



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

RELATÓRIO TCC
DESENVOLVIMENTO DE UMA APOSTILA SOBRE CONTROLE FUZZY

Aluno: Mateus Assis Maximo de Lima
Professor: Antonio Marcus Nogueira Lima

Campina Grande - PB, Agosto de 2007

TCC

Mateus Assis Maximo de Lima

28 de agosto de 2007

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Unidade Acadêmica de Engenharia Elétrica

Controladores Fuzzy

Mateus Assis Maximo de Lima
Aluno

Antonio Marcus Nogueira Lima
Orientador

Sumário

1	Introdução	1
2	Conjuntos Fuzzy	2
2.1	Definição	2
2.2	Propriedades	2
2.3	Operações	3
2.3.1	União e Interseção	4
2.3.2	Complemento	5
3	Lógica Fuzzy	5
3.1	Proposições	6
3.1.1	Conectivos lógicos	6
3.1.2	Negação	8
3.2	Regras	8
3.2.1	Representação	8
3.2.2	Inferência	8
4	Controle Fuzzy	10
4.1	Desenvolvendo um controlador Fuzzy	10
4.1.1	Entradas e Saídas	11
4.1.2	Base de Regras	12
4.1.3	Quantificação fuzzy do conhecimento	16
4.1.4	Mecanismo de Inferência	18
4.1.5	Defuzzificação	21
5	Estudo de caso	25
5.1	Introdução	25
5.2	Fundamentação teórica	25
5.3	Resultados	28
5.4	Conclusões	29
6	Conclusão	30

Lista de Figuras

1	Conjunto Fuzzy representando altura dos indivíduos.	3
2	Peso, núcleo e suporte de um conjunto fuzzy.	3
3	Resultados dos operadores lógicos.	7
4	Exemplo do uso dos operadores de máximo e mínimo.	9
5	Diagrama de blocos de um controlador Fuzzy.	11
6	Pêndulo invertido em cima de um carro.	11
7	Operador humano dentro da malha de controle.	12
8	Estrutura de controle.	13
9	Pêndulo invertido em diversas posições.	15
10	Função de pertinência para o valor lingüístico “positivo pequeno”. . .	16
11	Exemplos de funções de pertinência.	17
12	Funções de pertinência para o pêndulo invertido.	18
13	Funções de pertinência dos antecedentes.	19
14	Funções de pertinência dos antecedentes para a regra 1.	20
15	Funções de pertinência dos antecedentes para a regra 2.	21
16	Agregação das saídas das regras 1 e 2.	22
17	Resumo do processo de fuzzificação e defuzzificação.	23
18	Conjuntos fuzzy utilizando o operador produto.	24
19	Processo [1]	25
20	Controlador proposto	26
21	Conjuntos fuzzy para entradas $\Delta erro\%$ e y^*	26
22	Conjuntos fuzzy para a saída Δu_c	27
23	Saídas para os controladores Fuzzy e PI para o ponto de operação 0,2	28
24	Saídas para os controladores Fuzzy e PI para o ponto de operação 1,0	28
25	Saídas para os controladores Fuzzy e PI para o ponto de operação 5,0	29

1 Introdução

Neste relatório será estudada a utilização de lógica fuzzy em sistemas de controle. A motivação para tal uso consiste basicamente da dificuldade da tarefa de modelagem e simulação de sistemas complexos com o intuito de projetar sistemas de controle. Mesmo que um modelo preciso de um sistema dinâmico seja desenvolvido, geralmente ele se torna muito complexo para ser utilizado no projeto do controlador, principalmente para muitos procedimentos que requerem simplificações restritivas.

A grande vantagem dos controladores fuzzy consiste da fácil incorporação do conhecimento de um especialista humano no sistema de controle. A abordagem fuzzy provê uma metodologia formal para representar, manipular, e implementar este conhecimento, tornando-o bastante versátil.

Atualmente, este tipo de controlador tem sido bastante utilizado. Dentre as diversas aplicações pode-se destacar:

- Aeronáutica: No controle de vôo, motores e diagnóstico de falhas;
- Automóveis: Freios, transmissão, suspensão e controle do motor;
- Controle de Processos: Temperatura, pressão e controle de nível, bem como no controle de uma coluna de destilação e processos de dessalinização;
- Robótica: Controle de posição e planejamento de rotas.

Este material tem como objetivo familiarizar o leitor com a abordagem fuzzy. Inicialmente, na seção 2, será estudada a teoria de conjuntos fuzzy, que serve como base para o desenvolvimento da lógica fuzzy, abordada na seção 3. Na seção 4 será feito um estudo da aplicação do conhecimento adquirido nas duas primeiras seções em sistemas de controle. Para tal, utilizou-se uma abordagem tutorial, onde acompanhou-se um processo de desenvolvimento de um controlador fuzzy a ser utilizado para controlar um pêndulo invertido. Na última seção, 5, será feito um estudo de caso do processo de desenvolvimento de um controlador fuzzy para controlar uma válvula não linear. Por fim, algumas conclusões serão apresentadas acerca da utilização deste tipo de controlador.

2 Conjuntos Fuzzy

2.1 Definição

Conjuntos fuzzy são uma extensão da teoria clássica de conjuntos, sendo utilizadas na lógica fuzzy. Ao contrário da teoria clássica, onde um elemento pertence ou não a um dado conjunto, a teoria de conjuntos fuzzy permite que um elemento pertença de forma gradual a um conjunto. Isto é descrito por uma função que define o grau de pertinência de um elemento, $\mu \rightarrow [0, 1]$ [2].

A lógica fuzzy é uma generalização da lógica clássica que admite valores lógicos fracionários, enquanto a lógica tradicional admite apenas o par oposto falso/verdadeiro. Sendo assim, a lógica fuzzy é uma tentativa de implementar níveis intermediários de verdade, chamados de tons de cinza.

Na teoria clássica de conjuntos, a função que define o grau de pertinência de um determinado elemento pode ser descrita da seguinte forma:

$$\mu_A(s) = \begin{cases} 1, & \text{se } x \in A \\ 0, & \text{se } x \notin A \end{cases} \quad (1)$$

Isto significa que um elemento, na teoria clássica, ou pertence a um conjunto ($\mu_A(x) = 1$) ou não ($\mu_A(x) = 0$). Para muitas classificações, porém, não fica muito claro se um elemento pertence a um conjunto ou não. Por exemplo, considerando-se um caso em que se classifica uma pessoa como alta ou baixa, tal que uma pessoa é classificada com alta a partir de 1,80 metro de altura, deseja-se estudar como devem ser consideradas pessoas com 1,78 ou 1,82 metro. Observa-se que neste intervalo de altura não fica muito bem definido se uma pessoa é alta ou não. Neste caso, a utilização de lógica fuzzy torna-se bastante útil, visto que pode-se definir graus de pertinência de um indivíduo quanto ao conjunto dos baixos e dos altos.

Um conjunto fuzzy, introduzido por Zadeh em 1965, é definido no intervalo: $\mu_A(x) \in [0, 1]$. Pode-se observar como seria uma função de pertinência para o exemplo descrito anteriormente na figura 1.

2.2 Propriedades

Várias propriedades podem ser definidas para os conjuntos fuzzy. Neste relatório será abordado apenas as mais relevantes, destacando-se o *peso*, o *núcleo* e o *suporte*.

O *peso* de um conjunto fuzzy A, $hgt(A)$, é definido por:

$$hgt(A) = \sup_{x \in X} \mu_A(x) \quad (2)$$

Quando o *peso* de um conjunto é igual a 1, diz-se que este é um conjunto *normal*. Conjuntos *subnormais* são caracterizados por $hgt(A) < 1$. O *núcleo* de um conjunto fuzzy, é um subconjunto de A, tal que:

$$core(A) = \{x \in X | \mu_A(x) = 1\} \quad (3)$$

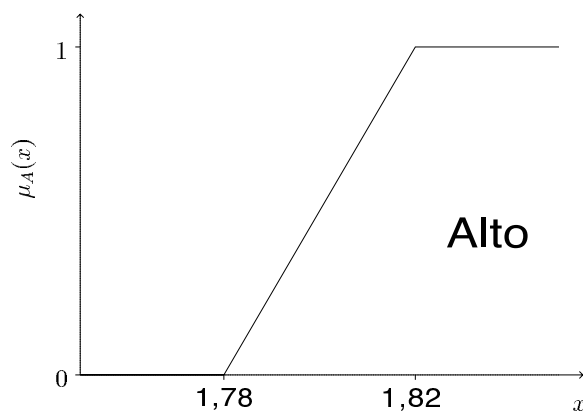


Figura 1: Conjunto Fuzzy representando altura dos indivíduos.

O *suporte* de um conjunto fuzzy é um subconjunto, tal que:

$$\text{supp}(A) = \{x \in X \mid \mu_A(x) > 0\} \quad (4)$$

Quando o *suporte* de um conjunto fuzzy é finito, diz-se que este é um *suporte compacto*. Observa-se na figura 2 uma demonstração dessas propriedades.

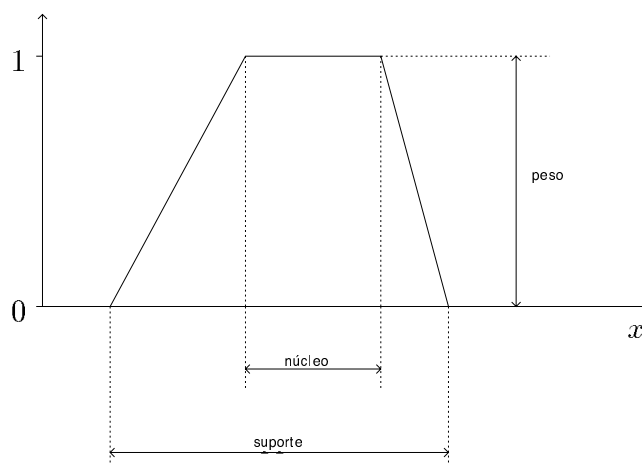


Figura 2: Peso, núcleo e suporte de um conjunto fuzzy.

