



UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DISSERTAÇÃO DE MESTRADO

**JOSÉ GUILHERME CASTRO DE ALMEIDA**

**SISTEMA DE CONTROLE BASEADO EM LÓGICA FUZZY PARA REGULAGEM  
DE TEMPERATURA EM AQUECEDOR DE ÓLEO TÉRMICO DA PLANTA DE  
ANODO VERDE DA ALBRÁS**

**DM 37/2011**

UFPA / ITEC / PPGE  
CAMPUS UNIVERSITÁRIO DO GUAMÁ  
66075-900 – BELÉM-PARÁ-BRASIL

**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**DISSERTAÇÃO DE MESTRADO**

**JOSÉ GUILHERME CASTRO DE ALMEIDA**

Dissertação submetida à Banca Examinadora  
do Programa de Pós-Graduação em  
Engenharia Elétrica da UFPA para obtenção  
do Grau de Mestre em Engenharia Elétrica

UFPA / ITEC / PPGE  
CAMPUS UNIVERSITÁRIO DO GUAMÁ  
66075-900 – BELÉM-PARÁ-BRASIL

---

A447s Almeida, José Guilherme Castro de

Sistema de controle baseado em lógica fuzzy para regulação de temperatura em aquecedor de óleo térmico da planta de anodo verde da Albras / José Guilherme Castro de Almeida; orientador, Walter Barra Junior. – 2011.

Dissertação (Mestrado) – Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2011.

1. Controle de processo – modelos matemáticos. 2. Alumínio - indústria. 3. Sistemas difusos. 4. Controladores programáveis. I. Orientador. II. Título.

CDD 22. ed. 670.4275

---

**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**  
**SISTEMA DE CONTROLE BASEADO EM LÓGICA FUZZY PARA REGULAGEM**  
**DE TEMPERATURA EM AQUECEDOR DE ÓLEO TÉRMICO DA PLANTA DE**  
**ANODO VERDE DA ALBRÁS**

**JOSÉ GUILHERME CASTRO DE ALMEIDA**

DISSERTAÇÃO DE MESTRADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DO PARÁ E JULGADA PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE SISTEMA DE ENERGIA.

APROVADA EM 23/12/2001

BANCA EXAMINADORA:

---

Prof. Dr. Walter Barra Junior  
(ORIENTADOR – ITEC/UFPA/PPGE)

---

Prof. Dr. Carlos Tavares da Costa Júnior  
(MEMBRO – ITEC/UFPA/PPGE)

---

Prof. Dr. José Augusto Lima Barreiros  
(MEMBRO – ITEC/UFPA/PPGE)

---

Prof. Dr. André Mauricio Damasceno Ferreira  
(MEMBRO – IFPA)

VISTO:

---

Prof. Dr. Marcus Vinicius Alves Nunes  
(COORDENADOR DO PPGE/ITEC/UFPA)  
UFPA / ITEC / PPGE

## **AGRADECIMENTOS**

Agradeço ao meu Deus por tudo que Ele proporcionou pela vida pelo cuidado que esteve sempre ao meu lado nas horas mais difíceis.

Aos meus pais que sempre me incentivaram com palavras de apoio e conselhos que com certeza contribuíram para minha formação.

A minha esposa e a minha filha que me dão energia em todos os momentos.

Ao meu amigo MSc Edson Silva Cruz pela orientação técnica na elaboração deste trabalho.

Ao meu orientador Dr. Walter Barra Junior por ter acreditado no meu trabalho.

## LISTA DE SÍMBOLOS

<b>e</b>	Erro
<b>ce=Δe</b>	Varição do Erro
<b>P<sub>m</sub></b>	Pressão Diferencial Média
<b>U<sub>p</sub></b>	Saída do Controlador Fuzzy
<b>SP</b>	Set Point
°	Composição de Relações <i>Fuzzy</i>
∧	Conjunção lógica
→	Implicação Lógica <i>Fuzzy</i>
∩	Intersecção de conjuntos
∉	Não pertence
~	Negação Lógica
∈	Pertence
∪	União de conjuntos
∨	União lógica
$\mu_A$	Índice de pertinência em relação ao conjunto A
<b>A</b>	Conjunto <i>Fuzzy</i>
<b>A'</b>	Complemento de um Conjunto <i>Fuzzy</i> A
<b>MOM</b>	Média dos Máximos
<b>COG</b>	Centro de gravidade
<b>R</b>	Relação <i>Fuzzy</i>
<b>X<sub>i</sub></b>	Universo de ordem i
<b>Min</b>	Mínimo
<b>Max</b>	Máximo
<b>CLP</b>	Controlador Lógico Programável
<b>Min_Erro_N</b>	Limite Mínimo do Predicado Negativo
<b>Max_Erro_N</b>	Limite Máximo do Predicado Negativo
<b>Min_Erro_Z</b>	Limite Mínimo do Predicado Zero
<b>Max_Erro_Z</b>	Limite Máximo do Predicado Zero
<b>Min_Erro_P</b>	Limite Mínimo do Predicado Positivo
<b>Max_Erro_P</b>	Limite Máximo do Predicado Positivo

## LISTA DE FIGURAS

FIG.1. 1 – FORNO DE INDUÇÃO	15
FIG.1. 2 – DETALHE DO FORNO DE REDUÇÃO	15
FIG.1. 3 .A– BANHO ELETROLITICO	16
FIG.1. 3.B – ANODOS GASTOS	16
FIG. 1.4 – RESUMO DO PROCESSO	17
FIG.1. 5 – ANODO CHUMBADO	18
FIG.1. 6 – ANODO POSTADO NA CUBA	18
FIG.1.7 – FORNO DE COZIMENTO	18
FIG.1. 8 – BUTT’S	18
FIG.1. 9 – COMPOSIÇÃO DO BLOCO ANODICO	19
FIG.1.10 – FLUXOGRAMA TÍPICO	20
FIG.1.11 – PENEIRA	21
FIG.1.12 – MOINHO DE BOLAS	21
FIG.1.13 – BALANÇA LOSS WEIGHT	22
FIG.1.14 – RECEITA DO PROCESSO	22
FIG.1.15 – PRÉ- AQUECEDOR DE COQUE	23
FIG.1.16 – MISTURADOR DE PASTA	24
FIG. 1.17 – MISTURADOR NO SUPERVISÓRIO	24
FIG. 1.18 – ENCHIMENTO	24
FIG. 1.19- VIBRAÇÃO.	24
FIG. 1.20- EJEÇÃO	24
FIG.1.2 1 – COMPACTADOR SISTEMA SUPERVISÓRIO	25
FIG.2.1 – SISTEMA DE INFERÊNCIA FUZZY	36
FIG.3.1– ESTRUTURA BÁSICA DE UM CONTROLADOR FUZZY	38
FIG.3.2 – REPRESENTAÇÃO GRÁFICA DO UNIVERSO FUZZY	40
FIG.3.3 – PROCEDIMENTO DE QUANTIZAÇÃO	41
FIG.3.4 – REPRESENTAÇÃO GRÁFICA DO UNIVERSO FUZZY DE 13 PREDICADOS	45
FIG.3.5 – REPRESENTAÇÃO GRÁFICA DOS PREDICADOS NP E ZE	45
FIG.3.6 – REPRESENTAÇÃO GRÁFICA DA SAÍDA DO CONTROLADOR	46

