

# Laboratório 3 - Controle nebuloso

PTC 3418 – Laboratório de Automação  
1º semestre de 2020  
Bruno A. Angélico / Ricardo P. Marques

Laboratório de Automação e Controle  
Departamento de Engenharia de Telecomunicações e Controle  
Escola Politécnica da Universidade de São Paulo

# Objetivo

- Familiarização com a técnica de **controle nebuloso (fuzzy)**.

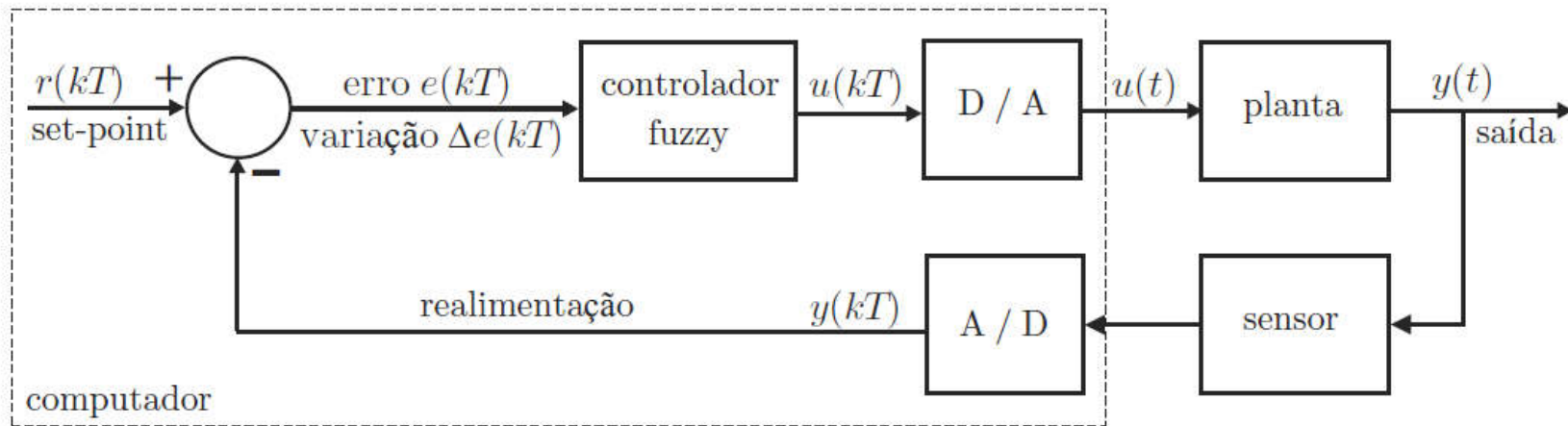


Figura 3.7: Diagrama de blocos de um sistema de controle em malha fechada.

# Conceitos básicos:

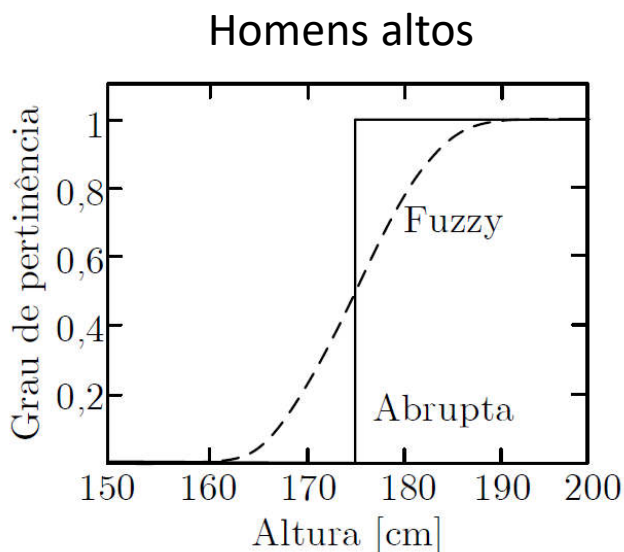
## Conjuntos *Fuzzy*

- Teoria apresentada inicialmente em 1965.
- Permite que informação imprecisa, qualitativa, possa ser expressa e manipulada através de um formalismo matemático.
- Exemplo:
  - o conjunto de homens altos;
  - o conjunto de carros caros.

# Conceitos básicos:

## Grau de pertinência

- Para definir, por exemplo, o conjunto de homens altos poderia se definir que uma altura  $x$  é alta se  $x \geq 176$  cm.
- Usando uma função com um **grau de pertinência** pode-se fazer a transição de “não alto” para “alto” de forma gradual.



# Conceitos básicos

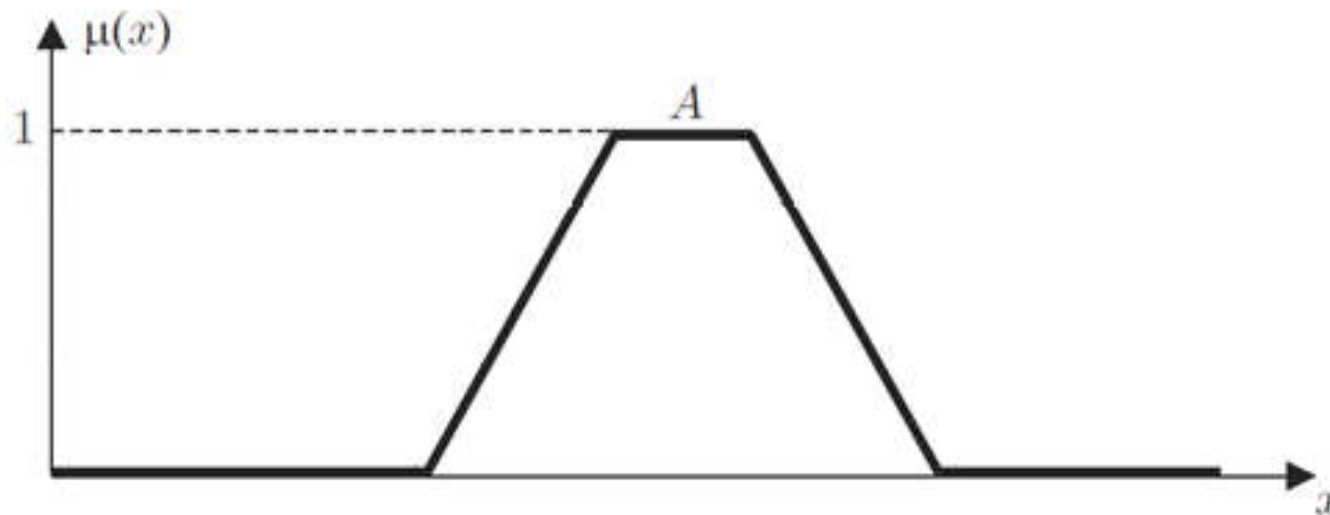
- **Definição:** dada uma coleção de objetos  $U$ , um conjunto fuzzy  $A$  é o conjunto do par ordenado

$$A = \{ [ x, \mu(x) ] \mid x \in U \}$$

- sendo que  $\mu(x)$  é chamada de **função de pertinência** do conjunto de todos os objetos  $x$  de  $U$ .

# Conceitos básicos: Função de pertinência

- A função de pertinência  $\mu(x)$ , de um conjunto *fuzzy* contínuo  $A$ , expressa o quanto um objeto  $x$  pertence ao conjunto  $A$



# Conceitos básicos: Funções de pertinência

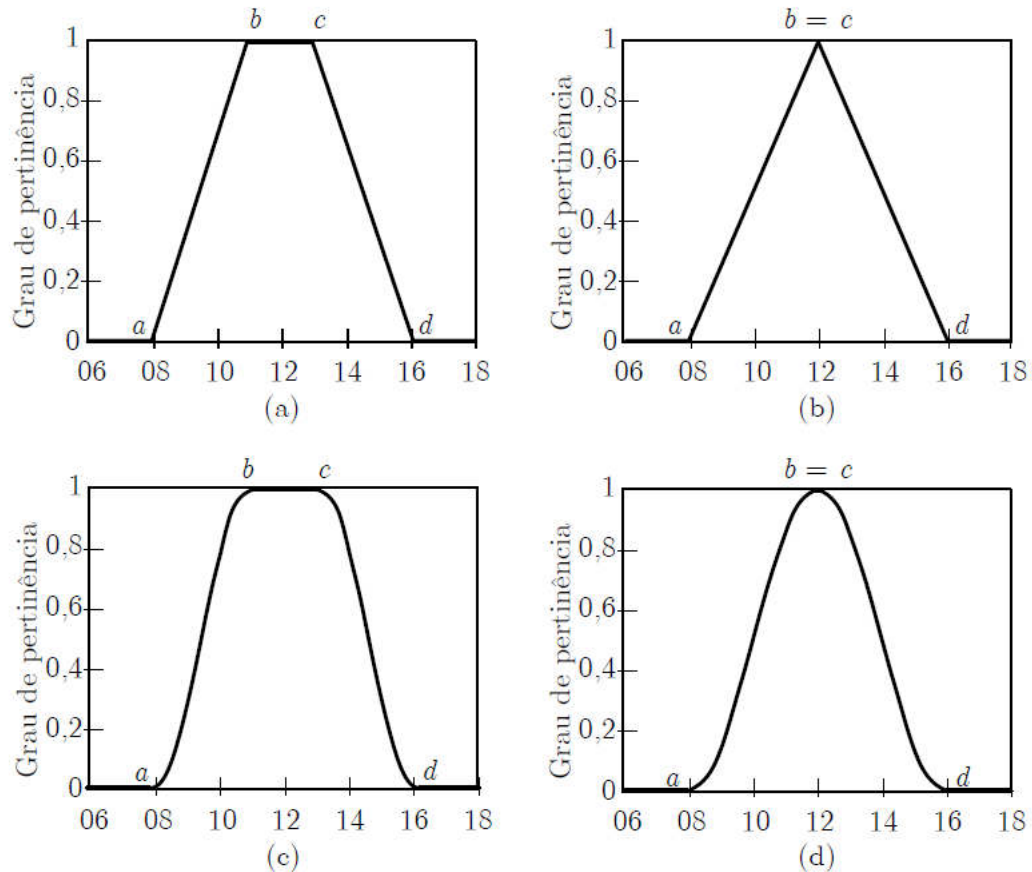


Figura 3.3: Funções de pertinência representando as horas por volta do meio-dia. (a) trapezoidal (b) triangular (c) trapezoidal suave (d) triangular suave.

# Conceitos básicos: Funções de pertinência

- Podemos utilizar uma associação de diferentes conjuntos  $A_i$ , com funções de pertinência  $\mu_i(x)$  para descrever os possíveis valores de um objeto  $x$

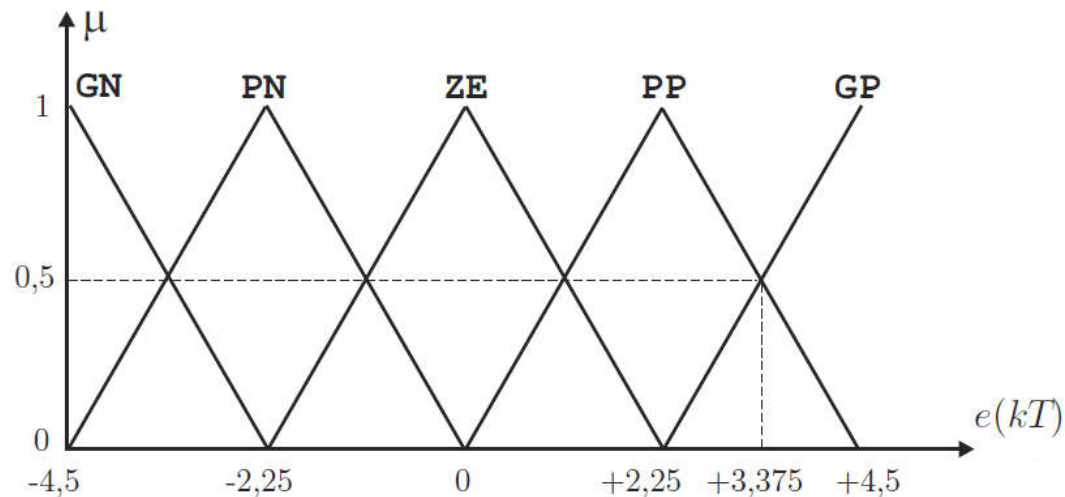


Figura 3.10: Funções de pertinência ou de associação.