2.3 Operações

Bem como na teoria de conjuntos clássicos, operações similares são definidas para os conjuntos fuzzy. A interseção e a união de dois conjuntos, bem como o comple-

A	B	A and B	A or B	not A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

mento são realizados da mesma forma que na teoria clássica, como pode ser observado na tabela 2.3.

Apesar disso, devido ao fato das funções de pertinência poderem assumir valores intermediários no intervalo $[0,1]$, estes operadores não podem ser unicamente definidos. Nas próximas seções será exposto a maneira pela qual essas operações são realizadas.

2.3.1 União e Interseção

Conjuntos fuzzy são normalmente interligados através de *interseções* e *uniões*. É fato que a interseção e a união entre conjuntos fuzzy devem resultar nos mesmos valores que a teoria clássica quando a função de pertinência for igual a 0 ou 1, visto que um conjunto clássico pode ser visto como um caso especial de um conjunto fuzzy. Zadeh propôs, em 1965, as seguintes definições:

$$\begin{aligned}\mu_{A \cap B} &= \min(\mu_A(x), \mu_B(x)) \text{ (interseção)} \\ \mu_{A \cup B} &= \max(\mu_A(x), \mu_B(x)) \text{ (união)}\end{aligned}\tag{5}$$

Como pode ser observado pela definição 5, quando as funções de pertinência forem iguais a 1 ou 0, o resultado é o mesmo da teoria clássica. Geralmente as operações de união e interseção são realizadas utilizando *normas triangulares*, norma T ou conorma T (norma S), definidas a seguir.

Definição 1. Dada uma transformação $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$. T é uma norma T se, e somente se, para todo $a, b, c \in [0, 1]$:

$$T(a, b) = T(b, a)\tag{6}$$

$$T(a, b) \leq T(a, c), \text{ se } b \leq c\tag{7}$$

$$T(a, T(b, c)) = T(T(a, b), c)\tag{8}$$

$$T(a, 1) = a\tag{9}$$

A norma T é utilizada para representar além da interseção, o conector lógico *and* entre duas proposições. Apesar de Zadeh ter proposto originalmente o operador *min*, atualmente o operador mais utilizado é o *produto*, visto que este não insere a não-linearidade resultante da operação de *mínimo*.

Definição 2. Dada uma transformação $S : [0, 1] \times [0, 1] \rightarrow [0, 1]$. S é uma norma S se, e somente se, para todo $a, b, c \in [0, 1]$:

$$S(a, b) = S(b, a) \quad (10)$$

$$S(a, b) \leq S(a, c), \text{ se } b \leq c \quad (11)$$

$$S(a, S(b, c)) = S(S(a, b), c) \quad (12)$$

$$S(a, 0) = a \quad (13)$$

Várias funções que seguem as definições 1 ou 2, podem ser utilizadas para as operações de interseção, união e conectores lógicos, como pode ser observado nos exemplos dispostos na tabela 1.

norma T	norma S	Propositor
$\min(a,b)$	$\max(a,b)$	Zadeh
$\max(a+b-1,0)$	$\min(a+b,1)$	Lukasiewicz
ab	$a+b-ab$	Probabilidade

Tabela 1: Operadores mais comuns.

2.3.2 Complemento

O operador de complemento para conjuntos fuzzy é definido da seguinte forma:

Definição 3. Dada uma transformação $\bar{T} : [0, 1] \rightarrow [0, 1]$. \bar{T} é o complemento λ de T se, e somente se:

$$\mu_{\bar{T}^\lambda}(x) = \frac{1 - \mu_A(x)}{1 + \lambda \mu_A(x)} \quad (14)$$

$$\bar{T}(T) = T \quad (15)$$

3 Lógica Fuzzy

No raciocínio fuzzy a decisão é baseada na informação da entrada e no conhecimento, descrito na forma de regras. As regras possuem tipicamente um número de proposições fuzzy no antecedente e a consequência pode incluir ou proposições fuzzy ou uma função das entradas. O método de raciocínio é classificado baseado na forma da consequência[3] que, neste relatório está restrito às regras definidas por Mandami.

Assim como a teoria clássica de conjuntos serve como base para a lógica clássica, a teoria de conjuntos fuzzy serve como base para a lógica fuzzy. Isto significa que as operações teóricas em conjuntos fuzzy são a base para as operações lógicas. As operações definidas para conjuntos, como união, interseção e complemento, têm um

correspondente significado lógico, como *ou(or)*, *e(and)* e *negação(not)*, respectivamente. Apesar disso, no caso de conjuntos fuzzy, estas operações possuem várias representações ao invés de apenas uma, como no caso da teoria clássica. Isto, obviamente, também é verdade para os operadores lógicos na lógica fuzzy.[4]

3.1 Proposições

Uma proposição fuzzy representa uma declaração “ x é A ”, onde x é uma variável e A é um rótulo lingüístico, representado por um conjunto fuzzy no universo de discurso da variável x [4]. Os *rótulos fuzzy* também são referidos como *constantes fuzzy*, *termos fuzzy* ou *notações fuzzy*. As proposições fuzzy conectam variáveis com significados lingüísticos definidos por estas variáveis.

Estas proposições são a base do processo de inferência da lógica fuzzy, visto que estas proposições podem ser combinadas utilizando conectivos lógicos, tais como *and* ou *or*. Além disso, ainda pode-se utilizar modificadores lingüísticos para modificar o significado de uma expressão lingüística. Como exemplo, pode-se utilizar o modificador *muito* para mudar o significado de uma expressão “ x é positivo” para “ x é *muito* positivo”.

3.1.1 Conectivos lógicos

Assim como na teoria clássica, proposições fuzzy podem ser combinadas utilizando-se conectivos lógicos *or* e *and*. Contudo, no caso da lógica fuzzy, estes conectivos são implementados utilizando normas T e normas S, respectivamente. Não existe nenhum guia que defina que norma T ou norma S deve ser utilizada em uma determinada situação. Apesar disso, levando-se em conta algumas propriedades das normas T e S, o uso de específicas normas T e S pode ser justificado intuitivamente em alguns casos. Na figura 3 observa-se o resultado da utilização das normas T e S definidas por Lukasiewicz comparado com as normas definidas por Zadeh para um mesmo universo de discurso.

Os operadores propostos originalmente por Zadeh possuem a vantagem de que a redundância é ignorada, ou seja, a combinação de duas proposições fuzzy iguais representam a mesma informação:

$$\mu_{A \cap A}(x) = \min(\mu_A(x), \mu_A(x)) = \mu_A(x) \quad (16)$$

$$\mu_{A \cup A}(x) = \max(\mu_A(x), \mu_A(x)) = \mu_A(x) \quad (17)$$

Apesar da vantagem citada acima, quando proposições fuzzy não são iguais mas são correlacionadas ou interativas, justifica-se o uso de outros operadores ao invés do *min* e *max*. A correlação ou interatividade representa casos onde existem dependência entre as proposições. Nestes casos pode-se, por exemplo, utilizar as normas definidas por Lukasiewicz, também conhecidas como *somas limitadas*.

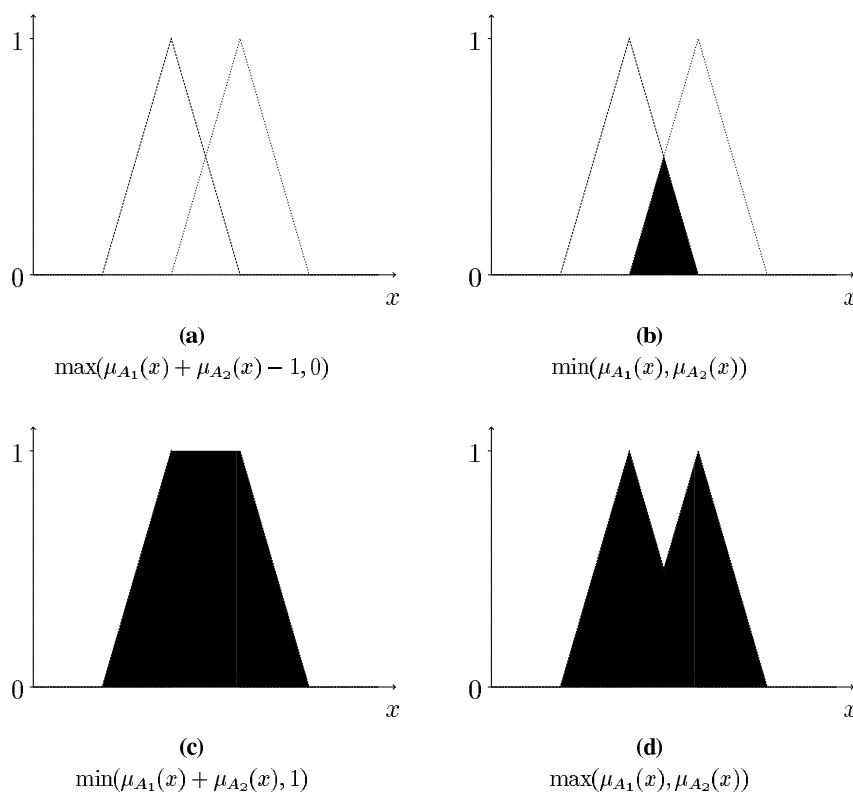


Figura 3: Resultados dos operadores lógicos.

Como pode ser observado na figura 3c, o conectivo *or* de duas proposições fuzzy resulta em um conjunto fuzzy que também possui uma função de pertinência completa para valores da variável “entre” dois conjuntos.[4]

Como pode ser observado, a escolha de que normas devem ser utilizadas para os conectivos lógicos depende do significado e do contexto em que as proposições foram definidas e as relações entre elas.