FIG.3.7 – REPRESENTAÇÃO DO ESPAÇO DE ESTADO DO ERRO E VARIÇÃO ERRO	48
FIG.3.8 – ESPAÇO DE FASE DAS VARIÁVEIS DO CONTROLADOR	51
FIG.4.1 – UNIDADE DE AQUECIMENTO MTHU-12	54
FIG.4.2 – UNIDADE DE POTÊNCIA ADELCO	54
FIG.4.3 – RESISTÊNCIA DE 166 KW TIPO TUBOLARES BLINDADAS	54
FIG.4.4 – FLUXO DO SISTEMA DE AQUECIMENTO	54
FIG.4.5 –PAINEL DE FORÇA DA UNIDADE TIRISTORIZADA	55
FIG.4.6 – ESQUEMA ELÉTRICO DA LIGAÇÃO DO TIRISTOR	55
FIG.4.7 – CHASSI COM: PROCESSADOR, CARTÕES DE ENTRADA E SAÍDA ANALÓGICA E DIGITAIS	57
FIG. 4.8 – ESTRUTURA BÁSICA DO CLP	57
FIG. 4.9 –BLOCO INICIALIZADOR DA VARIÁVEL ERRO	57
FIG. 4.10-BLOCO INICIALIZADOR DA VARIÁVEL VARIÇÃO DO ERRO	58
FIG. 4.11- BLOCO FUZZYFICADOR DA VARIÁVEL ERRO	58
FIG.4.12 – BLOCO FUZZYFICADOR DA VARIÁVEL VARIÇÃO DO ERRO	58
FIG.4.13 – BLOCO DE APLICAÇÃO DAS REGRAS FUZZY	59
FIG.4.14– BLOCO MÁXIMO/ MÍNIMOS FUZZY	59
FIG.4.15 – BLOCO DEFUZZYDICADOR	60
FIG.4.16 –TELA INICIAL DO COMPILADOR FUZZY	61
FIG.4.17 – ENTRADA DAS GRANDEZAS DE ATUAÇÃO	61
FIG.4.18 – TELA DE ENTRADA DE REGRAS	62
FIG.4.19 – TELA DE GERAÇÃO DE REGRAS	62
FIG.4.20 – DIAGRAMA ESQUEMÁTICO DA ESTRUTURA DO COMPILADOR FUZZY	64
FIG.4.21 – DIAGRAMA ESQUEMÁTICO DO COMPILADOR FUZZY DO AQUECEDOR DE ÓLEO	65
FIG.4.22 – FUNÇÃO DE PERTINÊNCIA TRIANGULAR DO ERRO	67
FIG.4.23 – FUNÇÃO DE PERTINÊNCIA TRIANGULAR DO INCREMENTO DO ERRO	67
FIG.4.24 – FUNÇÃO DE PERTINÊNCIA TRIANGULAR DA SAÍDA	67
FIG.4.25 – FATOR DE AJUSTE N	68
FIG.4.26 – ANÁLISE DINÂMICA DO SINAL DO ERRO E SUA DERIVADA	69

FIG. 4.27 – VARIÁVEL ERRO	71
FIG. 4.28 – VARIÁVEL VARIAÇÃO DO ERRO	71
FIG. 4.29- DEFUZIFICAÇÃO POR CENTRO DE GRAVIDADE	73
FIG.5.1 – SISTEMA DE CONTROLE DE TEMPERATURA NO SUPERVISÓRIO	75
FIG. 5.2- SALA DE CONTROLE DO SISTEMA SUPERVISÓRIO	76
FIG. 5.3 - MALHA DE CONTROLE DO SISTEMA DE AQUECIMENTO	77
FIG.5.4 – RASTREAMENTO DO CONTROLADOR FUZZY PARA NOVOS PONTOS DE OPERAÇÃO	78
FIG.5.5 – VELOCIDADE DE RESPOSTA DO CONTROLADOR FUZZY	79
FIG. 5.5 – VELOCIDADE DE RESPOSTA DO CONTROLADOR FUZZY	79
FIG.5.6 – ALCANCE DO PONTO DE ACOMODAÇÃO DO CONTROLADOR FUZZY	79
EM TORNO DO <i>SET-POINT</i>	
FIG.5.7 – CONTROLADOR REAGINDO A UMA MUDANÇA DA VARIÁVEL DE PROCESSO	80
FIG.5.8 - CONTROLADOR FUZZY EM ESTADO PERMANENTE DE ACOMODAÇÃO	81
FIG.5.9 - RESPOSTA A UM SINAL DEGRAU CONTROLADOR FUZZY	82
FIG.5.10- RESPOSTA A UM SINAL DEGRAU CONTROLADOR WATLOW PID	82

## LISTA DE TABELAS

Tabela 1- GRANULOMETRIA TÍPICA DAS FRAÇÕES DE COQUE.	21
Tabela 2- UNIVERSO	40
Tabela 3 - VARIÁVEIS LINGÜÍSTICAS	66
Tabela 4 - REGRAS LINGUÍSTICAS DO CONTROLADOR FUZZY	70
Tabela 5 – REGRAS LINGUÍSTICAS DO FATOR DE AJUSTE N	70
Tabela 6- REGRAS LINGUÍSTICAS	72
Tabela 7- CARACTERÍSTICAS DOS CONTROLADORES PID E FUZZY	83

