterdam, v.26, n.2, p.195-214, May 1988.
- [97] THOMASON, M.G. Convergence of powers of a fuzzy matrix, **Journal of Mathematical Analysis and Applications**, v.57, n.2, p.476-80, Feb. 1977.



- [98] TONG, R.M. A control engineering review of fuzzy systems. **Automatica**, Oxford, v.13, n.6, p.559-69, Nov. 1977.
- [99] TONG, R.M.; BECK, M.B.; LATTEN A. Fuzzy controller of the activated sludge wastewater treatment process. **Automatica**, Oxford, v.16, n.6, p.659-701, Nov. 1980.
- [100] TONG, R.M. Some properties of fuzzy feedback systems. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.10, n.6, p.327-331, June 1980.
- [101] TONG, R.M. The evaluation of fuzzy models derived from experimental data. **Fuzzy Sets and Systems**, Amsterdam, v.4, n.1, p.1-12, July 1980.
- [102] UMBERS, I.G.; KING, P.J. An analysis of human decision making in cement kiln control and implication for automation. **Int. J. Man-Machine Studies**, v.12, n.1, p.11-23, 1980.
- [103] VIDYASAGAR, M. **Nonlinear systems analysis**, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1978.
- [104] WANG, P.Z.; ZHANG, H.M.; XU, W. Pad-analysis of fuzzy control stability. **Fuzzy Sets and Systems**, Amsterdam, v.38, n.1, p.27-42, Oct. 1990.
- [105] WILLAEYS, D.; MALVACHE, N. The use of fuzzy sets for the treatment of fuzzy information by computer. **Fuzzy Sets and Systems**, Amsterdam, v.5, n.3, p.323-7, May 1981.
- [106] XU, C.W.; LU, Y.Z. Fuzzy model identification and self-learning for dynamic systems, **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.17, n.4, p.683-9, July/Aug. 1987.
- [107] XU, C.W. Decoupling fuzzy relational systems - an output feedback approach. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.19, n.2, p.414-8, Mar./Apr. 1989.
- [108] XU, C.W. **IEE proceedings**, London, Part D, v.136, n.4, p.146-50, July 1989.
- [109] XU, C.W.; LU, Y.Z. Decoupling in fuzzy systems: a cascade compensation approach. **Fuzzy Sets and Systems**, Amsterdam, v.29, n.2, p.177-85, Jan. 1989.
- [110] XU, C.W. Analysis and feedback/feedforward control of fuzzy relational systems. **Fuzzy Sets and Systems**, Amsterdam, v.35, n.1, p.105-13, March 1990.
- [111] YAMAKAWA, T. High-speed fuzzy controller hardware system: the Mega-FIPS machine. **Information Sciences**, New York, v.45, n.2, p.113-28, July 1988.
- [112] YAMAKAWA, T. Stabilization of an inverted pendulum by a high-speed fuzzy logic controller hardware system. **Fuzzy Sets and Systems**, Amsterdam, v.32, n.2, p.161-80, Sept. 1989.

- [113] YASUNOBU, S.; MYAMOTO, S. Automatic train operation system by predictive fuzzy control. In: SUGENO, M., ed. **Industrial Application of Fuzzy Control**. Amsterdam, North-Holland, 1985. p.1-18.
- [114] YASUNOBU, S.; HASEGAWA, T. Evaluation of a automatic container crane operation system based on fuzzy predictive fuzzy control. **Control - Theory and Advanced Technology**, Tokyo, v.2, n.3, p.419-32.
- [115] YAMAZAKI, T.; SUGENO, M. Fuzzy controls. **Systems and Controls**, v.28, p.442-46, 1984. (em japonês)
- [116] YING, H.; SILER, W.; BUCKLEY, J.J. Fuzzy control theory: a nonlinear case. **Automatica**, Amsterdam, v.26, n.3, p.513-20, March 1990.
- [117] ZADEH, L.A. Fuzzy sets. **Information and Control**, New York, v.8, n.3, p.338-353, June 1965.
- [118] ZADEH, L.A. Fuzzy algorithms. **Information and Control**, v.12, n.1-6, p.94-102, 1968.
- [119] ZADEH, L.A. A rationale of fuzzy control. **Transactions of ASME: Journal of Dynamics Systems, Measurement and Control**, serie G, New York, v.94, p.3-4, March 1972
- [120] ZADEH, L.A. Outline of a new approach to the analysis of complex systems and decision processes. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.3, n.1, p.28-44, Jan. 1973.
- [121] ZADEH, L.A. The concept of a linguistic variable and its application to approximate reasoning - I. **Information Sciences**, New York, v.8, n.3, p.199-249, 1975.
- [122] ZADEH, L.A. The concept of a linguistic variable and its application to approximate reasoning - II. **Information Sciences**, New York, v.8, n.4, p.301-57, 1975.
- [123] ZADEH, L.A. The concept of a linguistic variable and its application to approximate reasoning - III. **Information Sciences**, New York, v.9, n.1, p.43-80, 1975.
- [124] ZADEH, L.A. Fuzzy sets as a basis for a theory of possibility. **Fuzzy Sets and Systems**, Amsterdam, v.1, n.1, p.3-28, 1978.
- [125] ZADEH, L.A. The role of fuzzy logic in the management of uncertainty in expert systems. **Fuzzy Sets and Systems**, Amsterdam, v.11, n.3, p.199-227, Nov. 1983.
- [126] ZADEH, L.A. Syllogistic reasoning in fuzzy logic and its application to usability and reasoning with dispositions, **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.15, n.6, p.754-63, Nov./Dec. 1985.
- [127] ZADEH, L.A. Fuzzy logic. **IEEE Computer Magazine**, Los Alamitos, v.21, n.4, p.83-93, Apr. 1988.

- [128] ZADEH, L.A. Knowledge representation in fuzzy logic. **IEEE Transactions on Knowledge and Data Engineering**, New York, v.1, n.1, p.89-100, March 1989.