Se as proposições estão relacionadas a diferentes universos de discussão, um conectivo lógico resultará em uma relação fuzzy como demonstrado no exemplo a seguir:

$$p : x_1 \text{ é } A_1 \text{ and } x_2 \text{ é } A_2$$

Onde A_1 e A_2 possuem funções de pertinência $\mu_{A_1}(x_1)$ e $\mu_{A_2}(x_2)$. A proposição p pode então ser representada pela relação fuzzy P com função de pertinência definida por:

$$\mu_P(x_1, x_2) = T(\mu_{A_1}(x_1), \mu_{A_2}(x_2)) \quad (18)$$

3.1.2 Negação

Da mesma forma que os conectivos lógicos podem ser relacionados às operações de interseção e união de conjuntos fuzzy, a negação em uma proposição fuzzy pode ser relacionado ao complemento de um conjunto fuzzy.

Apesar de que vários complementos fuzzy são possíveis, geralmente o complemento padrão é usado, definido como:

$$\mu_{not(A)}(x) = 1 - \mu_A(x)$$

3.2 Regras

Para tornar possível a utilização da lógica fuzzy, o conjunto de regras fuzzy deve ser representada por uma função de implicação. Tal função tem o mesmo intuito da tabela verdade para a teoria clássica de conjuntos. Na lógica clássica a implicação é descrita como:

$$A \rightarrow B$$

Que pode ser entendida da seguinte forma:

se A então B

Na teoria da lógica fuzzy esse tipo de declaração é normalmente referida como regras *if-then*(se-então) ou regras fuzzy.

3.2.1 Representação

Quando uma regra possui proposições fuzzy combinadas por conectivos lógicos tanto no antecedente como na consequência, dá-se a ela o nome de regra fuzzy Mandami. Ela é normalmente definida da seguinte forma:

$$\text{se } x_1 \text{ é } X_1^i \text{ and } x_2 \text{ é } X_2^i \text{ and } \dots \text{ and } x_{n_x} \text{ é } X_{n_x}^i \text{ então } z \text{ é } Z^i$$

Onde X_j^i é o conjunto fuzzy da j -ésima entrada (n_x é o número de entradas) e Z^i é o conjunto fuzzy da saída relacionados à i -ésima regra.

3.2.2 Inferência

O mecanismo de inferência Mandami é realizado da seguinte forma. Inicialmente a *verdade* da regra é calculada de acordo com os conectivos incluídos no antecedente. Como no exemplo apresentado, a regra possui apenas conectivos *and*, o conjunto fuzzy da saída é obtido baseando-se no “nível de verdade” e no conjunto fuzzy resultante da aplicação da norma T, sendo portanto calculado da seguinte forma:

$$\begin{aligned}\psi_i &= T(\mu_{X_1^i}(x_1), \dots, \mu_{X_{n_x}^i}(x_{n_x})) \\ \mu_{Z^{i'}}(z) &= T(\psi_i, \mu_{Z^i}(z)) \forall z \in \mathbb{R}\end{aligned}\quad (19)$$

A consequência de todas as regras são agregadas à consequência final utilizando-se norma S:

$$\mu_{Z'}(z) = S(\mu_{Z^{1'}}(z), \dots, \mu_{Z^{m'}}(z)) \forall z \in \mathbb{R} \quad (20)$$

Onde m é o número das regras. Observa-se que tanto ψ_i como $\mu_{Z^{i'}}$ e $\mu_{Z'}(z)$ são funções da entrada x .

O processo de inferência pode ser observado na figura 4 para um sistema com duas entradas e duas regras, sendo estas:

$$\begin{aligned}\text{se } x_1 \text{ é } X_1^1 \text{ and } x_2 \text{ é } X_2^1 \text{ então } z \text{ é } Z^1 \\ \text{se } x_1 \text{ é } X_1^2 \text{ and } x_2 \text{ é } X_2^2 \text{ então } z \text{ é } Z^2\end{aligned}$$

Na figura 4 utiliza-se o operador *min* como norma T, e o operador *max* como norma S.

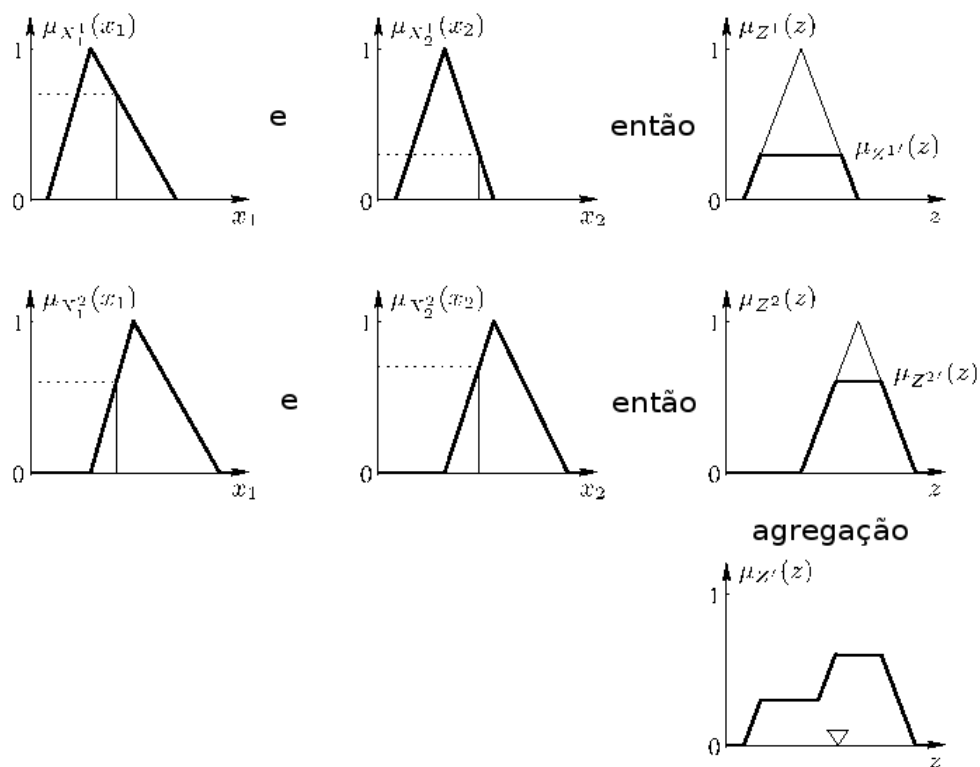


Figura 4: Exemplo do uso dos operadores de máximo e mínimo.

4 Controle Fuzzy

O principal objetivo da engenharia de controle é aplicar o conhecimento sobre como controlar um processo de tal sorte que o sistema de controle resultante apresente um desempenho desejado de maneira confiável e segura. Nesta seção será apresentado como a lógica fuzzy provê uma metodologia para representar e implementar o conhecimento de um controlador humano sobre como melhor controlar um processo.

Nas próximas seções será adotada uma abordagem tutorial, focando na elaboração de um controlador para o controle de um pêndulo invertido, que possui duas entradas e uma saída, observando todo o processo de desenvolvimento do controlador utilizando as operações fuzzy mais comuns.

Por fim, o conhecimento obtido na seção tutorial será aplicado a um pêndulo invertido com dados reais, tornando necessário um escalonamento da saída do controlador, culminando em um sistema que apresentará o comportamento desejado.

4.1 Desenvolvendo um controlador Fuzzy

A estrutura básica de um sistema de controle fuzzy, muitas vezes chamado de controlador de lógica fuzzy (FLC), encontra-se disposta na figura 5. Nela observam-se os seguintes elementos:

1. Uma *Base de Regras*, que contém uma quantificação em lógica fuzzy de conhecimento de um especialista descrevendo, de maneira lingüística, como conseguir um bom desempenho do controlador.
2. Um *Mecanismo de Inferência*, que simula a decisão de um especialista quando interpretando e aplicando o seu conhecimento no controle de um sistema.
3. Um interface de *Fuzzyficação*, que converte as entradas do controlador em uma informação que o mecanismo de inferência pode utilizar para ativar e aplicar as regras corretamente.
4. Uma interface de *Defuzzificação*, que converte as conclusões do mecanismo de interface em sinais de entrada para o processo.

Com a finalidade de desenvolver um controlador fuzzy de maneira intuitiva, será utilizado como exemplo de abordagem um sistema simples de um pêndulo invertido colocado em cima de um carro, como apresentado na figura 6. Nela, a variável y denota o ângulo que o pêndulo faz com o eixo vertical (em radianos), l define o tamanho da metade do tamanho do pêndulo (em metros), e u é a força que impulsiona o carro (em Newtons). Para denotar a posição angular do pêndulo será utilizada a variável r .

O objetivo do controle deste sistema é fazer com que o pêndulo se mantenha equilibrado na posição central, ou seja, $r = 0$, dado que inicialmente o pêndulo encontra-se com um ângulo diferente de zero com o eixo vertical. Como se trata de um sistema

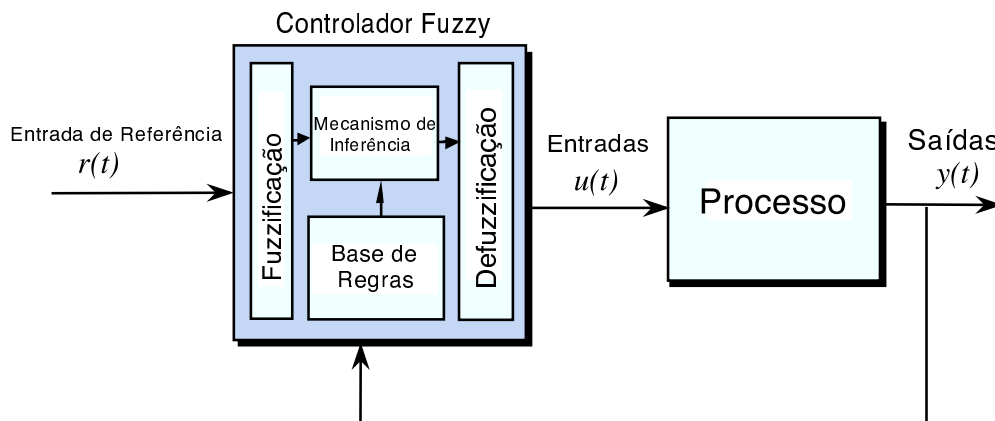


Figura 5: Diagrama de blocos de um controlador Fuzzy.