## SUMÁRIO

<b>CAPÍTULO 1</b>	11
1.1. CONSIDERAÇÕES GERAIS	11
1.2. DESCRIÇÃO DO PROCESSO DA FÁBRICA ALBRAS	14
1.2.1. Processo de Fabricação do Alumínio	14
1.2.2. Processo de Fabricação do Anodo Verde	17
1.3. ORGANIZAÇÃO DO TRABALHO	26
<b>CAPÍTULO 2</b>	27
2.1. FUNDAMENTOS DE LÓGICA FUZZY	27
2.1.2. Introdução	27
2.2. CONJUNTOS FUZZY	28
2.2.1. Fundamentos	28
2.2.2. Operações com Conjuntos Fuzzy	30
2.2.2.1. Operações com Conjuntos Fuzzy	30
2.2.3. Variáveis Linguísticas	31
2.2.4. Funções de Implicação	33
2.2.5. Regra de Inferência Composicional	35
2.2.6. Sistema de Inferência Fuzzy	36
2.3. CONCLUSÃO	37
<b>CAPÍTULO 3</b>	38

3.1.CONTROLE FUZZY	38
3.1.1.Introdução	38
3.2. INTERFACE COM O PROCESSO	39
3.2.1. Variáveis de Controle	39
3.2.2. Escalonamento e Quantização	40
3.2.3. Métodos de Defuzzificação	42
3.3. EXPRESSÃO DE REGRAS DE CONTROLE EM TERMOS MATEMÁTICOS	42
3.4. ASPECTOS DE IMPLEMENTAÇÃO	46
3.4.1. Armazenamento de Regras	46
3.4.2. Contribuição de cada Regra	48
3.4.3. Influência das Funções de Pertinência	49
3.4.4. Influência das Funções de Implicação	50
3.4.5. Espaço de Fase	50
3.5. CONCLUSÃO	51
<b>CAPÍTULO 4</b>	<b>53</b>
4.1. PROJETO E IMPLEMENTAÇÃO DO CONTROLE FUZZY DO AQUECEDOR DE ÓLEO TÉRMICO (HTM)	53
4.1.1.Introdução	53
4.1.2. Características do Sistema de Aquecimento de Óleo Térmico	53
4.1.3. Controlador Lógico Programável (CLP) Utilizado na Implementação Controle Fuzzy na Planta da Albrás	55
4.1.3.1. Controlador programável (clp) Linguagem Ladder	57

4.1.4. Gerador de Algoritmos fuzzy	60
4.1.5. Modelagem e Análise do Controlador Fuzzy	63
4.1.5.1. Fuzzificação	64
4.1.5.2. Funções de Pertinências	66
4.1.5.3. Análise Dinâmica da Variável do Erro	68
4.1.5.4. Dedução das Regras de Controle Fuzzy	70
4.1.5.5. Máquina de Inferência	70
4.1.5.6. Defuzificação	72
4.1.5.7. Conclusão	74
<b>CAPÍTULO 5</b>	<b>75</b>
5.1. RESULTADOS EXPERIMENTAIS NA PLANTA DA ALBRAS	75
5.1.1. Considerações Gerais	75
5.2. CONTROLE FUZZY NO SISTEMA DE AQUECIMENTO DE ÓLEO	76
5.3. COMPARAÇÃO DOS CONTROLADORES FUZZY E PID	81
5.4. CONCLUSÃO	84
<b>CAPÍTULO 6</b>	<b>85</b>
6.1. CONCLUSÃO	85
6.2. TRABALHOS FUTUROS	86
REFERÊNCIAS	87
ANEXOS	

## RESUMO

Neste trabalho são apresentados os resultados de uma técnica que permitiu implementar a estratégia de controle de temperatura do aquecedor de óleo térmico da fábrica de Anodo Verde da Albrás Alumínio Brasileiro S/A. No projeto utilizou-se um sistema hierarquizado baseado em conjuntos e lógica Fuzzy. O uso dessa metodologia fez com que o sistema fosse capaz de reagir adequadamente diante das variações do ponto de operação do aquecedor, pois o controle Fuzzy exibe algumas características do aprendizado humano, sendo considerado um exemplo de inteligência artificial. O aquecedor de óleo térmico é fundamental no processo de fabricação de blocos inódicos, utilizados como pólo positivo no processo de eletrólise na obtenção do alumínio primário. O sistema de óleo térmico aquece os misturadores e pré-aquecedor de coque, mantendo a temperatura desses equipamentos dentro dos limites estabelecidos pela engenharia de processo. A variável temperatura impacta diretamente na energia de mistura da pasta e na qualidade do produto final, que é o bloco anódico. A metodologia apresentada permitiu alcançar um controle de temperatura que atendeu satisfatoriamente os parâmetros de processo. O programa foi desenvolvido em linguagem ladder é executado em controladores lógicos programáveis (CLP'S) da Rockwell Automation. O controle já está em plena operação nas fábricas de anodos e os resultados obtidos demonstram a eficácia e viabilidade do sistema, que futuramente estará sendo implementado no controle de outros equipamentos da Albrás.

**Palavras-chave:** Controle Fuzzy; Aquecedor de óleo Térmico; Controlador Lógico Programável;

## ABSTRACT

This work presents the results of a fuzzy control strategy to regulate the temperature of the heater thermal oil, in the Anode Green process at Albrás Aluminum Brasileiro S/A factory. The use of this methodology has resulted in the system being able to react adequately against of the variations on the point of operation of the heater, because the Fuzzy control displays some characteristics of human learning. The heater thermal oil equipment is fundamental in the process of manufacture of blocks butts, used as positive pole in the process of electrolysis in the acquisition of primary aluminum. The system of thermal oil warms the mixers and pre-heating of coke, keeping the temperature of the equipment within the limits established by process engineering. The variable temperature impacts directly in the energy mix of the pulp and the quality of the final product, which is block anode. The methodology presented allowed to reach a temperature control that satisfactorily achieved the process parameters. The program was developed in language scada is executed in programmable logic controllers (PLC'S) of Rockwell Automation. The methodology presented allowed to reach a temperature control that satisfactorily achieved the process parameters. The control is already in full operation at the plants of anodes and the results obtained demonstrate the effectiveness and viability of the system, which in future will be implemented in the control of other equipment of Albras.