# Conceitos básicos:

## Operações com conjuntos *fuzzy*

- Dois conjuntos fuzzy A e B são iguais se eles têm a mesma função de pertinência para todo x

$$A = B \equiv \mu_A(x) = \mu_B(x)$$

- Um conjunto fuzzy A está contido num conjunto fuzzy B, se a função de pertinência de A é menor ou igual a de B, ou seja

$$A \subseteq B \equiv \mu_A(x) \leq \mu_B(x)$$

# Conceitos básicos:

## Operações com conjuntos *fuzzy*

- A união clássica dos conjuntos  $X$  e  $Y$ , simbolizado por  $X \cup Y$ , é o conjunto de todos os objetos que são membros de  $X$  ou  $Y$ , ou ambos, isto é,

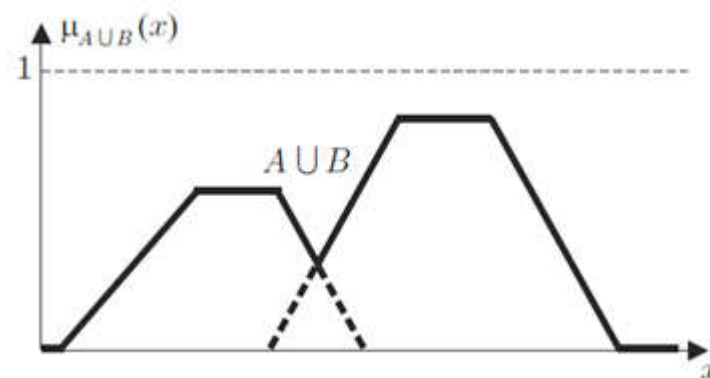
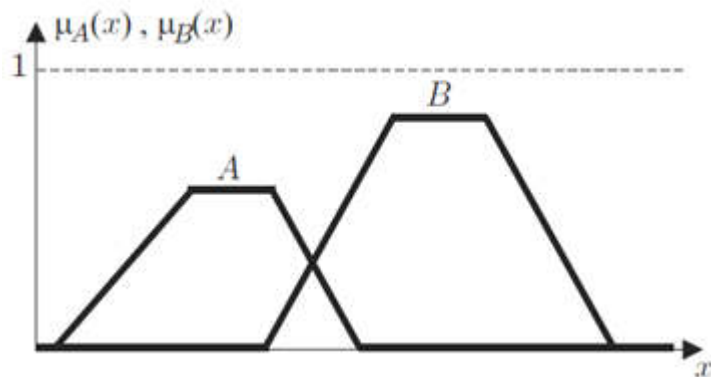
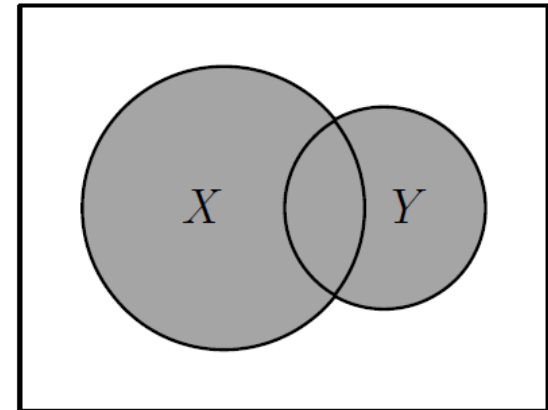
$$X \cup Y \equiv \{x \mid x \in X \text{ ou } x \in Y\}$$

- Por exemplo,

$$\{1, 2, 3\} \cup \{1, 3, 4\} = \{1, 2, 3, 4\}$$

- A união de dois conjuntos *fuzzy*  $A$  e  $B$  é

$$A \cup B \equiv \{[x, \mu_{A \cup B}(x)] \mid \mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]\}$$



# Conceitos básicos:

## Operações com conjuntos *fuzzy*

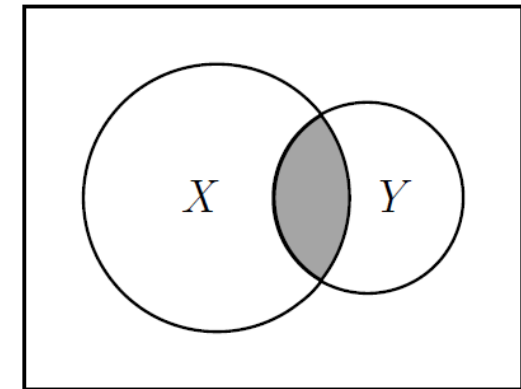
- A intersecção clássica dos conjuntos  $X$  e  $Y$ , simbolizado por  $X \cap Y$ , é o conjunto de todos os objetos que são membros de ambos  $X$  e  $Y$ , isto é,

$$X \cap Y \equiv \{x \mid x \in X \text{ e } x \in Y\}$$

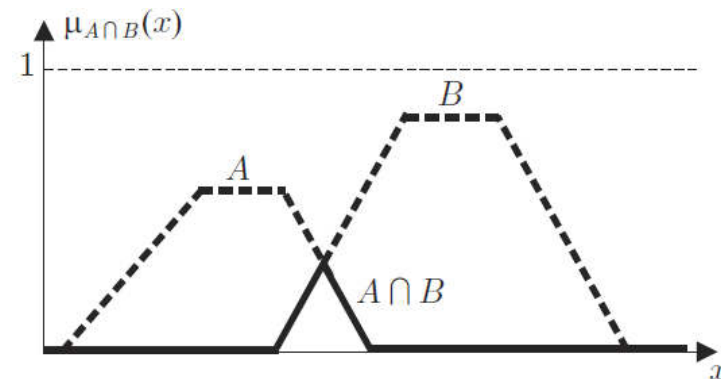
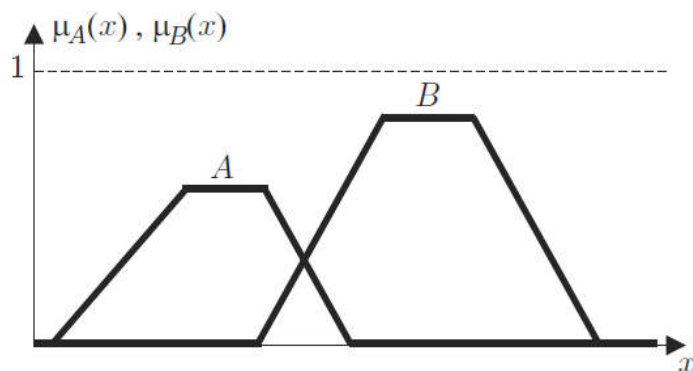
- Por exemplo,

$$\{1, 2, 3\} \cap \{1, 3, 4\} = \{1, 3\}$$

- A intersecção de dois conjuntos *fuzzy*  $A$  e  $B$  é



$$A \cap B \equiv \{ [x, \mu_{A \cap B}(x)] \mid \mu_{A \cap B}(x) = \min[ \mu_A(x), \mu_B(x) ] \}$$

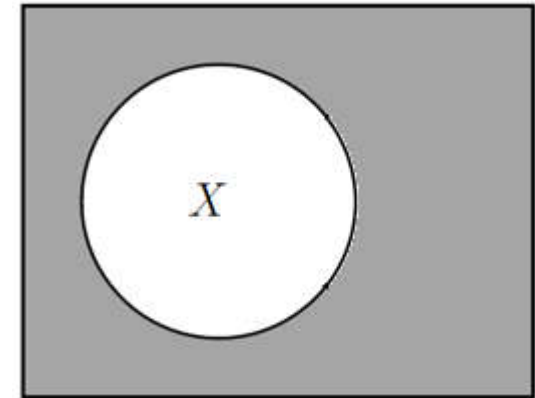


# Conceitos básicos:

## Operações com conjuntos *fuzzy*

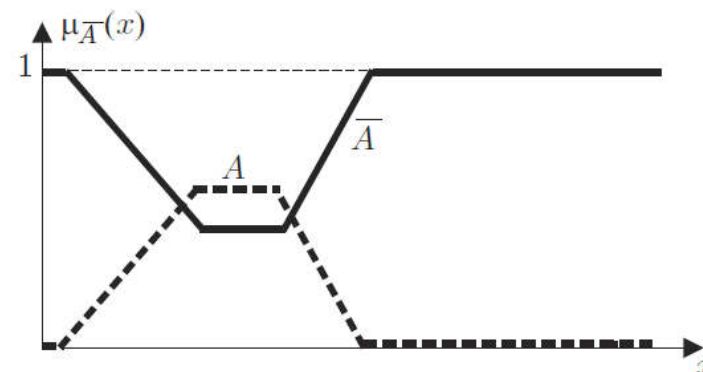
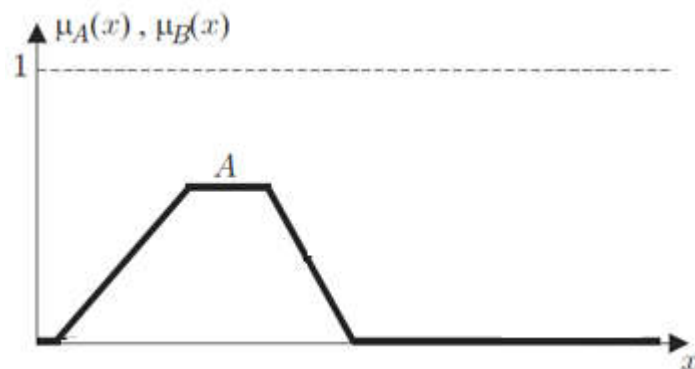
- O complemento clássico de um conjunto  $X$ , simbolizado por  $\overline{X}$ , é o conjunto dos membros  $x$  que não pertencem a  $X$ , isto é,

$$\overline{X} \equiv \{x \mid x \notin X\}$$



- O complemento de um conjunto *fuzzy*  $A$  é

$$\overline{A} \equiv \{[x, \mu_{\overline{A}}(x)] \mid \mu_{\overline{A}}(x) = 1 - \mu_A(x)\}$$



# Conceitos básicos:

## Operações com conjuntos *fuzzy*

- Exemplo: família de 4 pessoas quer comprar uma casa
  - O conjunto de casas para compra possui número de quartos:

$$U = \{1; 2; 3; 4; 5; 6; 7; 8; 9; 10\}$$

- O conjunto  $C$  representa conforto de uma casa, com grau de pertinência:

$$\mu_C = \{0,2; 0,5; 0,8; 1; 0,7; 0,3; 0; 0; 0; 0\}$$

- O conjunto  $G$  representa quão grande é uma casa, com grau de pertinência:

$$\mu_G = \{0; 0; 0,2; 0,4; 0,6; 0,8; 1; 1; 1; 1\}$$

- A intersecção dos conjuntos conforto e grande é tal que:

$$C \cap G = \min(\mu_C, \mu_G) = \{0; 0; 0,2; 0,4; 0,6; 0,3; 0; 0; 0; 0\}$$

Melhor solução: casa com 5 quartos (grau de pertinência 0,6)