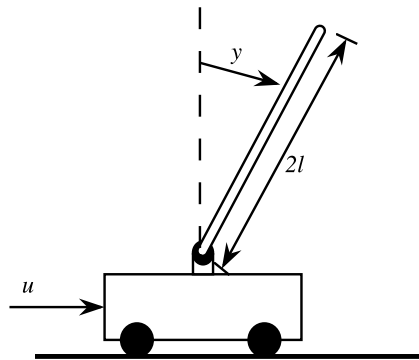


Figura 6: Pêndulo invertido em cima de um carro.

bastante simples normalmente utilizado como exemplo acadêmico de controle de um sistema não-linear, muitas técnicas já foram utilizadas para sua solução, tal como um simples PID, que por sua vez apresenta um bom resultado.

Apesar de provavelmente não ser realmente necessária a utilização de um controlador fuzzy para este sistema, este será utilizado como um problema conveniente para ilustrar o processo de desenvolvimento deste tipo de controlador.

4.1.1 Entradas e Saídas

No processo de desenvolvimento considera-se o comportamento de um humano controlando o sistema dentro da malha de controle, como demonstrado na figura 7. O objetivo do controlador fuzzy é automatizar o comportamento de um especialista humano quando controlando o sistema com sucesso. Inicialmente, o especialista define que informações são necessárias como entrada do controlador para ser utilizado no processo de decisão. Para o caso em questão serão considerados como entradas:

$$e(t) = r(t) - y(t) \text{ e}$$

e

$$\frac{d}{dt}e$$

Certamente, várias outras escolhas poderiam ser feitas, mas está é bem intuitiva. Além disso, faz-se necessário identificar que variável deve ser controlada. Para o caso do pêndulo só é permitido controlar a força que movimenta o carro. Sendo assim, esta será a variável a ser controlada.

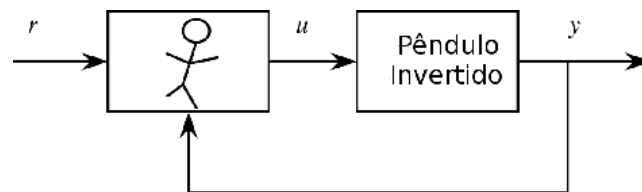


Figura 7: Operador humano dentro da malha de controle.

A escolha das entradas e saídas do controlador pode ser mais difícil para aplicações mais complexas. Essencialmente, deve-se ter certeza de que o controlador terá a informação correta disponível para que este possa tomar boas decisões, possibilitando ao sistema ser controlado na direção correta para que se obtenha uma operação com alto desempenho.

Uma vez tendo escolhido as entradas e saídas, deve-se determinar quais serão as entradas de referência. Para o caso em questão esta escolha está bem clara. Contudo, em algumas situações, deve-se escolher o sinal de referência r como uma constante não nula para balancear o pêndulo e colocá-lo na posição vertical. Para tal, o controlador deverá manter o carro em constante aceleração para que, desta maneira, o pêndulo não caia.

De posse de todas essas informações define-se o controlador fuzzy. Para o caso em questão, com as escolhas já feitas, resulta-se no sistema apresentado na figura 8. Pode-se ver que a escolha das entradas e saídas do controle impõem algumas restrições no tocante ao processo de desenvolvimento do controlador fuzzy. Se a informação correta não for dada ao controlador, ficará difícil projetar uma boa base de regras ou mecanismo de inferência. Além disso, mesmo que a informação correta estiver disponível para tomar decisões de controle, ela será de pouco uso se o controlador não é capaz de afetar corretamente as variáveis do processo através das entradas deste.[5]

4.1.2 Base de Regras

Supondo que um especialista forneça uma descrição de como é a melhor maneira de controlar a planta, observa-se que de alguma forma, esta descrição lingüística deve

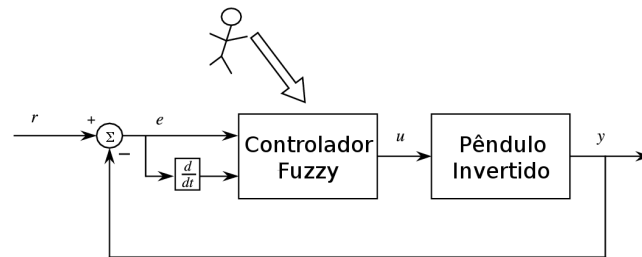


Figura 8: Estrutura de controle.

ser agregada ao controlador, como indicado na figura 8.

Descrições lingüísticas

Normalmente, descrições lingüísticas podem ser subdivididas em algumas partes. Devem haver “variáveis lingüísticas” que descrevem cada um das entradas e saídas variantes no tempo do controlador o que, para o caso em questão pode ser definido como:

“erro” descreve $e(t)$
 “variação do erro” descreve $\frac{d}{dt}e$
 “força” descreve $u(t)$

Existem diversas possíveis escolhas para descrições lingüísticas das variáveis. Alguns projetistas preferem escolher nomes descritivos para fins de documentação. Porém, tal prática as vezes resulta em descrições muito extensas. Outros preferem manter as variáveis com nomes mais concisos. Contudo, a escolha das variáveis lingüísticas não tem nenhum impacto na forma com que o controlador funciona, visto que se trata apenas de uma notação que visa facilitar a construção do controlador utilizando lógica fuzzy.

Da mesma forma que a variável $e(t)$ assume, por exemplo, um valor de 0.1 para $t=2$, variáveis lingüísticas assumem “valores lingüísticos”. Para o caso do pêndulo invertido pode-se utilizar como valores para as variáveis “erro”, “variação do erro” e “força”, por exemplo, os seguintes valores:

Exemplo 1	Exemplo 2
negativo grande	-2
negativo pequeno	-1
zero	0
positivo pequeno	1
positivo grande	2

No exemplo 1, obtêm-se nomes mais descritivos sobre o valor que a variável apresenta naquele momento. Porém, adotando-se a notação apresentada no exemplo 2,

nota-se que a descrição mantém a descrição lingüística mas mantém-se pequena, sendo portando a adotada daqui para frente.

Observa-se que o valor “-1” nada tem a ver com uma quantidade numérica. Ele apenas indica o sinal e o quão grande é o valor de uma determinada variável. A utilização deste tipo de linguagem torna mais fácil extrair idéias a respeito do processo de decisão de um especialista quando controlando um determinado sistema tendo como entradas e saídas os atributos escolhidos anteriormente.

No tocante às variáveis, as seguintes relações podem ser apresentadas:

$$\begin{aligned} e &= r - y, \text{ mas } r = 0, \text{ logo:} \\ e &= -y, \text{ e:} \\ \frac{d}{dt}e &= -\frac{d}{dt}y \end{aligned} \quad (21)$$

Ao projetista cabe avaliar como a declaração de que uma determinada variável possui um determinado valor lingüístico influencia no estado do pêndulo e, desta maneira, tornar-se apto a entender como o especialista quantifica a dinâmica do sistema. Tal tarefa pode, dependendo do problema, tornar-se mais complicada, mas nota-se que um melhor entendimento da dinâmica do processo geralmente leva a uma melhor quantificação.

Regras

Tendo realizado uma quantificação lingüística do estado das variáveis deve-se, agora, especificar uma base de regras que capture o conhecimento de um especialista sobre como controlar a planta. Em particular, para o caso do pêndulo invertido nas três posições apresentadas na figura 9, as seguintes regras podem ser definidas:

1. *Se o erro for negativo grande e a variação do erro for negativa grande, a força deve ser positiva grande.*

Esta regra refere-se à situação apresentada na figura 9(a) em que o pêndulo apresenta um ângulo grande e positivo e se encontra girando em sentido horário. Analisando este estado fica claro de que a aplicação de uma força positiva e grande fará com que o pêndulo mova-se em direção à posição de equilíbrio.

2. *Se o erro for zero e a variação do erro for positiva pequena, a força deve ser negativa pequena.*

Esta regra refere-se à situação apresentada na figura 9(b), onde o pêndulo está próximo da posição de equilíbrio e está se movendo em sentido anti-horário. Neste caso, faz-se necessária a aplicação de uma força da direita para esquerda (negativa) pequena, o que fará com que o sistema tenda a posição de equilíbrio.

3. Se o erro for **positivo alto** e a **variação do erro** for **negativa pequena**, a força deve ser **negativa pequena**.

Esta regra refere-se à situação apresentada na figura 9(c), em que observa-se que o pêndulo está longe e a esquerda do eixo central, porém encontra-se deslocando-se em sentido horário, ou seja, na direção do equilíbrio. Neste caso, a aplicação de uma força negativa e pequena colaborará para este movimento.

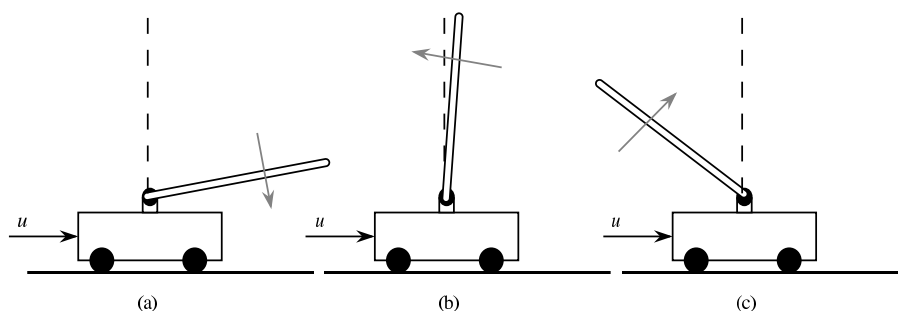


Figura 9: Pêndulo invertido em diversas posições.