**Keywords** – Fuzzy Control ,Thermal Oil Heater, Programmable Logic Controller.

## CAPÍTULO 1

### 1.1 – CONSIDERAÇÕES GERAIS

A reprodução de características inteligentes em máquinas construídas pelo homem é um dos grandes objetivos almejados pela comunidade científica e tecnológica. Muitos paradigmas simbólicos de aprendizagem surgiram, e muitos se desenvolveram, como métodos computacionais poderosos, incluindo aquisição indutiva de conceitos, sistemas classificadores, e aprendizagem baseada em explicações.

Um conjunto de ferramentas analíticas, conhecidas coletivamente como Inteligência Computacional, tem sido utilizado para solucionar problemas relacionados à Aprendizagem de Máquinas, permitindo assim a resolução de problemas que eram considerados difíceis ou impossíveis de serem tratados. A característica de destaque da Inteligência Computacional é que seu objetivo primário é alcançar tratabilidade, robustez, baixo custo e alto coeficiente de Inteligência de Máquina, através de uma exploração da tolerância à imprecisão e incerteza. O que se busca nesse contexto é uma solução aproximada para um problema formulado imprecisamente [1,2,3,4].

Um exemplo simples é o de estacionar um carro. Estaciona-se um carro com relativa facilidade porque a posição final e a orientação do carro não são especificadas precisamente. Se o fossem, o problema seria praticamente insolúvel para os seres humanos. Em contraste, quando o problema de se estacionar é formulado imprecisamente, torna-se difícil de resolvê-lo pelos métodos computacionais convencionais, porque tais métodos não tratam do problema de se explorar a tolerância por imprecisão.

O que esse exemplo simples quer ilustrar é o fato que, em geral, uma precisão alta acarreta um alto custo. O desafio, portanto, é explorar a tolerância à imprecisão pela busca de novos métodos de computação que levem a uma solução aceitável a um baixo custo. Pela sua própria natureza, a Inteligência Computacional é muito mais próxima ao raciocínio humano do que os modos tradicionais de computação. Nessa combinação, os principais constituintes da Inteligência Computacional são as Redes Neurais Artificiais, a Lógica Fuzzy

(Nebulosa), o Raciocínio Probabilístico e os Sistemas Caóticos. As Redes Neurais se preocupam principalmente com a teoria da aprendizagem; a Lógica Fuzzy, com a imprecisão e o raciocínio aproximado e o Raciocínio Probabilístico (Algoritmos Genéticos), com o gerenciamento da incerteza.

No desenvolvimento dissertação será enfocada e utilizada a Lógica Fuzzy, em razão das suas qualidades em manusear informações qualitativas e imprecisas de uma maneira rigorosa, resultando em um controlador fuzzy, que propiciará um controle mais preciso, além de um desempenho estável e robusto, visando melhorar as qualidades do processo de fabricação de anodo na Albras – Alumínio Brasileiro S. A..

A grande simplicidade de implementação de sistemas de controle fuzzy pode reduzir a complexidade a um ponto em que problemas anteriormente intratáveis passam agora a ser solúveis. A Lógica Fuzzy é a lógica que suporta os modos de raciocínio que são aproximados ao invés de exatos. Controle fuzzy e Modelagem de sistemas são técnicas para o tratamento de informações qualitativas de uma forma rigorosa. Derivada do conceito de conjuntos fuzzy, a Lógica Fuzzy constitui a base para o desenvolvimento de métodos e algoritmos de modelagem e controle de processos, permitindo a redução da complexidade de projeto e implementação, tornando-se a solução para problemas de controle até então intratáveis por técnicas clássicas. Ela difere dos sistemas lógicos em suas características e seus detalhes. Nesta lógica, o raciocínio exato corresponde a um caso limite do raciocínio aproximado, sendo interpretado como um processo de composição de relações nebulosas.

Na Lógica Fuzzy, o valor verdade de uma proposição pode ser um subconjunto fuzzy de qualquer conjunto parcialmente ordenado, ao contrário dos sistemas lógicos binários, onde o valor verdade só pode assumir dois valores: verdadeiro (1) ou falso (0). Nos sistemas lógicos multivalores, o valor verdade de uma proposição pode ser um elemento de um conjunto finito, num intervalo, ou uma álgebra booleana. Na lógica nebulosa, os valores verdade são expressos linguisticamente (verdade, muito verdade, não verdade, falso, muito falso), onde cada termo linguístico é interpretado como um subconjunto fuzzy do intervalo unitário. Outras características da lógica fuzzy podem ser sumarizadas como segue [5]:

- nos sistemas lógicos binários, os predicados são exatos (par, maior que), ao passo que, na lógica fuzzy os predicados são nebulosos (alto, baixo);
- nos sistemas lógicos clássicos, o modificador mais utilizado é a negação, enquanto que na lógica fuzzy uma variedade de modificadores de predicados são possíveis (muito, mais ou menos). Estes modificadores são essenciais na geração de termos linguísticos, tais como: muito alto, mais ou menos perto, etc);
- na lógica clássica existem somente os quantificadores existenciais e universais. A lógica fuzzy admite, em adição, uma ampla variedade de quantificadores, como: pouco, vários, usualmente, freqüentemente, em torno de cinco, etc;
- a probabilidade, no contexto da lógica clássica, é um valor numérico ou um intervalo. Na lógica nebulosa existe a opção adicional de se empregar probabilidades linguísticas (provável, altamente provável, improvável), interpretadas como números fuzzy e manipuladas pela aritmética fuzzy. Também em contraste com a lógica modal clássica, o conceito de possibilidade é interpretado utilizando-se subconjuntos fuzzy no universo dos reais.

Em teorias de controle clássica e moderna, o primeiro passo para implementar o controle de um processo é derivar o modelo matemático que descreve o mesmo [3,6]. O procedimento, requer que se conheça detalhadamente o processo a ser controlado, o que nem sempre é factível se o processo é muito complicado. As teorias de controle existentes se aplicam a uma grande variedade de sistemas, onde o processo é bem definido. Várias técnicas, tais como para controle linear multivariável, estimação de estado a partir de medidas ruidosas, controle ótimo, sistemas lineares estocásticos, além de certas classes de problemas não-lineares determinísticos, foram desenvolvidas e aplicadas com sucesso em um grande número de problemas bem postulados. Entretanto, todas estas técnicas não são capazes de resolver problemas reais cuja modelagem matemática é impraticável. Por exemplo, em diversas situações um volume

considerável de informações essenciais só é conhecido, a priori, de forma qualitativa. Do mesmo modo, critérios de desempenho só estão disponíveis em termos linguísticos.