# Conceitos básicos:

## Operações com conjuntos *fuzzy*

- Continuação do exemplo:
  - Se for desejável conforto **ou** casa grande, toma-se a união dos conjuntos:

$$C \cup G = \max(\mu_C, \mu_G) = \{0,2; 0,5; 0,8; \mathbf{1}; 0,7; 0,8; \mathbf{1}; \mathbf{1}; \mathbf{1}; \mathbf{1}\}$$

- Se os filhos estiverem para se casar e forem mudar de casa em alguns meses então é necessário comprar uma casa com conforto e não grande:

$$C \cap \bar{G} = \min(\mu_C, 1 - \mu_G) = \{0,2; 0,5; \mathbf{0,8}; 0,6; 0,4; 0,2; 0; 0; 0; 0\}$$

Melhor solução: casa com 3 quartos (grau de pertinência 0,8)

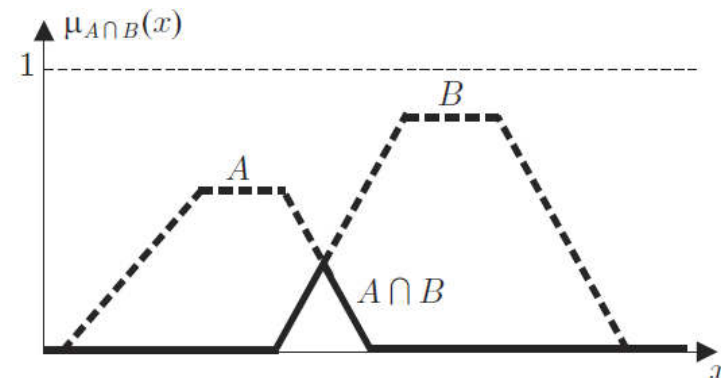
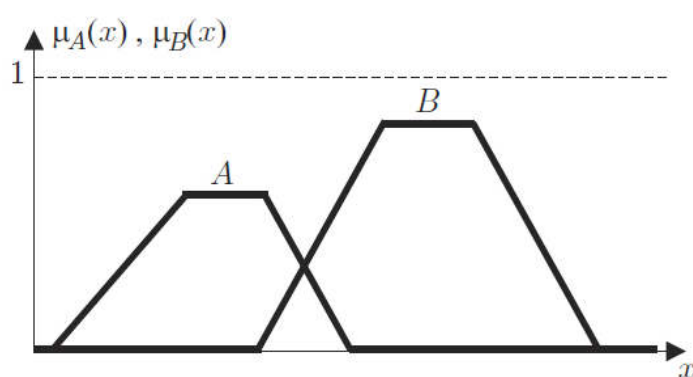
# Conceitos básicos: Regra de Mamdani

- Frequentemente utilizada em controle *fuzzy*.
- Sejam A e B dois conjuntos *fuzzy* definidos em x e y, respectivamente. Então

$$\mu(x, y) = \min[\mu_A(x), \mu_B(y)]$$

- Ou seja, a regra de Mamdani equivale a uma intersecção

$$A \cap B \equiv \{ [ x, \mu_{A \cap B}(x) ] \mid \mu_{A \cap B}(x) = \min[ \mu_A(x), \mu_B(x) ] \}$$



# Conceitos básicos: Variáveis linguísticas

- Uma variável linguística possui palavras ou sentenças como valores.
- Considere o conjunto de idades:

$U(\text{idade}) = \{\text{jovem, muito jovem, não muito jovem, mais ou menos velho, velho}\}$

$$U = [ 0 \quad 20 \quad 40 \quad 60 \quad 80 ]$$

- Com funções de pertinência:

$$\mu_{\text{jovem}} = [ 1 \quad 0,6 \quad 0,1 \quad 0 \quad 0 ] \quad \text{e} \quad \mu_{\text{velho}} = [ 0 \quad 0 \quad 0,1 \quad 0,6 \quad 1 ]$$

- A função de pertinência do conjunto mais ou menos velho é

$$\mu_{\text{mais ou menos velho}} = \sqrt{\mu_{\text{velho}}} = [ 0 \quad 0 \quad 0,32 \quad 0,77 \quad 1 ]$$

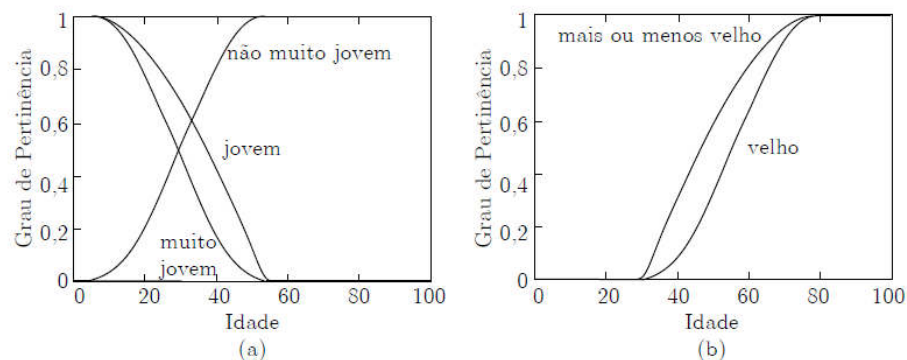


Figura 3.6: Funções de pertinência modificadas.



# Controle *Fuzzy*

- **Controle nebuloso** significa controle por regras (nebulosas).

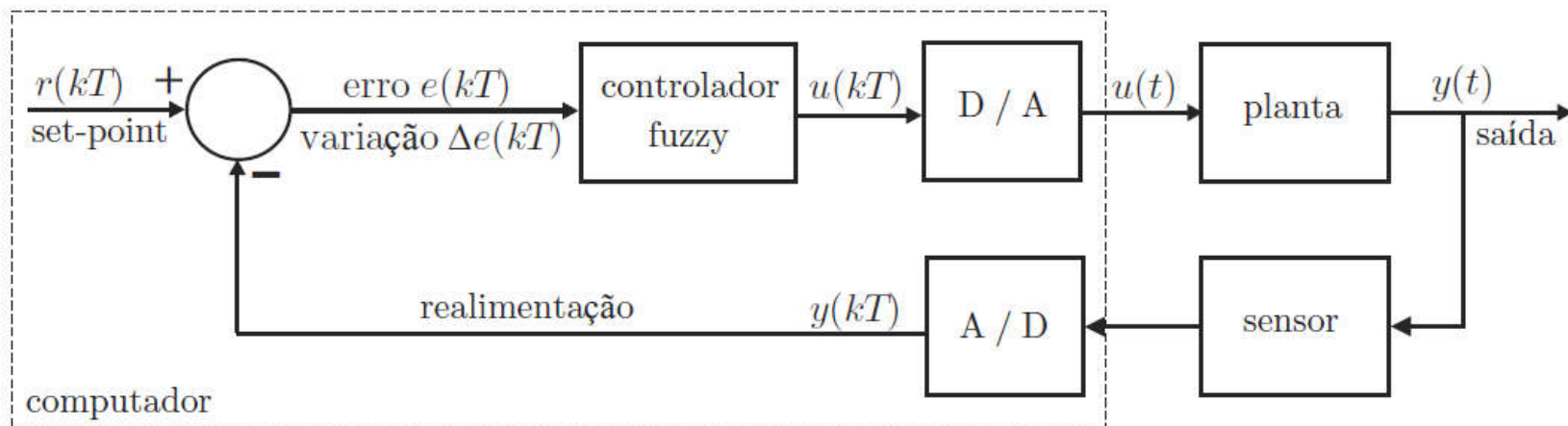


Figura 3.7: Diagrama de blocos de um sistema de controle em malha fechada.

# Tipos de Controladores *Fuzzy*

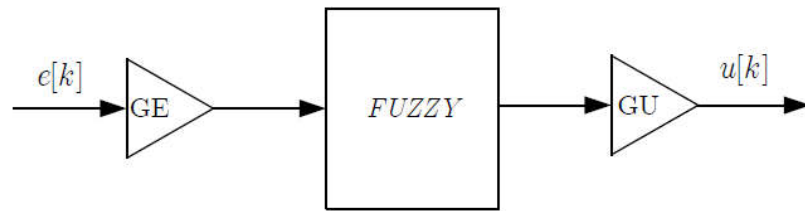


Figura 3.15: Digrama de blocos do controlador fuzzy P.

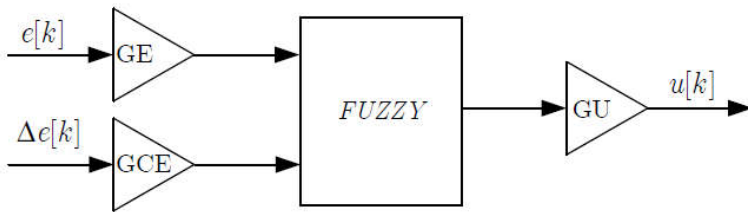


Figura 3.16: Digrama de blocos do controlador fuzzy PD.

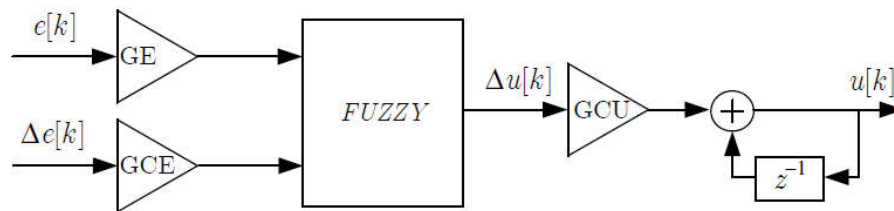


Figura 3.17: Digrama de blocos do controlador fuzzy incremental (PI).

# Estrutura de um controlador *fuzzy*

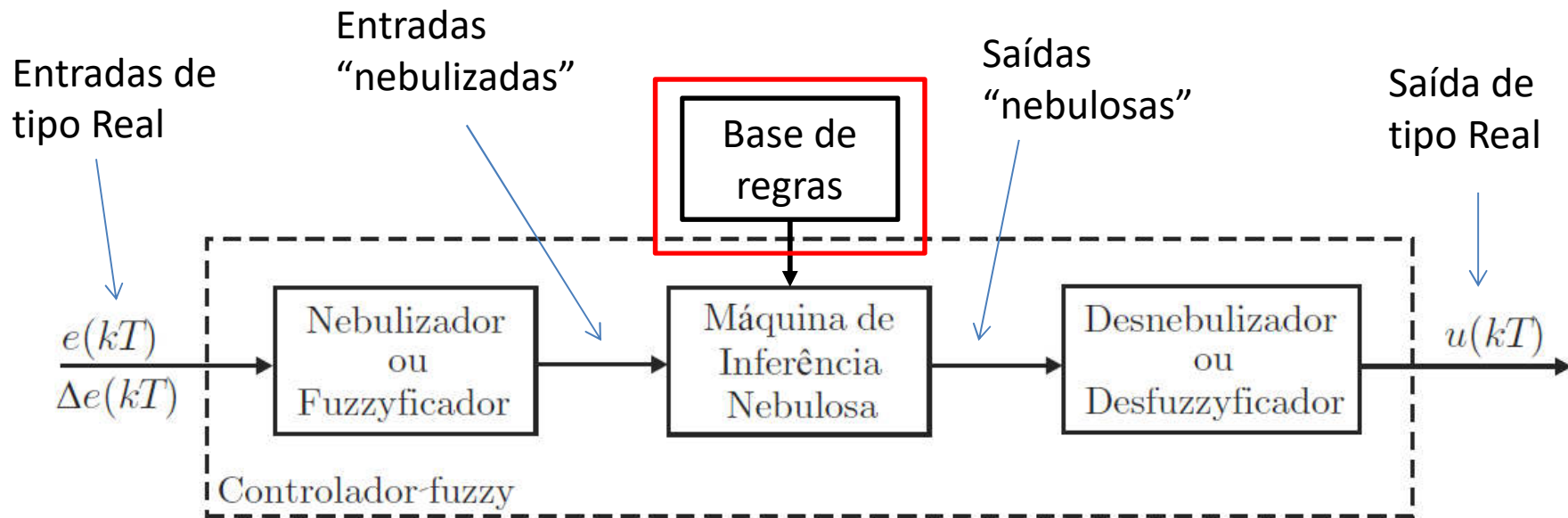


Figura 3.8: Estrutura de um controlador fuzzy.