Observa-se que as regras apresentam a mesma estrutura apresentada na seção 3.2 em que os antecedentes estão associados às entradas no controlador e os conseqüentes às saídas. Nota-se ainda que cada antecedente, bem como o conseqüente, são compostos por uma junção de diversos “termos” e que, a quantidade destes termos está limitada ao número de entradas e saídas respectivamente.

A base de regras

Se o processo descrito anteriormente para a definição das regras for continuado, pode-se continuar escrevendo regras para todos os possíveis casos em que o pêndulo pode se encontrar. Visto que o número de variáveis e valores lingüísticos são limitados, existe um conjunto finito de regras que contempla todas estas possibilidades. Para o caso do pêndulo invertido, com duas entradas e cinco valores lingüísticos para cada uma, o número total de combinações é $5^2 = 25$.

Uma forma bastante usual de se escrever uma base de regras, principalmente quando este conjunto vai se tornando cada vez maior, é utilizar uma representação em forma de tabela. Uma representação deste tipo para o caso do pêndulo invertido encontra-se disposto na tabela 4.1.2. Nela, uma combinação linha-coluna, em que a linha define o valor lingüístico do erro e a coluna da variação do erro, resulta em um valor lingüístico que deve ser atribuído ao conseqüente.

Nota-se que a matriz resultante apresenta uma diagonal com zeros, além de uma certa simetria, que ocorre devido ao caráter inerentemente simétrico do sistema. Este tipo de estrutura é bem comum, sendo muitas vezes exploradas durante o processo de desenvolvimento do controlador.

Força u		Variação no Erro \dot{e}				
		-2	-1	0	1	2
Erro e	-2	2	2	2	1	0
	-1	2	2	1	0	-1
	0	2	1	0	-1	-2
	1	1	0	-1	-2	-2
	2	0	-1	-2	-2	-2

4.1.3 Quantificação fuzzy do conhecimento

Até aqui foi quantificado, de maneira abstrata, o conhecimento de um especialista humano de como controlar uma planta, utilizando para isso apenas variáveis lingüísticas. Nesta seção será apresentado como a lógica fuzzy quantifica as descrições lingüísticas de tal sorte que automatiza, no controlador fuzzy, as regras de controle especificadas pelo especialista.[5]

Funções de pertinência

Para quantificar o significado de variáveis lingüísticas, a lógica fuzzy utiliza “funções de pertinência”. Como exemplo, pode-se observar a figura 10. A função μ quantifica o grau de pertinência de um determinado valor de $e(t)$ com relação ao conjunto dos “positivos pequenos”.

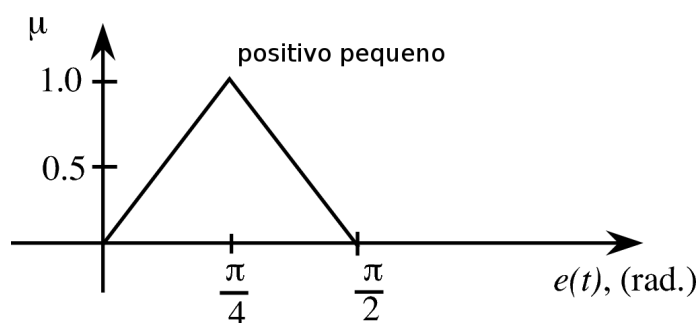


Figura 10: Função de pertinência para o valor lingüístico “positivo pequeno”.

É importante notar que a função de pertinência apresentada na figura 10 é apenas uma possível definição do significado do “erro é positivo pequeno”. Pode-se utilizar funções com forma de sino, trapezóides, dentre outras. Vários exemplos de formas de funções de pertinência podem ser observados na figura 11.

Como pode ser visto, dependendo da aplicação e do projetista, diferentes escolhas podem ser feitas para as funções de pertinência visto que cada tipo de função exprime um comportamento diferente para a classificação de uma determinada entrada.

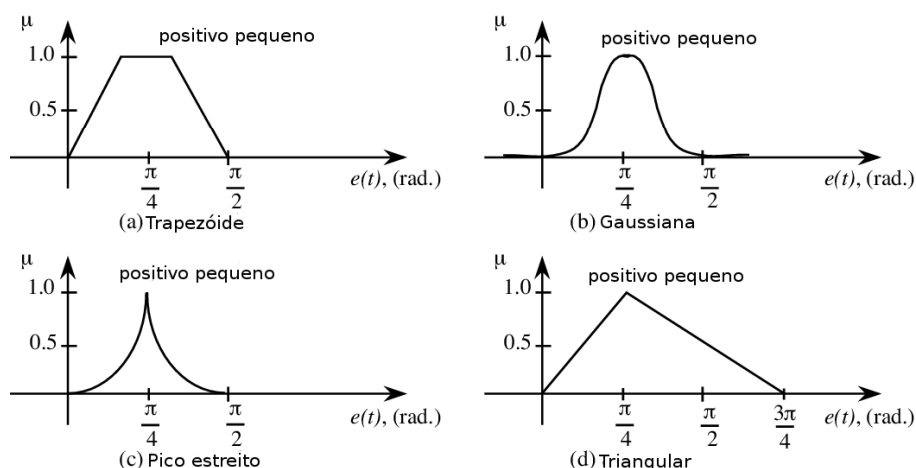


Figura 11: Exemplos de funções de pertinência.

Assim como o eixo vertical da figura 10 representa a certeza, o eixo horizontal também tem um nome especial. Ele é chamado de “universo de discurso” da entrada $e(t)$, visto que ele provê a faixa de valores de $e(t)$ que pode ser quantificado pelos conjuntos fuzzy. Na terminologia convencional, um universo de discurso de uma entrada ou saída de um sistema fuzzy é simplesmente a faixa de valores que as entradas e saídas podem assumir.[5]

Agora que já foi estudado como especificar o significado de um valor lingüístico através de uma função de pertinência, pode-se facilmente especificar as funções de pertinência para todos os 15 valores lingüísticos do exemplo do pêndulo inverso. Estas funções encontram-se dispostas na figura 12.

As funções de pertinência dos extremos merecem uma atenção em especial. Observa-se que para as entradas $e(t)$ e $\frac{d}{dt}e$ essas funções “saturam” para um valor de um. Isto faz sentido visto que para um especialista, agrupar todos os valores muito altos numa única descrição lingüística caracteriza apropriadamente o fenômeno de “maior que” ou “menor que” bastante comum no controle de processos.

Para a saída u , as funções de pertinência dos extremos não podem saturar, visto que no processo de decisão que está sendo abordado, deseja-se que a saída do controlador especifica um valor exato para a entrada do sistema. Ou seja, o controlador não irá indicar ao atuador do processo que, “qualquer valor maior que k é aceitável”.

Fuzzificação

O processo de Fuzzificação é, em linhas gerais, o ato de obter um valor da variável de entrada e encontrar os valores numéricos da pertinência desse valor às funções de pertinência. Por exemplo, se $e(t) = \pi/4$ e $\frac{d}{dt}e = \pi/16$, o processo de fuzzificação encontrará os valores das funções de pertinência para esse valor, obtendo:

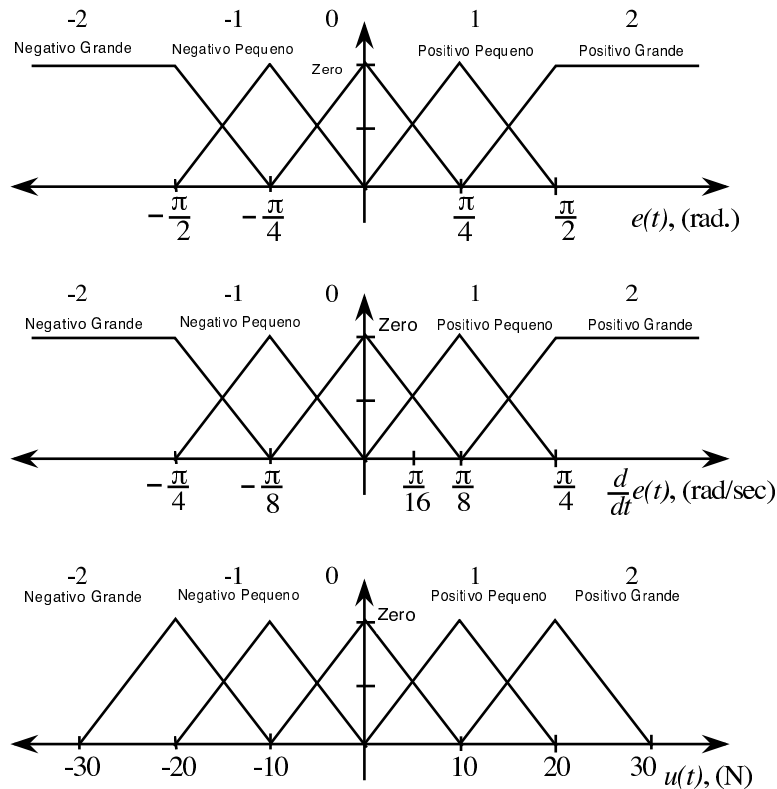


Figura 12: Funções de pertinência para o pêndulo invertido.

$$\mu_{\text{positivobaixo}}(e(t)) = 1$$

com todas as outras funções sendo igual a zero e:

$$\mu_{\text{zero}}\left(\frac{d}{dt}e(t)\right) = \mu_{\text{positivobaixo}}\left(\frac{d}{dt}e(t)\right) = 0.5$$

Este processo também pode ser visto como uma “codificação” dos valores de entrada, para que estes possam ser utilizados no processo de inferência do controlador.

4.1.4 Mecanismo de Inferência

O mecanismo de inferência geralmente envolve dois passos:

1. Os antecedentes de todas as regras são comparados às entradas do controlador para determinar que regras estão ativas para a situação atual. Este processo envolve a determinação do grau de certeza que cada regra possui para determinar que regra deve ser predominante nesta situação.

2. As consequências são determinadas utilizando as regras que estão ativas no presente momento, definindo portanto a saída do controlador naquele momento.