Este panorama leva a imprecisões e falta de exatidão que inviabilizam a maioria das teorias utilizadas até agora. A modelagem e o controle fuzzy consideram o modo como a falta de exatidão e a incerteza são descritas e, fazendo isso, tornam-se suficientemente poderosas para manipular de maneira conveniente o conhecimento. A sua utilização em sistemas de controle de processos em tempo real, em computadores é das mais convenientes, dado que, geralmente não envolvem problema computacional sério. Controle fuzzy trata do relacionamento entre entradas e saídas, agregando vários parâmetros de processo e de controle.

A lógica fuzzy pode ser aceita como a melhor ferramenta para modelar o raciocínio humano, que é aproximado e parcial em sua essência. A teoria dos conjuntos e lógica fuzzy objetiva modelar os modos de representação e raciocínio imprecisos que têm um papel essencial na tomada de decisões racionais em ambientes de imprecisão e incerteza. A diversificação de tecnologias advinda da lógica fuzzy tem também permitido sua aplicação em diversas áreas de conhecimento. O processo pode ser controlado a partir de um conjunto de regras fuzzy do tipo "Se-Então", capaz de coordenar conhecimentos incompletos, incertos ou mesmo conflitantes.

## 1.1 – DESCRIÇÃO DO PROCESSO DA FÁBRICA ALBRAS

### 1.1.1 – Processo de Fabricação do Alumínio

A operação do processo Hall-Héroult se realiza pela eletrólise da alumina em fusão, em que se empregam fornos especiais (cubas eletrolíticas)[30]. A alumina purificada é dissolvida em criolita fundida e sofre eletrólise pela ação de corrente direta a baixa voltagem. Sob a influência da corrente, o oxigênio da alumina ( $Al_2O_3$ ) é depositado no anodo e é liberado como dióxido de carbono, enquanto o alumínio líquido liberado é depositado no pólo inferior da célula. O alumínio fundido é retirado periodicamente.

O processo é contínuo e basta adicionar alumina à proporção que for sendo decomposta e consumida. Depois de retirado da cuba, o alumínio é novamente fundido e reduzido a lingotes Figura 1.1. [30]



Figura-1.1 Forno de Redução

Este processo tem por base a eletrólise (processo de oxi-redução), que é a obtenção de uma reação química devido à passagem de corrente elétrica por um condutor denominado eletrólito. Faz-se necessário a eletricidade por ser uma reação não espontânea. É essencial para isto um anodo (eletrodo positivo) e um catodo (eletrodo negativo) conforme pode ser visto na Figura 1.2. [30]

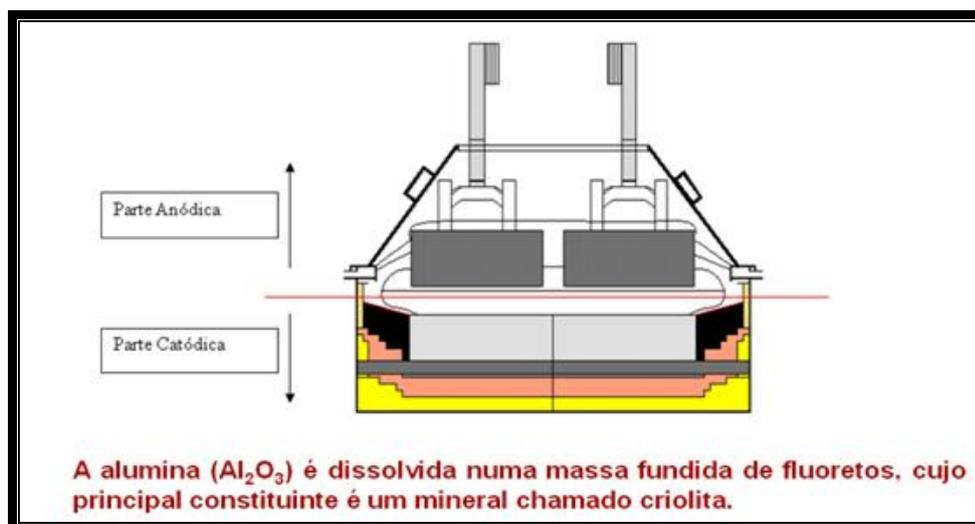
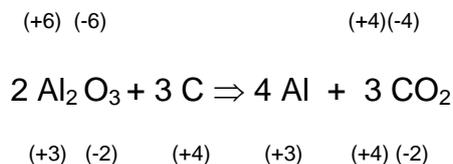


Figura 1.2- Detalhe do forno de redução.

O processo Hall-Heroult é representado pela seguinte reação:



A produção de alumínio ocorrerá em células eletrolíticas ou cubas, a aproximadamente 960° C. Os blocos de carbono (anodos), serão imersos no banho (Figura 1.3.a) e trocados a cada 25 dias, em média. Os anodos gastos são denominados de Butt (Figura 1.3.b).



Figura 1.3.a.



Figura 1.3.b.

No processo Hall-Hérout, o carbono do anodo é consumido na reação do forno eletrolítico. O consumo efetivo real de carbono está entre 400 e 450 kg/ton. de metal produzido, enquanto que o valor teórico é de 333 kg/ton. de metal. Este consumo adicional ocorre principalmente devido a:

- ✓ Reações secundárias, tais como, oxidação pelo ar e reação inversa do alumínio com o dióxido de carbono
- ✓ Erosão da superfície do anodo (ataque por ar e CO<sub>2</sub>)
- ✓ Perdas mecânicas devido a trincas e quebras

Os principais insumos e matérias-primas para a produção de uma tonelada de alumínio primário são:

- Alumina 1.930 kg
- Coque 405 kg
- Piche 100 kg
- Criolita 12 kg
- Fluoreto de alumínio 23 kg
- Energia elétrica 15.400 kwh