# Base de regras de um Controle *Fuzzy*

- Um controle *fuzzy* pode possuir regras empíricas, baseadas principalmente em plantas controladas por operadores.
- As regras são do tipo **se-então** com a premissa do lado do **se** e a conclusão do lado do **então**.
- Exemplo:
  - **Se** o erro é N **e** a variação do erro é N **então** o controle é GN.
  - **Se** o erro é N **e** a variação do erro é Z **então** o controle é MN.
- Os termos linguísticos N (Negativo) ou Z (Zero) são premissas e GN (Grande Negativo) ou MN (Médio Negativo) são conclusões.

# Base de regras de um Controle *Fuzzy*

- As regras do controlador são baseadas na linguagem natural e podem ser entendidas por operadores ou por equipes de manutenção.
- Alguns critérios para a elaboração das regras são apresentados a seguir:
  - *Experiência de funcionários e conhecimento de engenheiros*: elaborar um questionário detalhado a ser preenchido por funcionários ou engenheiros com o objetivo de extrair regras de controle *fuzzy*.
  - *Baseado nas ações de controle do operador*: observando as ações de controle do operador pode-se encontrar relações do tipo entrada-saída que podem ser utilizadas na elaboração das regras *fuzzy*.
  - *Examinando manuais* também podem ser obtidas informações úteis para a elaboração das regras.
  - *Baseado no modelo da planta*: uma regra linguística pode ser vista como o inverso da planta. Assim, é possível obter regras *fuzzy* invertendo o modelo da planta. Este método é restrito a sistemas de baixa ordem.

# Base de regras de um Controle *Fuzzy*

- **regra 1:** se  $e(kT)$  é ZE e  $\Delta e(kT)$  é PP **então**  $u(kT)$  deve ser PP;  
**ou**
- **regra 2:** se  $e(kT)$  é ZE e  $\Delta e(kT)$  é ZE **então**  $u(kT)$  deve ser ZE;  
**ou**
- **regra 3:** se  $e(kT)$  é PN e  $\Delta e(kT)$  é PN **então**  $u(kT)$  deve ser PN;  
**ou**
- **regra 4:** se  $e(kT)$  é GP e  $\Delta e(kT)$  é ZE **então**  $u(kT)$  deve ser GP.

# Base de regras de um Controle *Fuzzy*

Tabela 3.1: Regras básicas de controle.

Erro	Variação do Erro	Sinal de Controle
GN	GN	GN
GN	PN	GN
GN	ZE	GN
GN	PP	MN
GN	GP	MN
PN	GN	MN
PN	PN	MN
PN	ZE	ZE
PN	PP	ZE
PN	GP	MP
ZE	GN	MN
ZE	PN	MN
ZE	ZE	ZE
ZE	PP	MP
ZE	GP	MP
PP	GN	MN
PP	PN	ZE
PP	ZE	ZE
PP	PP	MP
PP	GP	MP
GP	GN	MP
GP	PN	MP
GP	ZE	GP
GP	PP	GP
GP	GP	GP



Tabela 3.2: Forma compacta de visualização das regras básicas de controle.

		Erro				
		GN	PN	ZE	PP	GP
Variação do Erro	GN	GN	MN	MN	MN	MP
	PN	GN	MN	MN	ZE	MP
	ZE	GN	ZE	ZE	ZE	GP
	PP	MN	ZE	MP	MP	GP
	GP	MN	MP	MP	MP	GP

# Estrutura de um controlador *fuzzy*

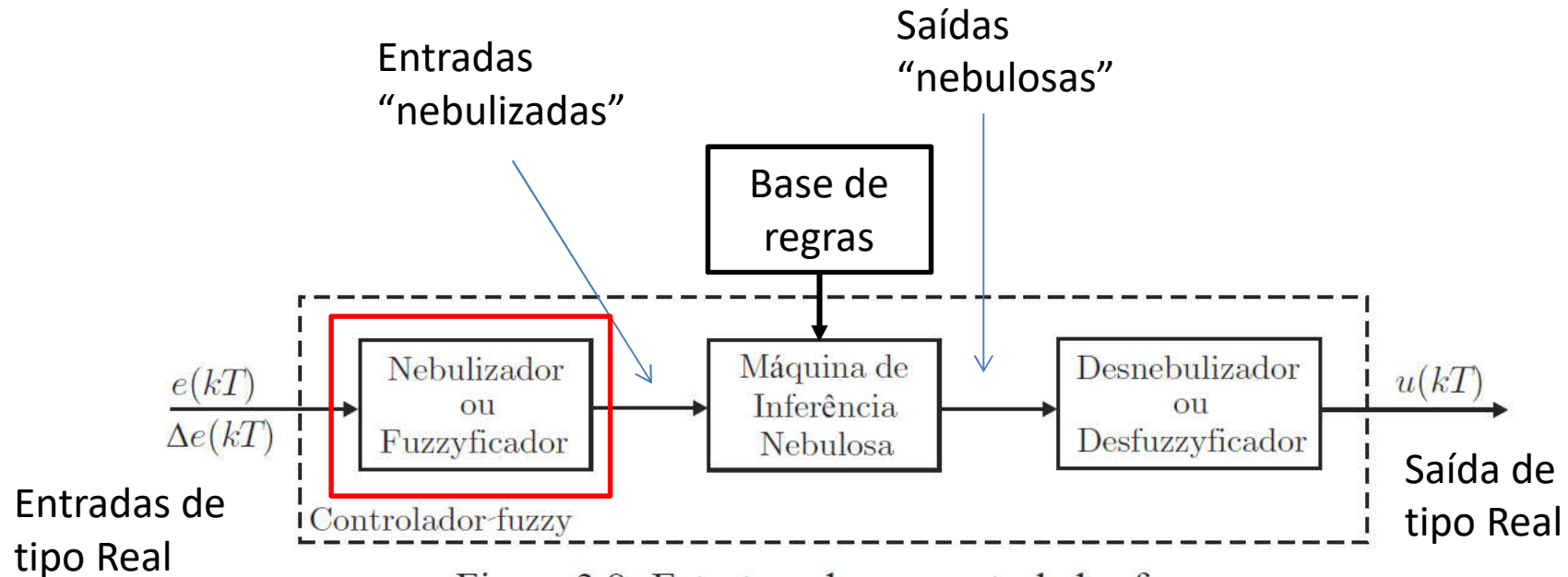


Figura 3.8: Estrutura de um controlador fuzzy.



# Nebulizador

- mudar as escalas das variáveis de entrada do controlador, normalizando-as de maneira a pertencerem aos universos de discurso dos conjuntos nebulosos correspondentes;
- converter esses valores numéricos em valores linguísticos associados aos conjuntos nebulosos.

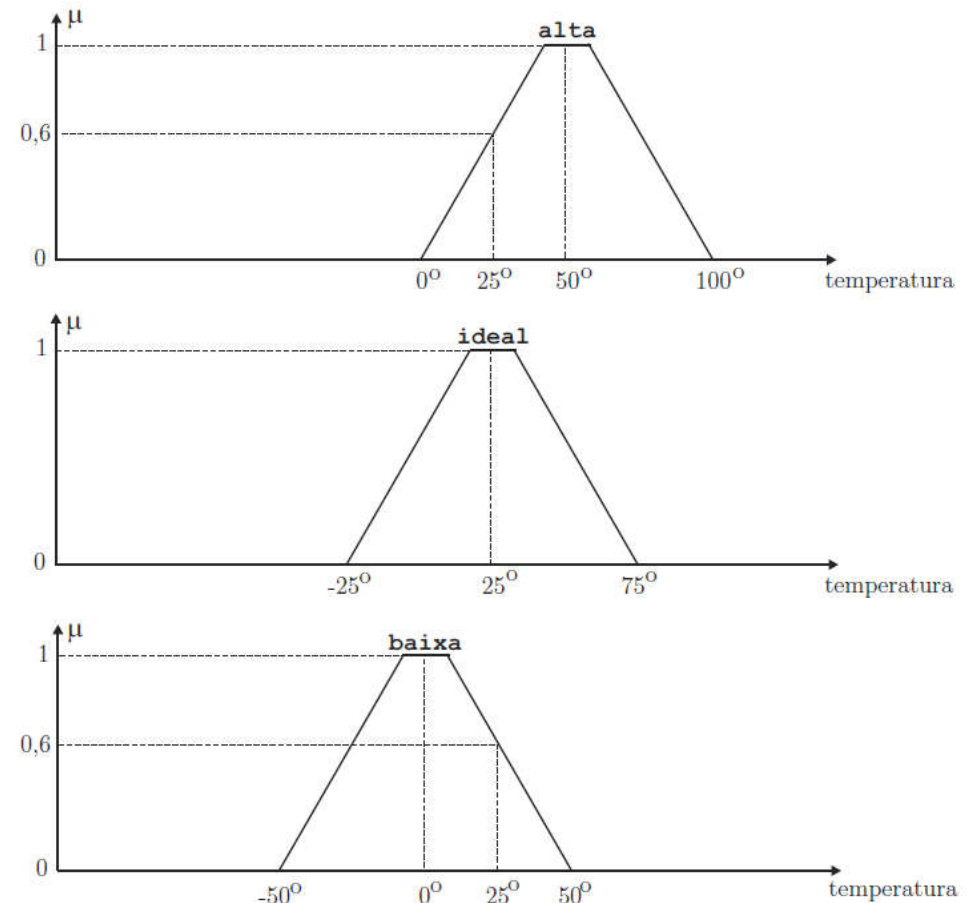


Figura 3.9: Funções de pertinência.

# Exemplo de Nebulizador para o servomecanismo

- Num servomecanismo, o ângulo de saída (posição medida) pode ser uma tensão variando, por exemplo, entre  $-4,5V$  a  $+4,5V$ .
- O erro e a variação do erro podem ser quantificados, por exemplo, da seguinte forma:
  - GP = Grande Positivo;
  - PP = Pequeno Positivo;
  - ZE = Zero;
  - PN = Pequeno Negativo;
  - GN = Grande Negativo.
- Para cada um dos conjuntos nebulosos GP, PP, ZE, PN, GN deve-se associar funções de pertinência  $\mu$ . A forma da função é arbitrária. Por simplicidade, as mais utilizadas são as funções trapezoidais e triangulares.