Quantização dos antecedentes

Para executar o mecanismo de inferência quantifica-se a pertinência dos antecedentes das regras que são compostas de vários termos, onde cada termo envolve uma entrada do controlador. Considerando como exemplo a figura 13, a qual se refere a seguinte regra:

Se o erro é zero e a variação do erro é positiva pequena, então a força é negativa pequena.

Através do processo descrito na figura 13, o significado dos termos “erro é zero” e “variação do erro é positivo pequeno” foram quantificados. Deseja-se portanto, avaliar como a utilização do conectivo *and*(e) opera sobre esses valores para produzir a saída.

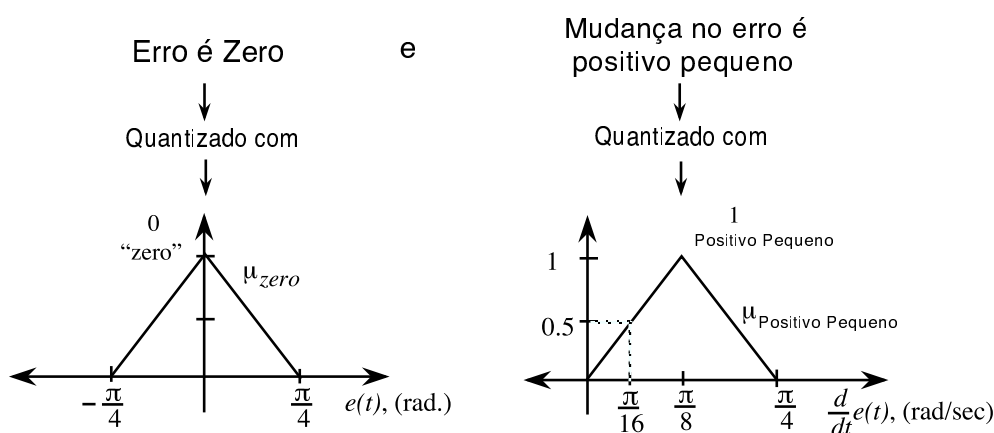


Figura 13: Funções de pertinência dos antecedentes.

Para este exemplo, será suposto que $e(t) = \pi/8$ e $\frac{d}{dt}e(t) = \pi/32$. Desta forma, utilizando-se a figura 13 obtém-se:

$$\mu_{zero}(e(t)) = 0.5$$

e

$$\mu_{positivopequeno}\left(\frac{d}{dt}e(t)\right) = 0.25$$

Para obter o resultado da regra pode-se utilizar vários tipos de normas T, dentre as quais destaca-se os operadores de *min* e do *produto*, que resultam nos seguintes valores:

Mínimo: $\mu_{antecedente} = \min(0,5; 0,25) = 0,25$

Produto: $\mu_{antecedente} = 0,5 \cdot 0,25 = 0,125$

A escolha entre os diversos operadores é uma definição do projetista. Algumas considerações a respeito dos operadores encontra-se na seção 2.

Determinando que regras estão ativas

Nesta seção será estudado como determinar que conclusões devem ser implementadas quando as regras que estão ativadas são aplicadas para decidir a força que deve ser aplicada ao carro que carrega o pêndulo invertido. Para tal, considera-se inicialmente as recomendações de cada regra independentemente. Logo após, estas recomendações são combinadas e a força é calculada.

Para melhor ilustrar o processo de agregação entre recomendações de duas regras, será apresentado inicialmente a saída do sistema para as duas regras separadamente e, logo após, a agregação delas para obter o resultado final.

Considerando a conclusão obtida pela regra:

Se o erro é zero e a variação do erro é zero, então a força é zero.

Utilizando o operador de mínimo para realizar a composição dos antecedentes da regra, obtém-se:

$$\mu_{antecedentes}(1) = \min(0,25; 1) = 0,25$$

A notação $\mu_{antecedentes}(1)$ representa a função de pertinência resultante da regra (1) e o resultado obtido indica que existe uma certeza de 0,25 que esta regra se aplica a atual situação. A função de pertinência do conseqüente encontra-se disposto na figura 14(a). A função de pertinência da conclusão alcançada pela regra 1, denotada por $\mu(1)$, está disposta na figura 14(b) e é dada por:

$$\mu(1)(u) = \min(0,25; \mu_{zero}(u))$$

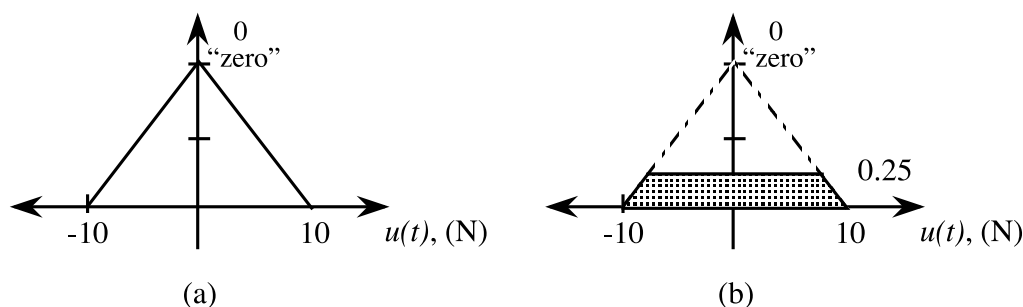


Figura 14: Funções de pertinência dos antecedentes para a regra 1.

Considerando agora a seguinte outra regra:

Se o erro é zero e a variação do erro é positiva pequena, **então** a força é negativa pequena.

A função de pertinência do resultado da composição dos antecedentes desta regra, a ser chamada de agora em diante de regra(2), é a seguinte:

$$\mu_{\text{antecedentes}}(2) = \min(0,75; 1) = 0,75$$

Sendo assim, observa-se que esta outra regra apresenta um maior nível de certeza que a regra(1). A função de pertinência para a conclusão obtida pela regra, chamada de $\mu(2)$, encontra-se disposta na figura 15, sendo esta definida por:

$$\mu(2)(u) = \min(0,75; \mu_{\text{negativopequeno}}(u))$$

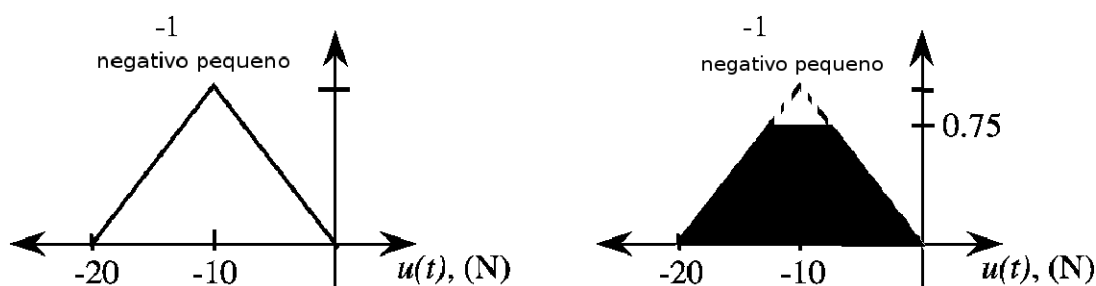


Figura 15: Funções de pertinência dos antecedentes para a regra 2.

4.1.5 Defuzzificação

Nesta seção será estudada o último componente do controlador fuzzy, o Defuzzificador. Este componente opera sobre os conjuntos fuzzy produzidos pelo mecanismo de inferência e combina os efeitos das diversas regras para obter uma composição delas que será utilizada como saída do controlador.

Para melhor entender o processo de defuzzificação observa-se o gráfico resultante da agregação das consequências das regras 1 e 2 na figura 16. Deseja-se obter uma saída u que represente da melhor forma possível as conclusões do controlador fuzzy para as diversas regras. Existem muitas abordagens para o processo de Defuzzificação porém, neste documento, serão abordados apenas dois tipos.

Combinando recomendações

Devido a sua grande popularidade, o método de Defuzzificação chamado “centro de gravidade” (COG) será abordado inicialmente. Este método combina todas as recomendações representadas pelas funções de pertinência inferidas pelas regras,

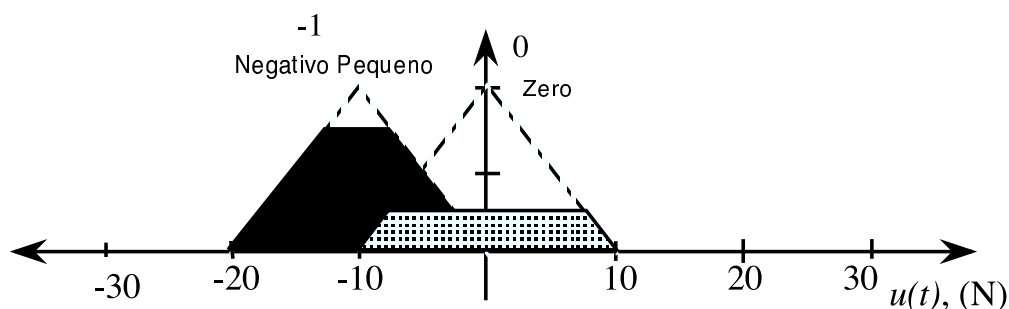


Figura 16: Agregação das saídas das regras 1 e 2.

apresentando na saída uma composição bastante utilizada. Define-se b_i como centro da função de pertinência da consequência da regra (i). Para o caso exposto na seção anterior, tem-se:

$$b_1 = 0 \text{ e } b_2 = -10$$

Define-se a área apresentada na figura 16 como:

$$\int \mu_{(i)}$$

O método do centro de gravidade calcula o valor da saída u através da seguinte expressão:

$$u = \frac{\sum_i b_i \int \mu_{(i)}}{\sum_i \int \mu_{(i)}} \quad (22)$$

Que é a forma clássica para o cálculo do centróide de figuras geométricas. Três itens são importantes de ser ressaltados quando da utilização desse tipo de técnica:

1. Na prática, não se pode utilizar funções de pertinência que possuam área infinita, visto que isto resultaria em uma saída infinita, que não é desejada em um sistema de controle.
2. Deve-se tomar cuidado na definição das funções de pertinência para as entradas e saídas para que a soma no denominador da equação 22 não seja igual a zero para quaisquer entradas do controlador. Isto significa que o controlador deve ser capaz de inferir alguma conclusão para todas as possíveis situações que o controlador possa encontrar.
3. Enquanto a primeira vista possa parecer que o cálculo de $\int \mu_{(i)}$ é muito difícil, a utilização de funções simétricas, como no caso da triangular simétrica, facilita bastante esse cálculo. Neste caso, o sinal de controle pode ser obtido a partir do

suporte w da função de pertinência (largura da base) e do seu peso h , resultante do processo de inferência, utilizando a seguinte expressão:

$$u = w\left(h - \frac{h^2}{2}\right) \quad (23)$$

Sendo assim, o número de operações necessárias para calcular o sinal de controle torna-se bastante reduzido.