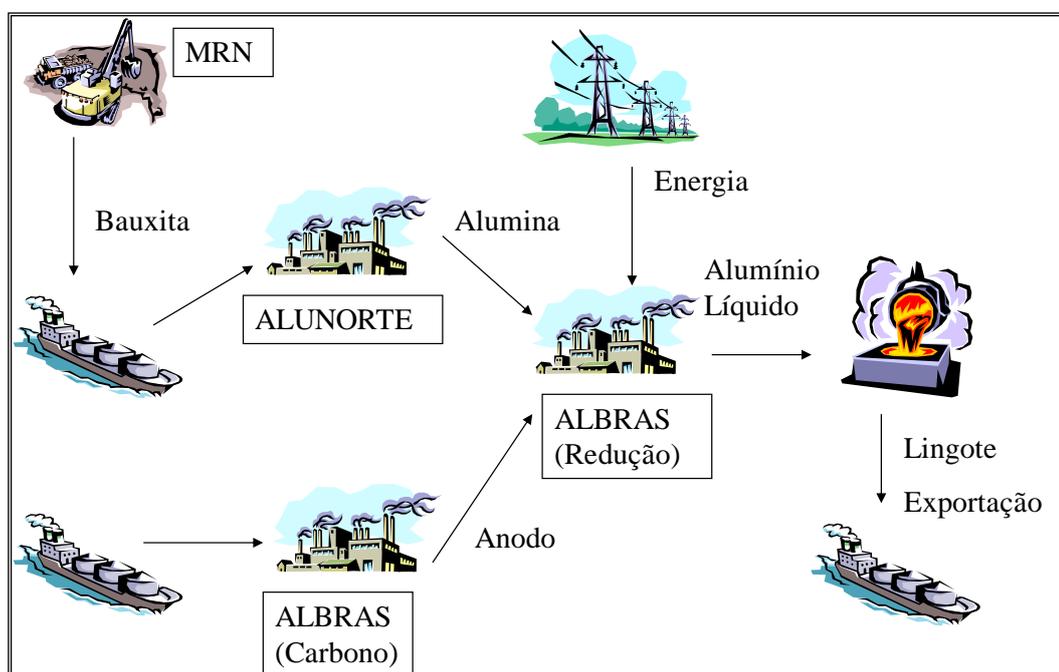


Figura 1.4- Resumo do Processo

### 1.1.2 – Processo de Fabricação do Anodo Verde

Anodo (pólo positivo (Figura 1.5) da célula de redução) é uma peça mergulhada em um banho eletrolítico, conectada a uma fonte de F.E.M., através da qual a corrente penetra no banho e para a qual convergem as partículas do banho portadoras de carga negativa (ânions). O pólo negativo do banho é chamado de cátodo Cuba Figura-1.6 [30].

O anodo utilizado na indústria de alumínio é um bloco de carbono policristalino com uma estrutura não completamente desenvolvida, composto de

coque e piche. São empregados atualmente dois tipos de anodos nas células para produção de alumínio: anodo Soderberg e anodo pré-cozido.



Figura 1.5- Anodo Chumbado.

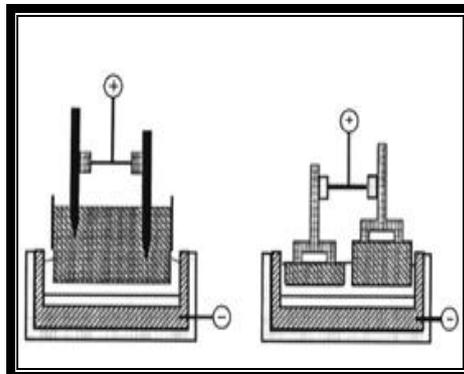


Figura 1.6- Anodo postado na cuba.

Atualmente, as fábricas de anodos estão sendo construídas de acordo com a tecnologia de anodo pré-cozido, isto é, os anodos precisam ser cozidos em separado em outro forno Figura 1.7. Barras de contato são fixadas aos anodos e, após haver sido consumido cerca de 80% do corpo do anodo, normalmente após quatro semanas, o anodo é substituído por um outro. A parte remanescente, chamado de "butt" (Figura 1.7) é reciclado.

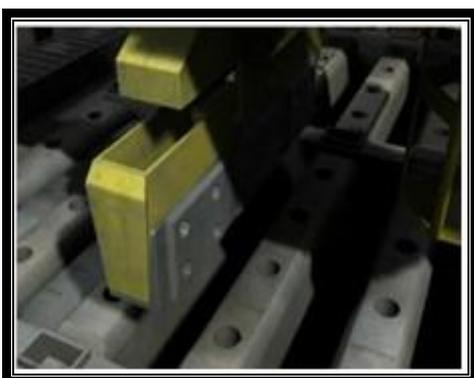


Figura 1.7- Forno de Cozimento



Figura 1.8- Butt

A única desvantagem da tecnologia dos fornos de anodos pré-cozidos é o maior investimento inicial se comparados aos fornos Soderberg.

Existe a necessidade de investir em:

- Instalações para prensagem ou compactação dos blocos anódicos (Fábrica de Anodo Verde);
- Fornos de cozimento de anodos
- Instalações para Chumbamento de Hastes nos anodos e reaproveitamento dos restos de anodos usados (butts).

As matérias-primas utilizadas para a produção de um anodo são:

- |                  |             |
|------------------|-------------|
| • Coque          | 55-65%      |
| • Piche          | 13-15%      |
| • Butt           | 20-30%      |
| • Rejeito Verde  | ~ 1% - 1,5% |
| • Rejeito Cozido | < 0,5%      |

O objetivo da Fábrica de Anodo Verde (Figura 1.10- Fluxo ) e produzir a pasta a partir das matérias-primas e conformar o bloco anódico(Figura 1.9).

As principais etapas são: Preparação do Agregado Seco (Britagem, Moagem e Classificação), Dosagem, Pré-Aquecimento do Agregado, Mistura e Compactação [30].

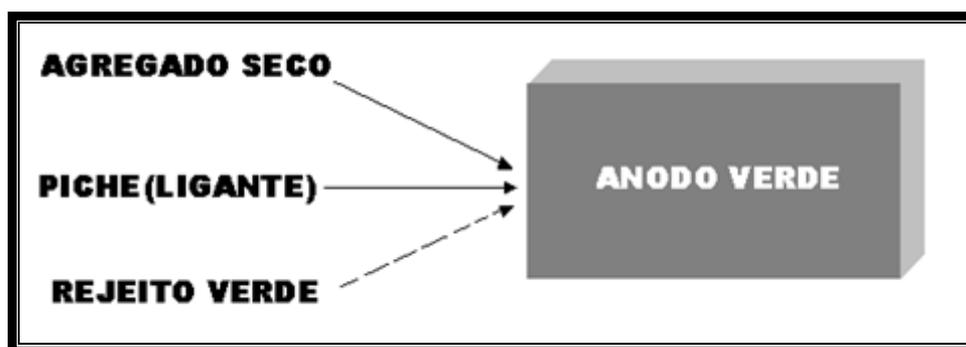


Figura1.9 – Composição do bloco anódico.