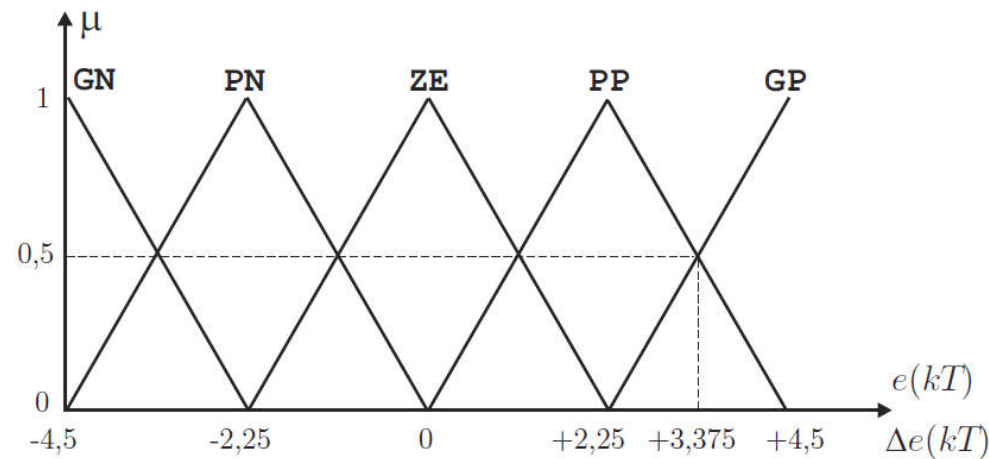


Figura 3.10: Funções de pertinência ou de associação.

# Estrutura de um controlador *fuzzy*

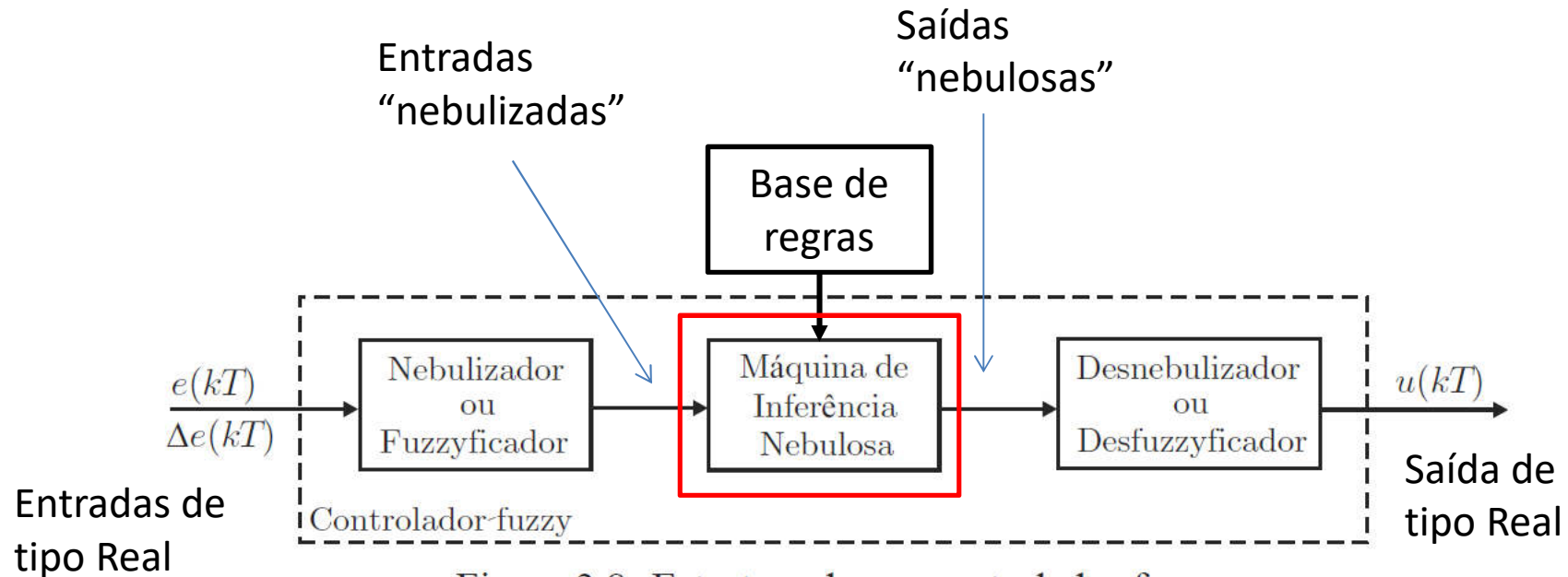


Figura 3.8: Estrutura de um controlador fuzzy.

# Máquina de inferência nebulosa

## Conjunto nebuloso obtido a partir das regras de um Controle *Fuzzy*

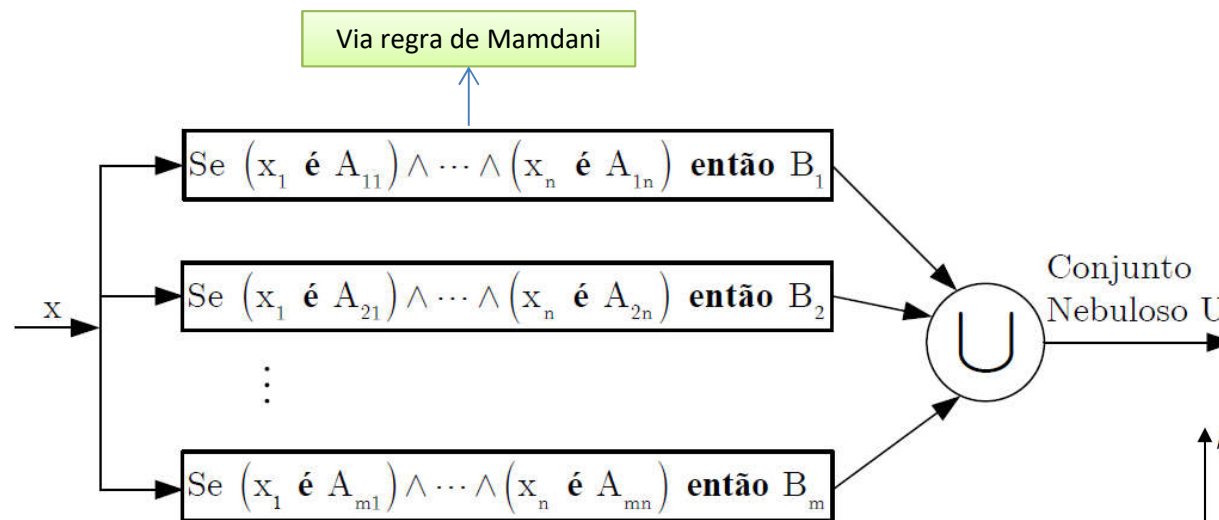


Figura 3.11: Sistema nebuloso aditivo.

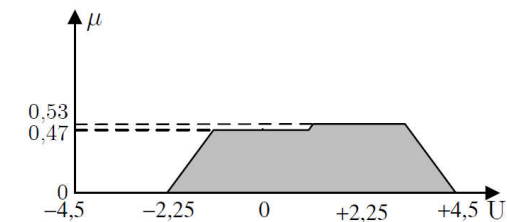
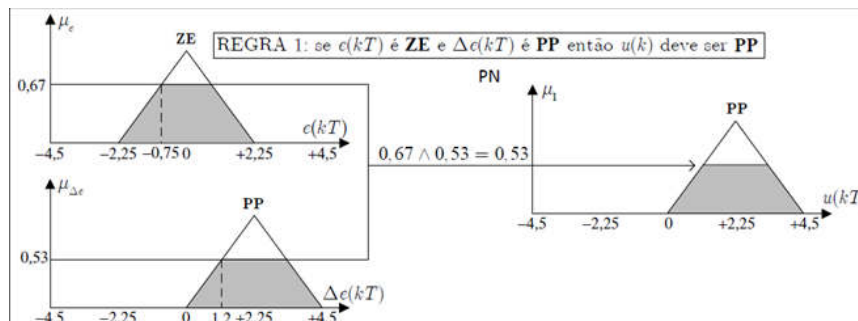
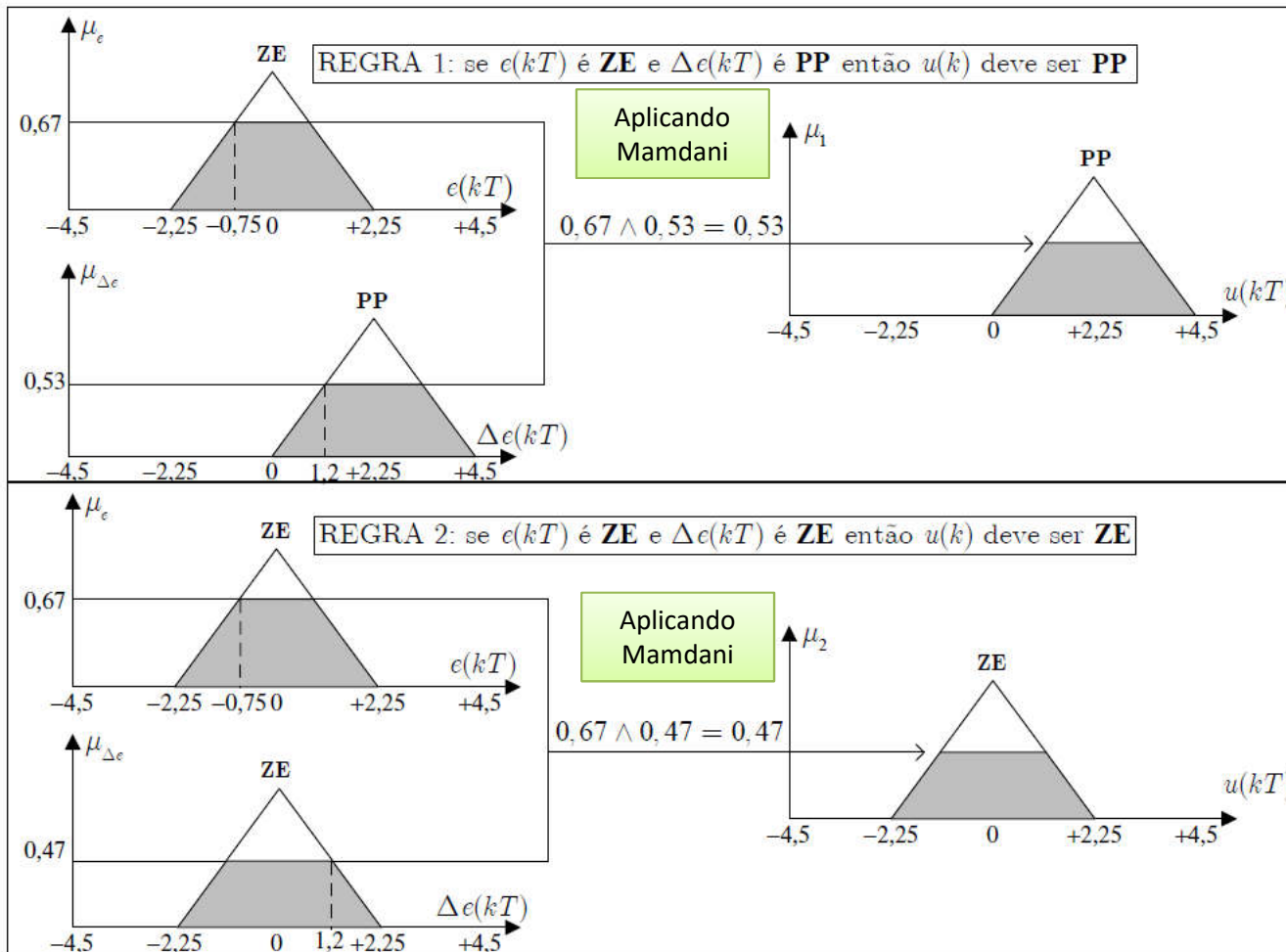
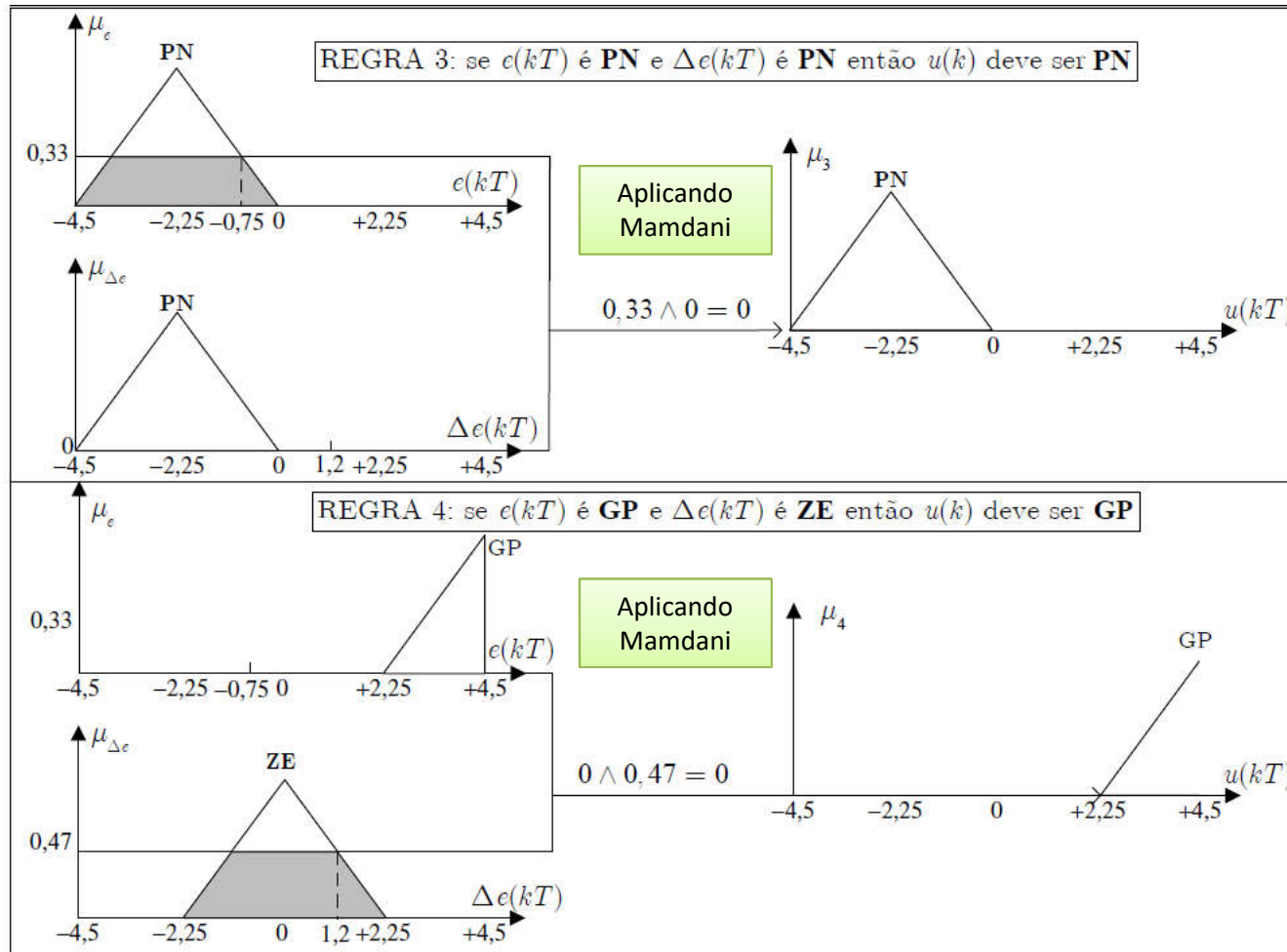