Um resumo gráfico de todo o processo de fuzzificação e defuzzificação apresentamos como exemplo encontra-se disposto na figura 17.

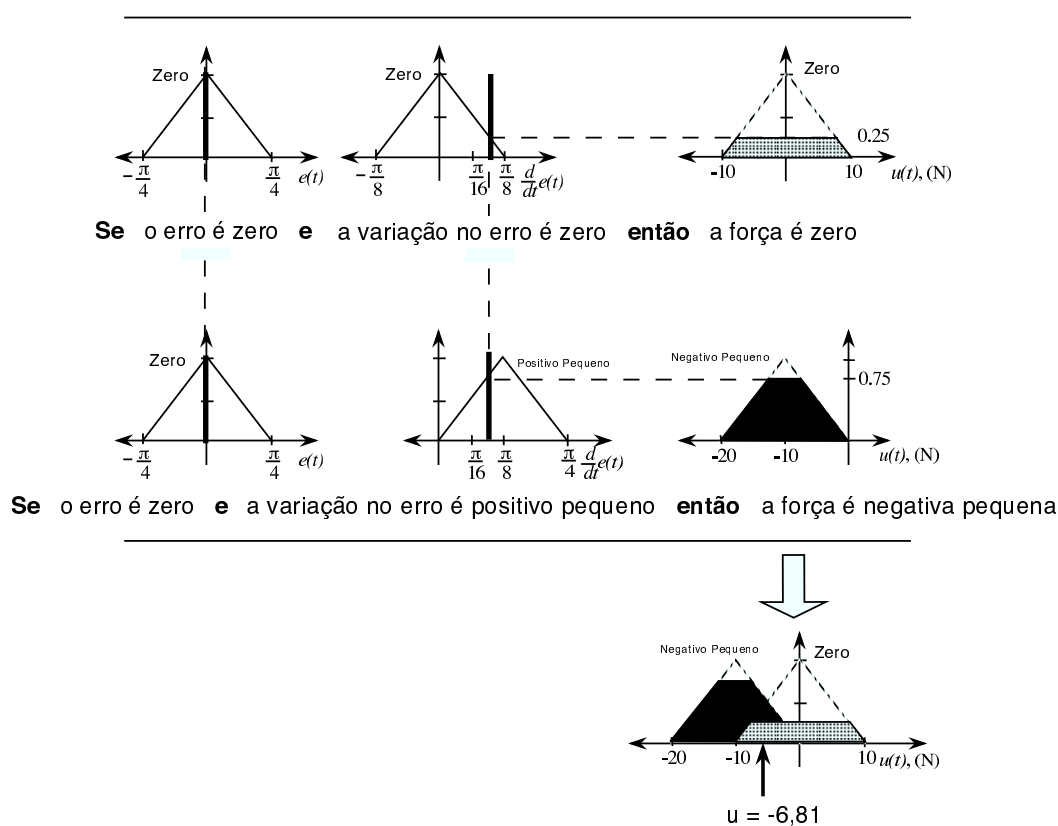


Figura 17: Resumo do processo de fuzzificação e defuzzificação.

Outra maneira de combinar recomendações

Uma outra maneira para calcular a composição das recomendações das regras, dentre diversas que podem ser utilizadas, é utilizar o operador de *produto* para representar a implicação. Na figura 18 observa-se as funções de pertinência para a saída com valores de “negativo pequeno” e “zero” em linhas pontilhadas. A função resultante da regra 1 é dada por:

$$\mu_{(1)}(u) = 0,25\mu_{zero}(u)$$

Esta função encontra sombreada na figura 18. Para a regra 2 obtém-se:

$$\mu_{(1)}(u) = 0,75\mu_{negativopequeno}(u)$$

Esta função está apresentada na figura 18 com um triângulo preto. Observa-se que o cálculo da saída do controlador utilizando como método de defuzzificação o COG fica bastante facilitado, visto que para o cálculo da área basta calcular $A = \frac{1}{2}wh$, que é a área do triângulo resultante. Utilizando o produto para representar a implicação obtém-se:

$$u = \frac{(0)(2,5) + (-10)(7,5)}{2,5 + 7,5} = -7,5$$

O que também faz sentido com relação ao comportamento do sistema e das variáveis de entrada.

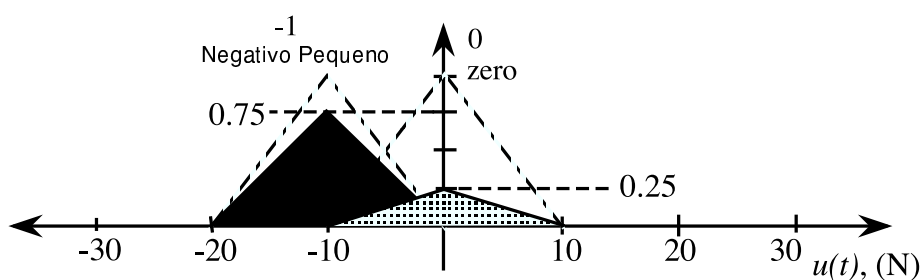


Figura 18: Conjuntos fuzzy utilizando o operador produto.

5 Estudo de caso

5.1 Introdução

Neste exemplo estudaremos o comportamento do sistema abaixo, onde sua entrada será a saída de uma válvula não linear dada pela característica abaixo:

A função de transferência do sistema:

$$G_0(s) = \frac{1}{(s+1)^3}$$

e a característica estática da válvula:

$$v = f(u) = u^4 \quad u \geq 0$$

O sistema pode ser descrito a partir do seguinte diagrama:

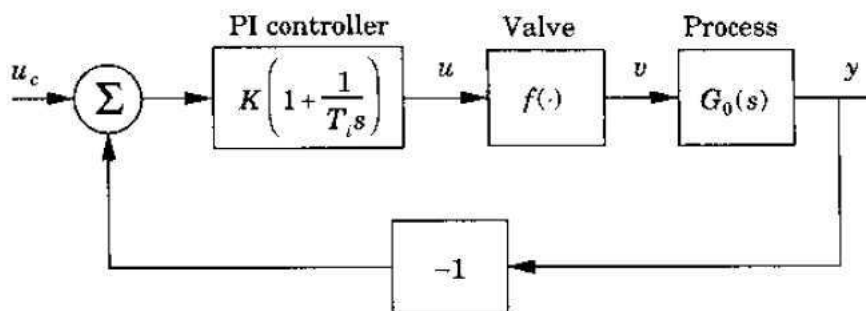


Figura 19: Processo [1]

5.2 Fundamentação teórica

Para o desenvolvimento do controlador fuzzy, estudou-se inicialmente quais seriam as entradas e as saídas deste bloco. Nesta etapa, definiu-se como entradas o erro percentual e o sinal de controle (setpoint) e como saída a variação percentual no sinal de controle.

Desta forma, tornou-se necessário a implementação de um bloco de pré-processamento, bem como outro bloco de pós-processamento.

O sistema resultante pode ser descrito da seguinte forma:

O bloco de pré-processamento se encarrega de calcular o erro percentual bem como sua variação temporal, logo após saturando tais valores em 1 ou -1 caso a resposta seja maior, em módulo. Tais cálculos são feitos da seguinte forma:

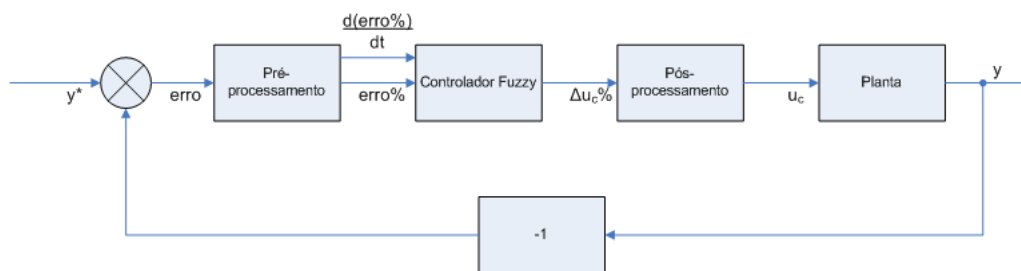


Figura 20: Controlador proposto

$$erro\% = \frac{erro}{y^*} \quad (24)$$

$$\Delta erro\% = \frac{d(erro\%)}{dt} \quad (25)$$

O bloco de pós-processamento, por sua vez, calcula o sinal de controle da seguinte forma:

$$u_c(t) = u_c(t-1) + \gamma \Delta u_c \quad (26)$$

Onde γ é um parâmetro que pode ser escolhido para variar a velocidade com que o controlador responderá. Neste caso, foi escolhido $\gamma = 0.7$, visto que desta forma o sistema respondeu satisfatoriamente para uma faixa muito grande de sinais de referência.