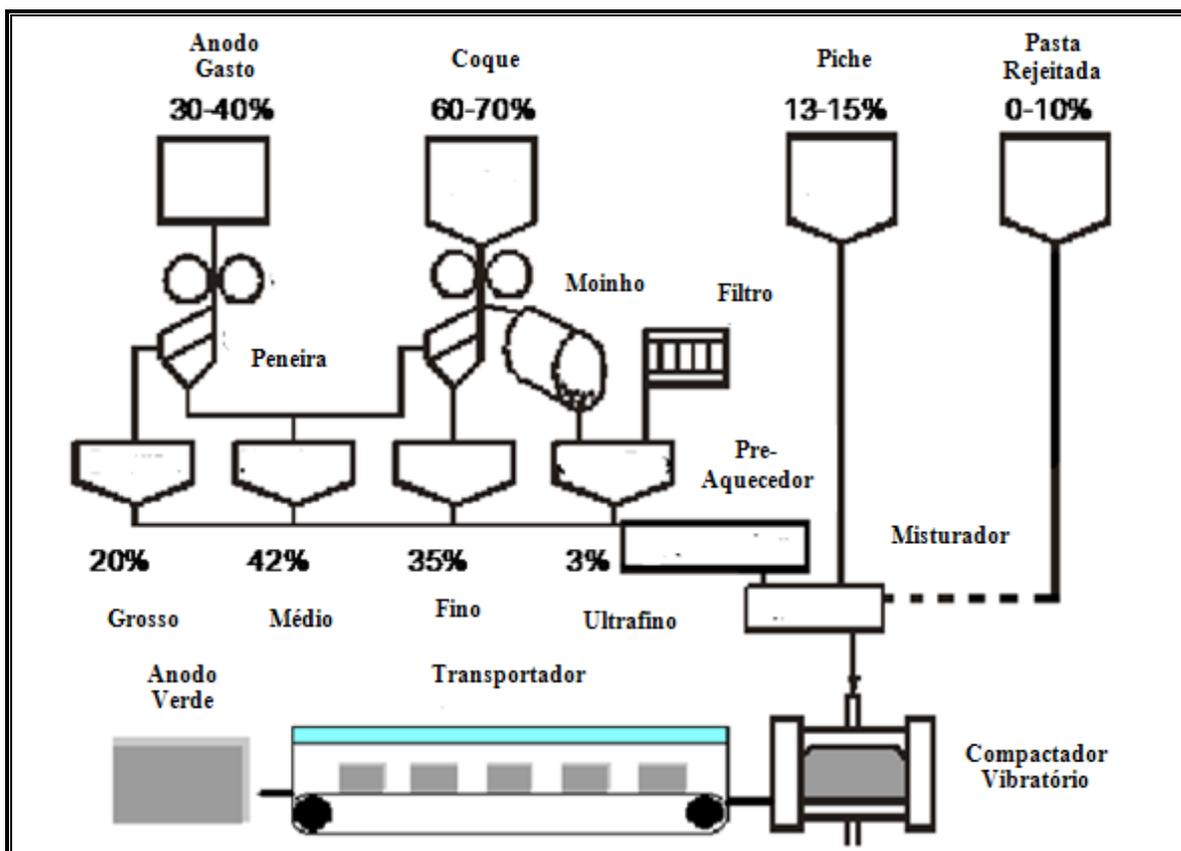


Figura1.10 – Fluxograma típico.

#### 1.2.2.1- Preparação do agregado seco

O agregado seco é preparado em relação a uma receita pré-determinada, na qual é definida a proporção de Coque e Butt a ser utilizado, por exemplo 65% de coque e 35% de butt [30]. A sua preparação consiste na separação granulométrica do coque e butt, formando quatro frações: Grosso, Médio, Fino e Ultrafino. Uma fração intermediária (misto) é necessária para a produção de Fino. Nesta etapa se faz necessário o uso de equipamentos tais como: britadores cônicos ou de rolos, peneiras (Figura 1.11) e Moinho de bolas (Figura 1.12).

A estocagem das frações é feita em silos com capacidades apropriadas. A granulometria típica das frações consta na Tabela 1 a seguir:

Tabela 1- Granulometria típica das frações de coque.

<b>GROSSO</b>	<b>2 a 17 mm</b>
<b>MÉDIO</b>	<b>0,2 a 3,5 mm</b>
<b>FINO</b>	<b>&lt; 0,85 mm (20#) / 2500 a 3000 Blaines</b>
<b>ULTRAFINO</b>	<b>&lt;0,045 mm (325 #) / 8000 a 10000 Blaines</b>



Figura 1.11- Peneira



Figura 1.12- Moinho de bolas

### 1.2.2.2- Dosagem

A dosagem da receita é feita por balanças dosadoras, e a receita (Figura 1.14) é calculada previamente. Balanças Loss Weight (Figura 1.13)



Figura 1.13- Balanças Loss Weight

Produção de Agregado	%
Coque	> = 60
Butt	< = 40
<b>Total</b>	<b>100</b>
<b>Dosagem do agregado</b>	<b>%</b>
Grosso	20
Médio	42
Fino + ultrafino	38
<b>Total</b>	<b>100</b>
<b>Dosagem do piche</b>	<b>%</b>
Agregado seco	85 a 87,5 %
Piche	12,5 a 15 %
<b>Total</b>	<b>100</b>
<b>Dosagem de rejeito verde</b>	<b>%</b>
Pasta	90 a 100 %
Rejeito Verde	0 a 10 %
<b>Total</b>	<b>100</b>

Figura 1.14- Receita do Processo.