Figura 3.17: Conjunto fuzzy resultante.

# Exemplo de Máquina de Inferência Nebulosa do servomecanismo



# Exemplo de Máquina de Inferência Nebulosa do servomecanismo



# Exemplo de Máquina de Inferência Nebulosa do servomecanismo

- Dos resultados da aplicação das regras:

regra 1:  $\mu_1 = \min[\mu_e, \mu_{\Delta e}] = \min[0,67; 0,53] = 0,53.$

regra 2:  $\mu_2 = \min[\mu_e, \mu_{\Delta e}] = \min[0,67; 0,47] = 0,47.$

regra 3:  $\mu_3 = \min[\mu_e, \mu_{\Delta e}] = \min[0,33; 0] = 0.$

regra 4:  $\mu_4 = \min[\mu_e, \mu_{\Delta e}] = \min[0; 0,47] = 0.$

- obtemos um conjunto nebuloso de saída:

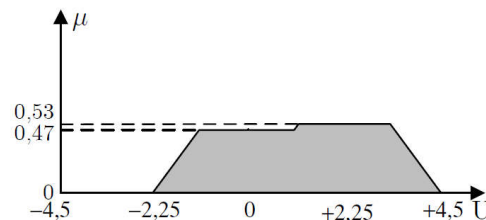


Figura 3.17: Conjunto fuzzy resultante.

# Estrutura de um controlador *fuzzy*

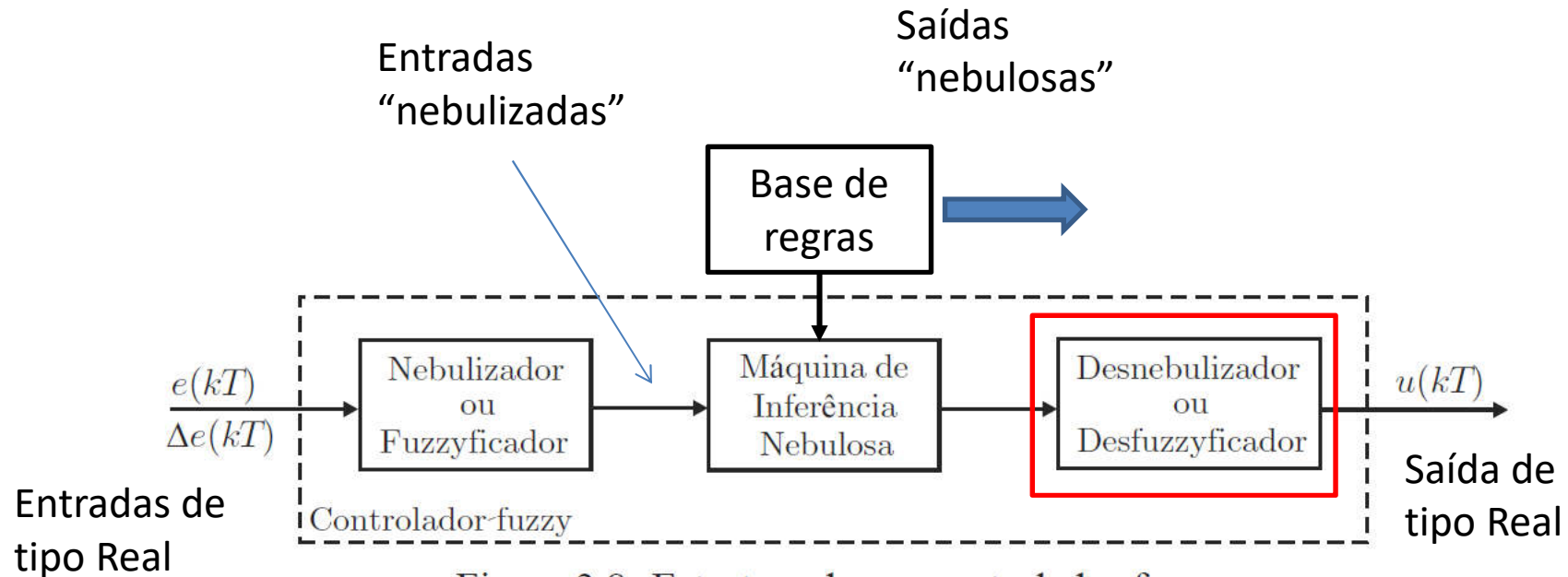


Figura 3.8: Estrutura de um controlador fuzzy.



# Desnebulizador

- É responsável por:
  - converter os valores nebulizados em variáveis numéricas
  - encontrar o esforço de controle a ser aplicado ao processo

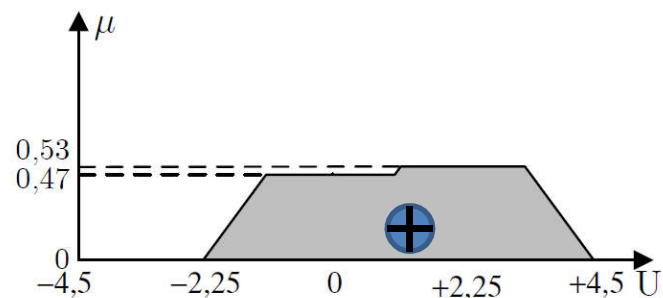


Figura 3.17: Conjunto fuzzy resultante.

# Desnebulizador

- Diferentes métodos possíveis. Dentre eles, citamos dois:
  - Médias ponderadas
  - Centro de gravidade

- Médias ponderadas

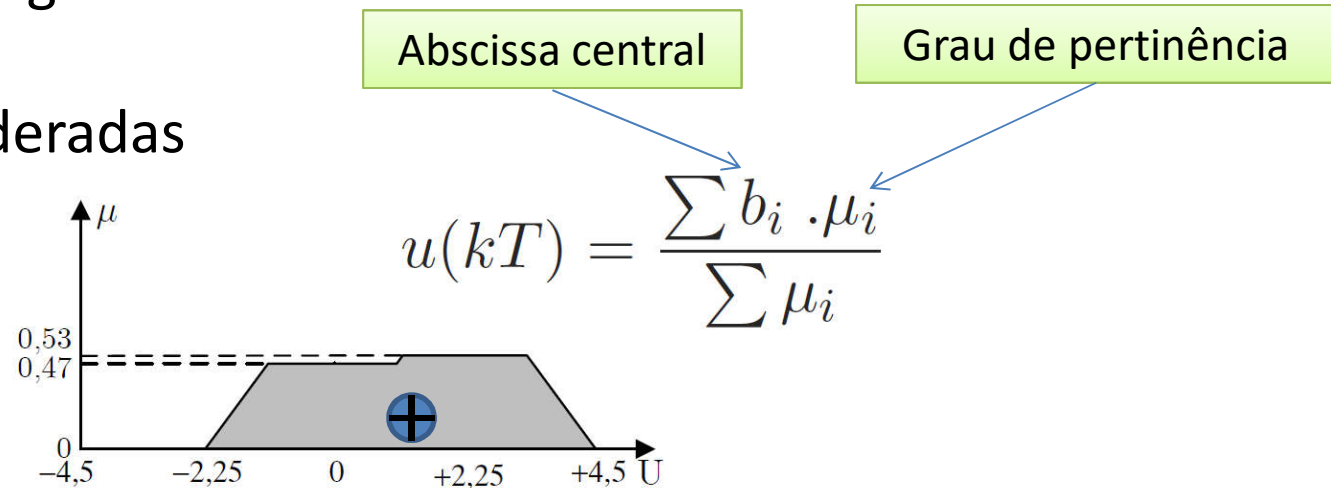
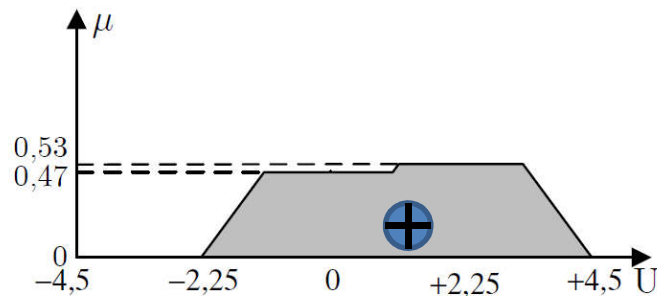


Figura 3.17: Conjunto fuzzy resultante.