Após essa etapa, passou-se a etapa de definir o controlador fuzzy. Para tal, iniciou-se essa etapa, mapeando as entradas crisp em conjuntos fuzzy, da seguinte forma:

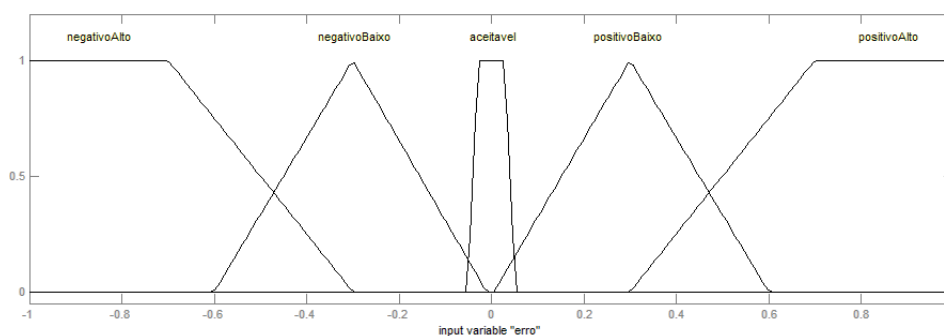


Figura 21: Conjuntos fuzzy para entradas $\Delta erro\%$ e y^*

Para a saída Δu_c , foram definidos os seguintes conjuntos:

A partir daí definiu-se a tabela que mapeia o comportamento da saída do controlador fuzzy baseado nas entradas desse. Tal tabela encontra-se a seguir:

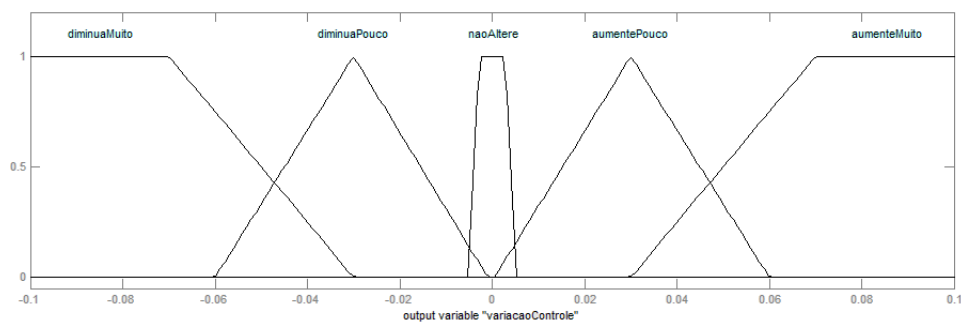


Figura 22: Conjuntos fuzzy para a saída Δu_c

erro% → $\Delta\text{erro}\% \downarrow$	NA	NB	A	PB	PA
NA	DM	DM	NA	NA	AP
NB	DM	DM	NA	AP	AM
A	DM	DP	NA	AP	AM
PB	DM	DP	NA	AM	AM
PA	DP	NA	NA	AM	AM

onde: NA → negativo alto
 NB → negativo baixo
 A → aceitável
 PB → positivo baixo
 PA → positivo alto
 AP → aumente pouco
 AM → aumente muito
 NA → não altere
 DP → diminua pouco
 DM → diminua muito

A partir da tabela 5.2, elaborou-se o conjunto de regras.

Para as operações de OU e Agregação foi utilizada a função máx e para as operações de AND e Implicação a função mín. No processo de Defuzzificação foi utilizada a operação de SOM (smallest of maximum).

5.3 Resultados

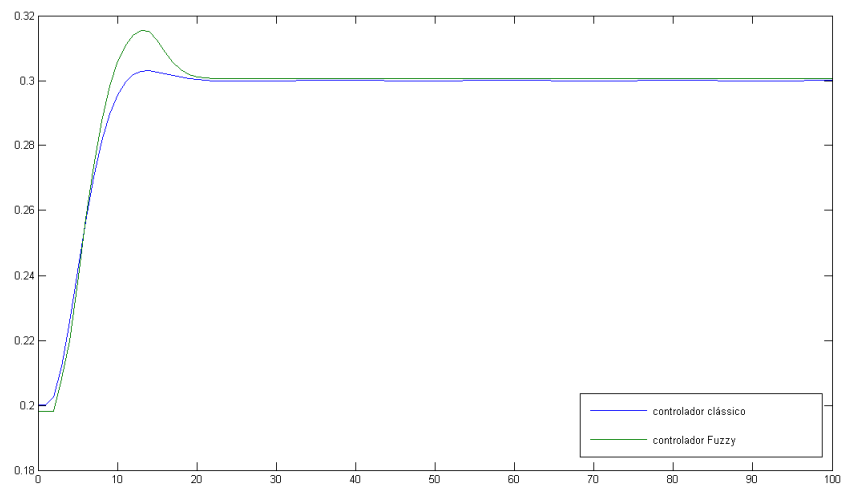


Figura 23: Saídas para os controladores Fuzzy e PI para o ponto de operação 0,2

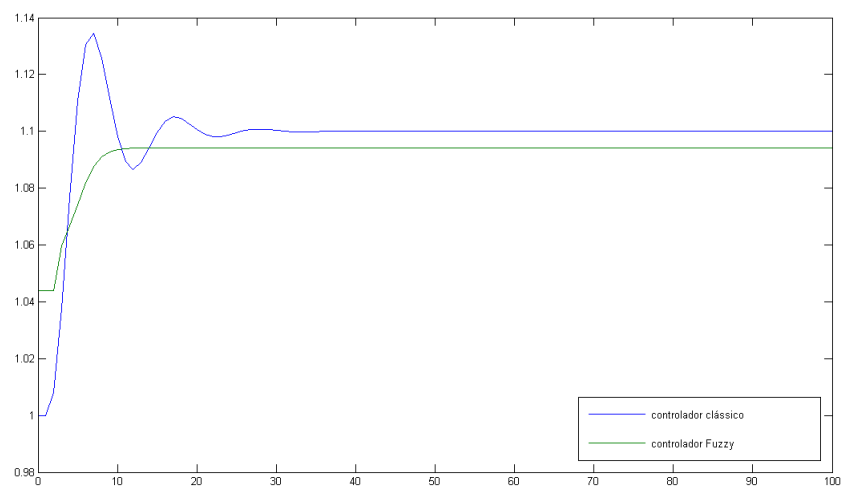


Figura 24: Saídas para os controladores Fuzzy e PI para o ponto de operação 1,0

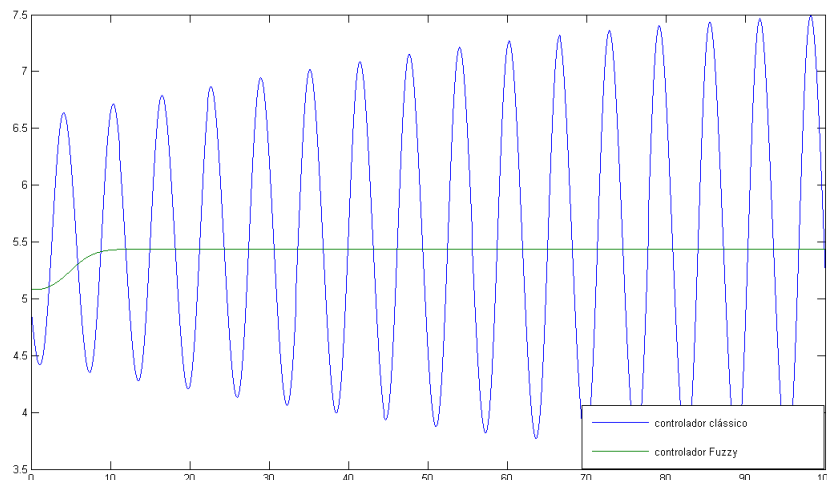


Figura 25: Saídas para os controladores Fuzzy e PI para o ponto de operação 5,0

5.4 Conclusões

Observa-se nos resultados, que o controlador Fuzzy obteve resultados satisfatórios e que sua grande vantagem quanto ao controlador PI clássico se deu pela robustez, visto que para diversos valores de referência (setpoints), a saída do sistema teve um comportamento satisfatório, o mesmo não ocorrendo para o controlador PI.

Observa-se que mesmo quando a saída do sistema se tornou instável com o controlador PI, a resposta do sistema com controlador fuzzy foi estável e bastante satisfatória.

6 Conclusão

Como foi visto, a utilização de lógica fuzzy em sistemas de controle torna bastante fácil o processo de desenvolvimento de um controlador, dada a abordagem inerentemente intuitiva deste processo.

Averiguou-se, neste trabalho, a influência de diversas escolhas durante a síntese do controlador, observando-se o comportamento do sistema quando dessas mudanças. Além disso, foi chamada a atenção para aspectos de implementação do controlador, de onde conclui-se que dependendo das escolhas que foram tomadas durante o processo de desenvolvimento, a realização das diversas etapas do controlador pode ser bastante simplificada.

Apesar do grande apelo intuitivo do processo de desenvolvimento e do desempenho que é, em muitos casos, melhor que o desempenho de controladores convencionais, chama-se a atenção que, dependendo do sistema, abordagens diferentes podem incorrer em melhores resultados.

O estudo de controladores fuzzy ainda encontra-se em constante desenvolvimento, principalmente no que concerne ao desenvolvimento de uma teoria que defina, analiticamente, critérios de estabilidade e desempenho. Várias teorias vêm sendo propostas e possivelmente, em pouco tempo, este tipo de controlador possa mais utilizado em mais processos.

Referências

- [1] K. J. A. e Bjorn Wittenmark, *Adaptive Control*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1994.
- [2] WikiPedia - Fuzzy Set - Acessado em 25/08/2007 - http://en.wikipedia.org/wiki/Fuzzy_set.
- [3] P. Viljamaa, “Fuzzy gain scheduling and tuning of multivariable fuzzy control - methods of fuzzy computing in control systems,” Ph.D. dissertation, Tampere University of Technology, 2002.
- [4] R. Jager, “Fuzzy logic in control,” Ph.D. dissertation, Technische Universiteit Delft., 1995.
- [5] K. M. P. e Stephen Yurkovich, *Fuzzy Control*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.