- 1ª etapa: dado um conjunto de func. de pertinência, encontrar a abscissa central
- 2ª etapa: calcular a média ponderada, que corresponde ao esforço de controle

# Exemplo de Desnebulizador do servomecanismo

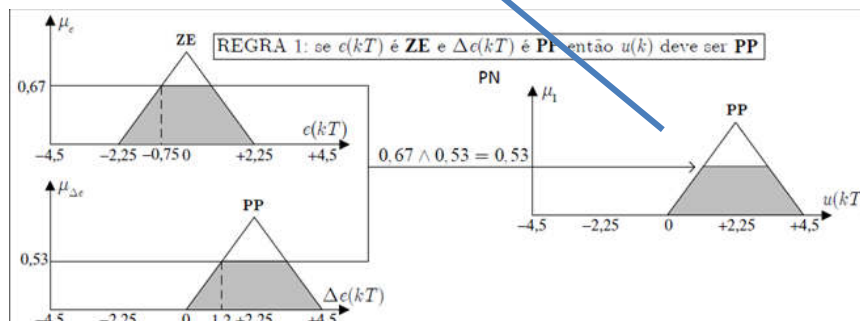
- Médias ponderadas



$$u(kT) = \frac{\sum b_i \cdot \mu_i}{\sum \mu_i}$$

Figura 3.17: Conjunto fuzzy resultante.

$$u(kT) = \frac{0,53 \cdot (+2,25) + 0,47 \cdot (0) + 0 \cdot (+2,25) + 0 \cdot (+4,5)}{0,53 + 0,47 + 0 + 0} \approx 0,896$$



# Desnebulizador

- Centro de gravidade

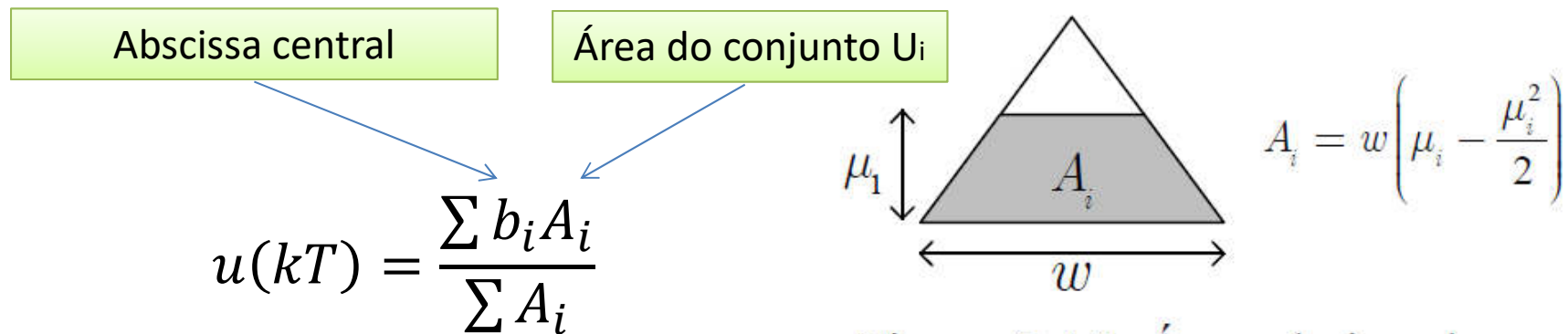


Figura 3.14: Área abaixo de  $\mu_i$ .

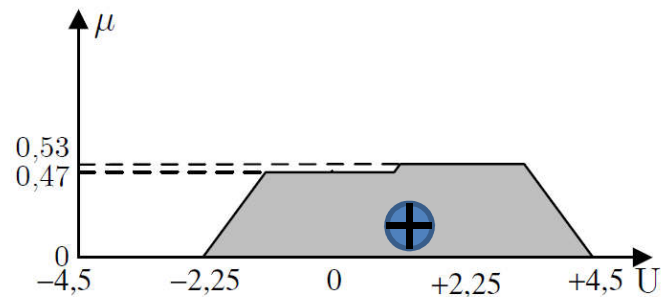


Figura 3.17: Conjunto fuzzy resultante.

# Especificações de projeto

- Efetuar o controle digital de posição do servomecanismo, utilizando controle nebuloso PD.

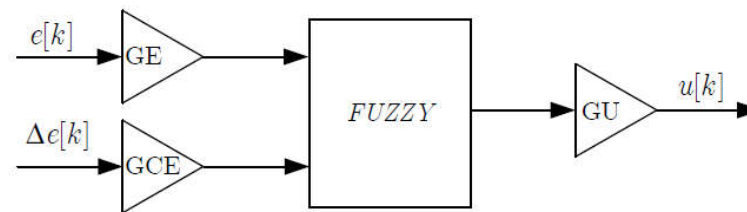


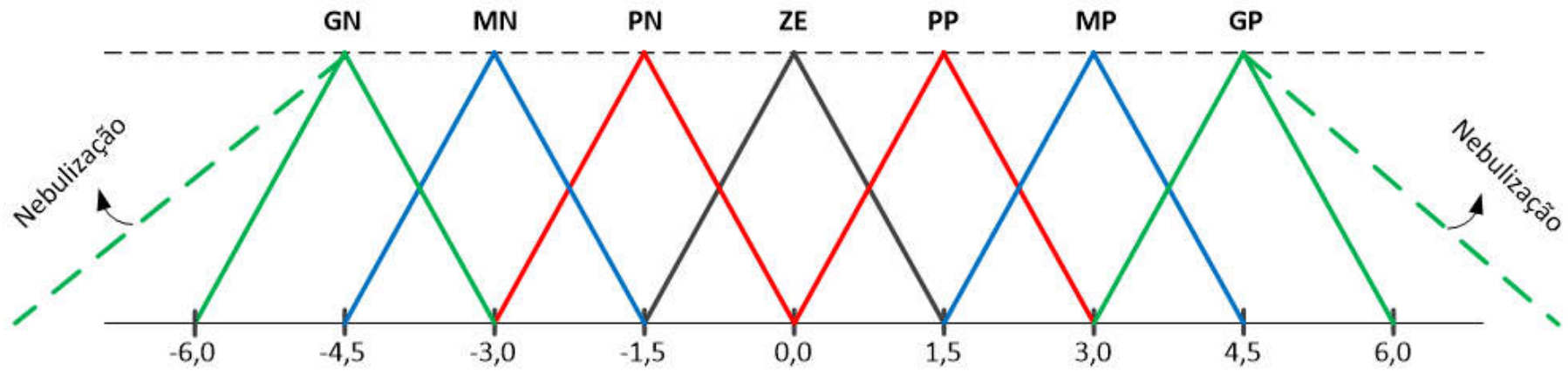
Figura 3.16: Digrama de blocos do controlador fuzzy PD.

# Base de Regras para a Experiência

Tabela 4.3: Regras de controle para o servomecanismo

		Erro						
		GN	MN	PN	ZE	PP	MP	GP
Variação do Erro	GN							
	MN			MN		PN	PP	
	PN	GN			PN	ZE	MP	
	ZE			MN	ZE		GP	
	PP	MN		ZE	PP	MP		
	MP				MP			
	GP	PN		PP	MP			

# Conjuntos Fuzzy



# Sugestões

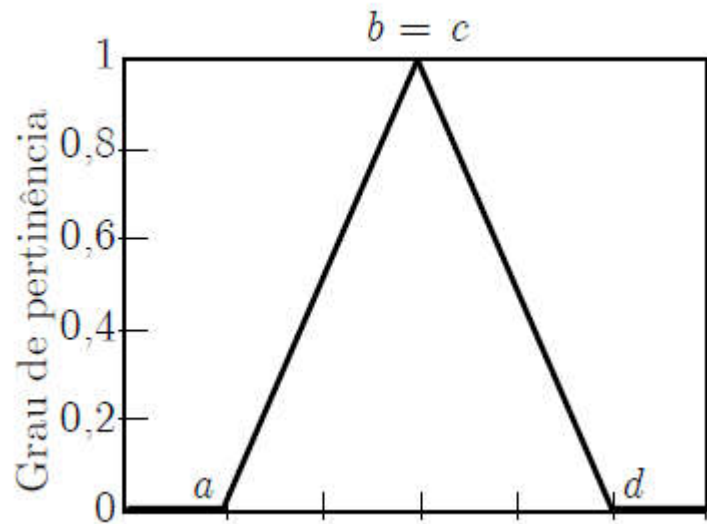
- Sugestão 1: crie uma função para retornar o grau de pertinência considerando como entrada os pontos  $(a(i), b(i), d(i))$  da  $i$ -ésima função triangular e o valor do erro (ou da variação do erro) e como saída o grau de pertinência do valor nesse conjunto:

```
function mu =  
triang_membership_function(a(i), b(i), d(i), x)
```



# Sugestões

- A função deve implementar o seguinte:



$$\mu(x) = \begin{cases} 0 & , x < a \\ \frac{x-a}{b-a} & , a \leq x < b \\ \frac{d-x}{d-b} & , b \leq x < d \\ 0 & , d \leq x \end{cases}$$

# Sugestões

- Utilize um laço `for` de 1 a 7 e gere um vetor `mu_e` com o grau de pertinência do valor do erro para cada conjunto *fuzzy*, utilizando a função `triang_membership_function` criada.
- Faça o mesmo com o valor da variação do erro.

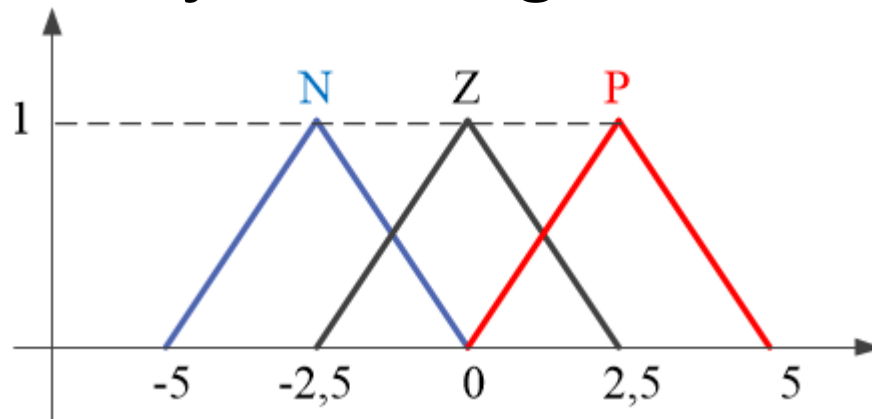
# Sugestões

- Método 1: utilize 49 condições `if` para varrer todas as regras;
- Método 2: solução matricial. Crie uma matriz 7x7, chamada `Mu_s`, que aplica a regra de Mamdani aos valores de `mu_e` e `mu_de` calculados. Isso funciona bem no Matlab, mas em C/C++ talvez seja melhor fazer com os `ifs` mesmo. Por exemplo se fossem apenas 3 regras, seria algo como:

# Sugestões

$$\text{Mu}_s = \begin{bmatrix} \min(\mu_e(1), \mu_{de}(1)) & \min(\mu_e(2), \mu_{de}(1)) & \min(\mu_e(3), \mu_{de}(1)) \\ \min(\mu_e(1), \mu_{de}(2)) & \min(\mu_e(2), \mu_{de}(2)) & \min(\mu_e(3), \mu_{de}(2)) \\ \min(\mu_e(1), \mu_{de}(3)) & \min(\mu_e(2), \mu_{de}(3)) & \min(\mu_e(3), \mu_{de}(3)) \end{bmatrix}$$

- Crie uma matriz  $B_s$  com o valor da abscissa central de saída, de acordo com a tabela de regras. Por exemplo, suponha que os conjuntos *fuzzy* de saída e a tabela de regras sejam da seguinte forma.



	Erro		
Variação do Erro	P	Z	N
	Z	Z	Z
	P	Z	N

# Sugestões

- Assim, tal matriz seria do tipo:

$$B_s = \begin{bmatrix} 2,5 & 0 & -2,5 \\ 0 & 0 & 0 \\ 2,5 & 0 & -2,5 \end{bmatrix}$$

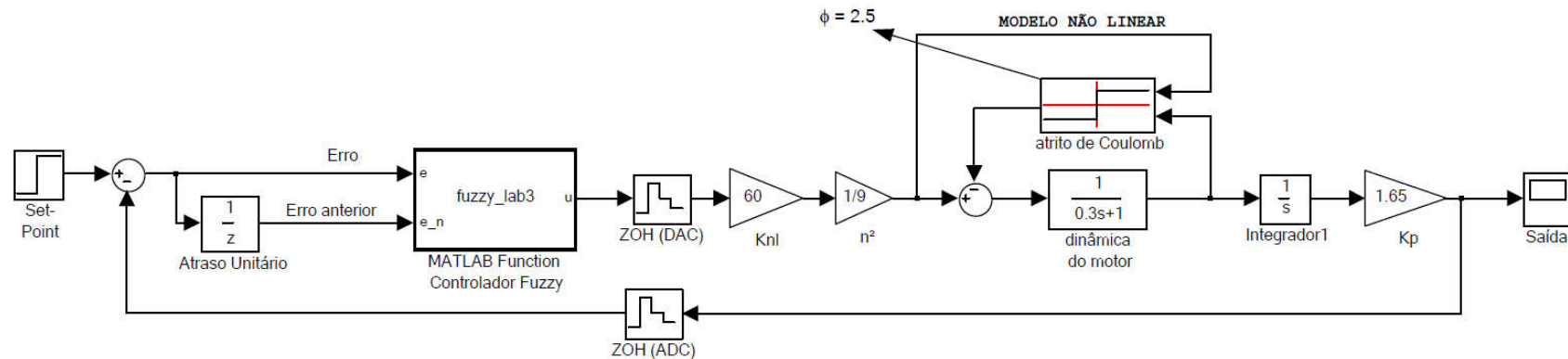
- Depois, se o método da abscissa central for escolhido, então a saída *defuzzificada* no Matlab, fica como:

$$u = \text{sum}(\text{sum}(B_s .* Mu_s)) / \text{sum}(\text{sum}(Mu_s))$$

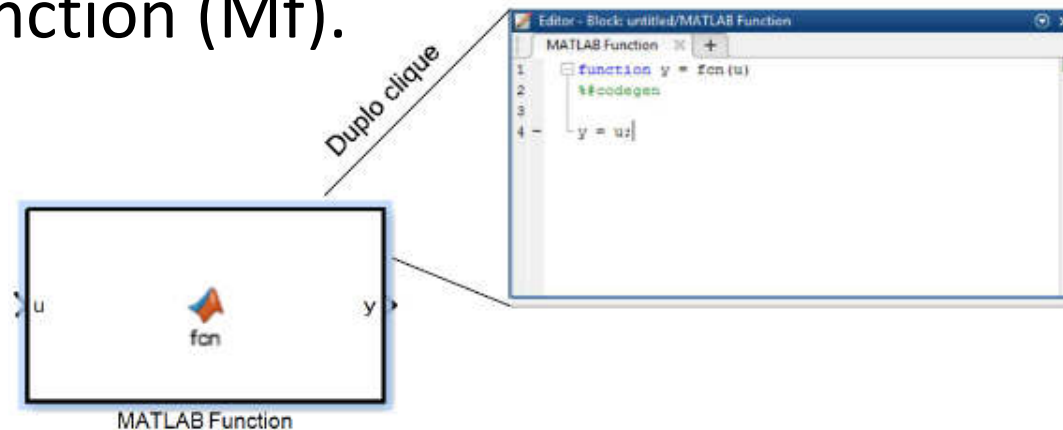
- Se o método do centro de gravidade for escolhido, a partir da matriz  $Mu_s$  e do valor de  $w = 5$ , calcula-se a matriz com as áreas,  $A_s$ , e a saída fica como:

$$u = \text{sum}(\text{sum}(A_s .* B_s)) / \text{sum}(\text{sum}(A_s))$$

# Simulação

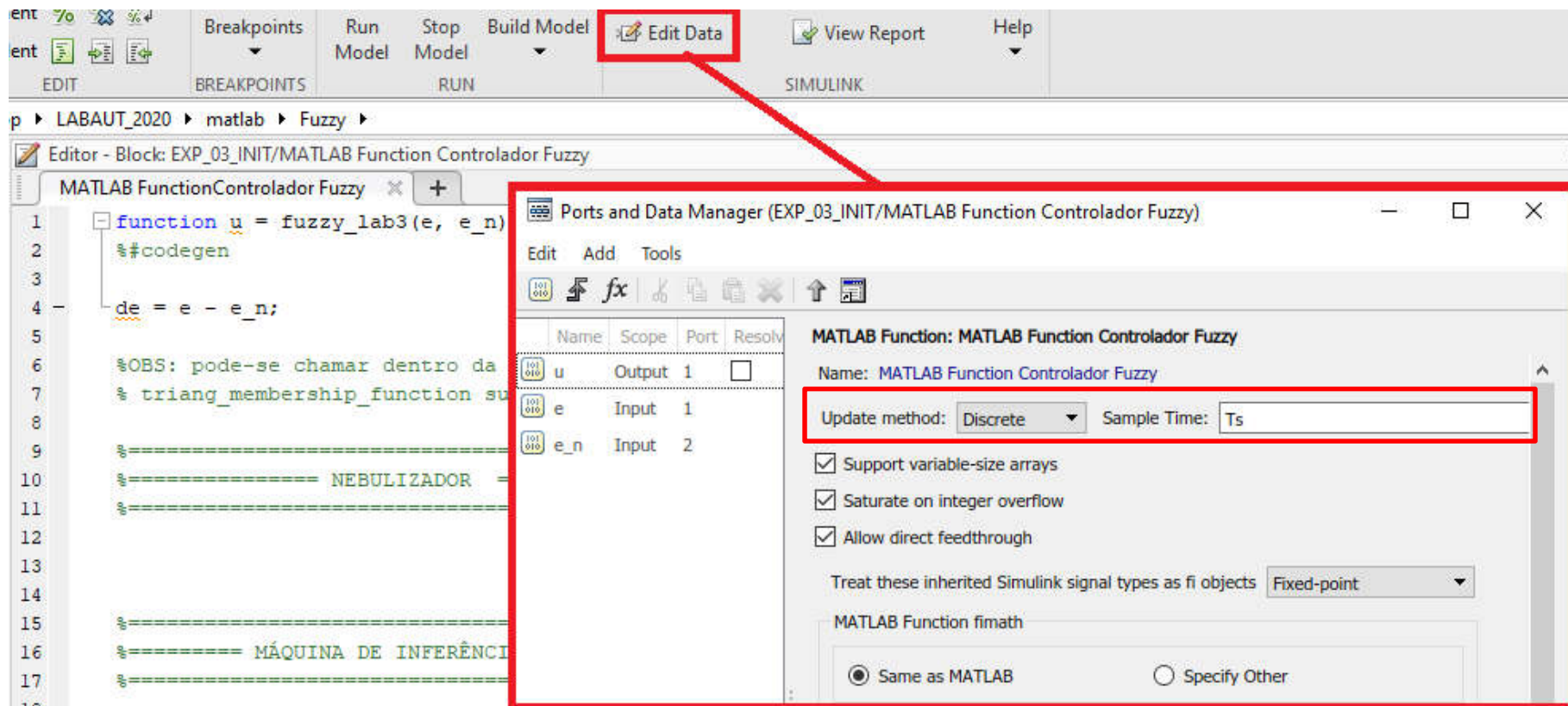


- Considere o arquivo `EXP_03_INIT.slx` que contém o modelo não-linear do servo mecanismo.
- O controlador será implementado utilizando o bloco MATLAB function (Mf).



# Simulação

- Entradas e saídas definidas na função aparecerão como entradas e saídas físicas do bloco.
- Clique em *Editor* -> *Edit Data* e configure o método de atualização:



# Simulação

- Se precisar passar parâmetros para a Mf, faça:
  1. Defina a variável que deseja passar para a Mf (ex, param.);
  2. Clique em Editor -> Edit Data, selecione Parameter em Scope e elimine a seleção Tunable:

The screenshot shows the MATLAB/Simulink environment. The 'Edit Data' button in the top toolbar is highlighted with a red box. A red arrow points from this button to the 'Ports and Data Manager' dialog box, which is also highlighted with a red border. The dialog box shows a table of data parameters and a 'Data param' configuration panel.

Name	Scope
u	Output
e	Input
e_n	Input
param	Parameter

**Data param**

General Description

Name: param

Scope: Parameter  Tunable

Size: -1  Variable size

Complexity: Inherited

First index: Scalar

Type: Inherit: Same as Simulink >>



# Simulação

- Observações:
  1. Implemente o controlador *fuzzy* em linha de código dentro da Matlab function;
  2. Pode-se chamar dentro da Matlab function a função `triang_membership_function.m` sugerida, desde que ela esteja no mesmo diretório do arquivo do Simulink;
  3. Teste diferentes tempos de amostragem (ex:  $T_s = 1/10$  s,  $T_s = 1/20$  s,  $T_s = 1/40$  s. Comente as diferenças;
  4. Pode escolher qualquer forma de “desfuzificação”.
  5. Consegue melhorar o desempenho alterando a base de regras?

# Atividades Práticas

- Explore o conteúdo copiado de `c:/labaut` e leia o manual “Acesso ao Módulo Lynx AC1160-VA” para familiarização com as funções Matlab de configuração e uso do módulo de aquisição de dados Lynx.
- Analise o código do arquivo `prog4.m` e utilize-o como base para elaboração das atividades propostas.

## Programa 4 (prog4.m)

Um programa que implementa um controlador proporcional operando a 5Hz durante 120s.

---

```
%Prog4.m (V1.0 RP Marques)
%Controle proporcional a 5Hz durante 120s (600 pontos)
% Controle:  $u(k) = K_c \cdot e(k)$ , onde  $e(k) = r - y(k)$ 
% r é o setpoint
% y é a saída da planta
% u é a saída do controlador

T = 1/5 ;           % Frequencia de amostragem de 5Hz
U = zeros(1,600);  % Vetor de 2400 pontos para guardar os dados
Y = zeros(1,600);  % Vetor de 2400 pontos para guardar os dados
R = 0;             % Setpoint
Kc = 1;            % Ganho do controlador

das_start;         % Configura o módulo Lynx
srt_conf(T);       % Configura o sistema de tempo real
                  % (colocar imediatamente antes do loop principal)

for k=1:600,       % LOOP PRINCIPAL (início)
    [Y(k), INP1] = das_readAD; % Lê os dados (INP1 não é utilizado)
    U(k) = Kc*(R - Y(k));      % Calcula u(k)
    das_writeDA(U(k),U(k));   % Escreve u(k) no conversor D/A
    srt_waitbusy;            % Aguarda o fim do período de 0,2s
end                          % LOOP PRINCIPAL (final)
```