### 1.2.2.3 - Pré-aquecimento do Agregado

Tem a finalidade de homogeneizar a mistura de coque e butt e pré aquecê-la. O aquecimento é feito por circulação forçada do fluido térmico; a temperatura do óleo térmico na entrada do pré-aquecedor (L15) (Figura 1.15) fica em torno 265 °C, e sua temperatura de saída está entre 255°C. Já a temperatura de saída do agregado em torno de 160°C. O mesmo aquecedor de óleo térmico também aquece os misturadores.



Figura 1.15- Pré- Aquecedor de Coque

### 1.2.2.4 - Mistura

Na Etapa de mistura acontece a adição do piche e rejeito verde (quando usado) ao agregado seco pré-aquecido. Isto se dá no primeiro misturador (L17).

A mistura deve ser eficiente para atingir a uniformidade da pasta e isto é dependente do processo e dos equipamentos. A Albras utiliza misturadores contínuos. Dois misturadores são necessários se o piche usado é sólido, como é o caso da Albras. O primeiro misturador (L17-Figura 1.16) é usado para amolecer e distribuir o piche sobre a superfície do agregado e o segundo misturador (L24) é necessário para assegurar uma mistura adequada. Representação no supervisor Figura 1.17[30].



Figura 1.16- Misturador de Pasta

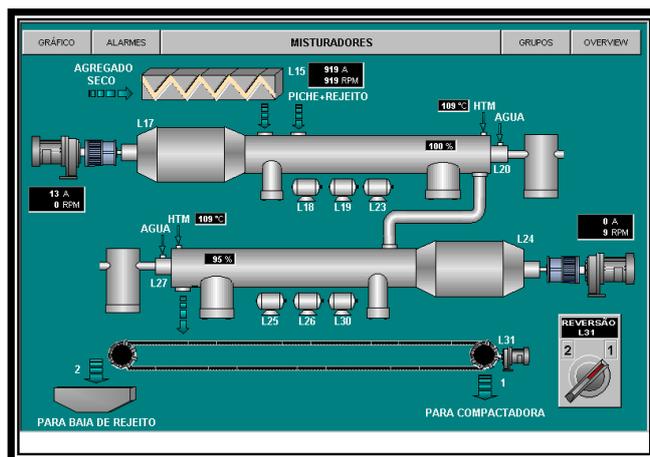


Figura 1.17- Misturador no Supervisório

### 1.2.2.5 - Compactação

Etapa final onde é definida a densidade do Anodo Verde.

A compactadora possui três estágios operacionais:

- ✓ Pesagem ou enchimento do molde: Por meio de uma balança dosadora, a pasta com o “peso desejado” é liberada no molde para a próximo estágio.
  - ✓ Vibro-compactação: Neste estágio a pasta é Vibrada e Compactada (Figuras: 1.21-Sistema Supervisório ) sob vácuo ,que é gerado por meio de uma bomba de vácuo, e esta opera com 800 a 900 mbar.
  - ✓ Ejeção do bloco: Estágio que completa o ciclo da máquina, Consiste em liberar o bloco. Neste estágio é medido a “altura “do anodo.
- Os estágios estão representados abaixo nas Figuras : 1.18; 1.19 e 1.20.



Figura 1.18- Enchimento



Figura 1.19- Vibração



Figura 1.20- Ejeção

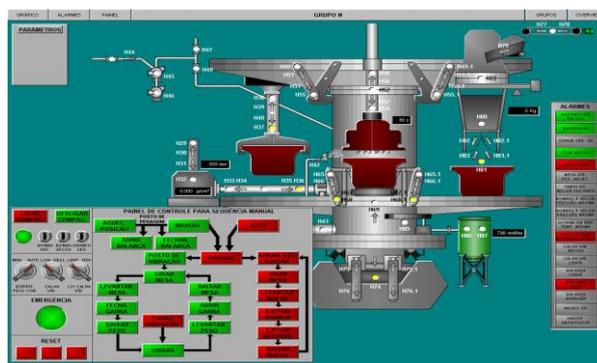


Figura 1.21- Compactador Vibratório Sistema Supervisório

#### 1.2.2.6 - Requisitos para um bom anodo

Dentre as características desejadas para um bom anodo, pode-se citar:

- Alta condutibilidade elétrica;
- Baixa condutibilidade térmica;
- Boa resistência mecânica;
- Boa estabilidade térmica e mecânica;
- Baixo nível de impurezas;
- Baixo nível de porosidades;
- Adequada resistência ao choque térmico;
- Elevada densidade aparente.

#### 1.2.2.7 - Importância da qualidade dos anodos

A qualidade dos anodos influencia diretamente o custo de produção do alumínio.

- ✓ Custo de produção do alumínio:

- Custos de Capital	= 33%
- Alumina	= 24%
- Fabricação de Anodos	= 17%
- Energia	= 13%
- Mão-de-obra	= 7%
- Manutenção	= 6%

O produto final da planta de Anodo Verde, ou seja, o bloco anodico é transportado para fábrica de Cozimento onde receberá a propriedade de rigidez e depois será encaminhado para fábrica de Chumbamento que após a colocação da haste envia para a fábrica de Redução.

## 1.2 – ORGANIZAÇÃO DO TRABALHO

No capítulo 1 contém uma abordagem geral sobre a inteligência artificial, com foco na Lógica Fuzzy a qual será dissertada neste trabalho com aplicação em escala industrial, teremos também uma breve descrição do processo de fabricação de alumínio primário produzido pela Albras, dando uma visão abrangente da indústria situada no município de Barcarena no estado do Pará. O capítulo 2 descreve os fundamentos da Lógica Fuzzy bem como o detalhamento do sistema de inferência fuzzy. Já o capítulo 3 compõe-se de uma apresentação do controle fuzzy relacionado com as variáveis que o mesmo irá processar quando estiver operando. O capítulo 4 trata da implementação do controlador fuzzy para o Aquecedor de Óleo Térmico (HTM), dando uma descrição detalhada do sistema de aquecimento, bem como a inserção da Lógica Fuzzy em linguagem ladder no CLP- 5/80 da Rockwell Automation. No capítulo 5 serão abordados os resultados obtidos de maneira experimental na planta de Anodo Verde da Albras, fazendo um estudo detalhado da resposta do controlador fuzzy aplicado no Aquecedor de Óleo Térmico, também foi feita uma comparação do controlador fuzzy com o controlador PID que antes controlava o sistema de aquecimento. Para finalizar o capítulo 6 faz a conclusão a respeito da aplicação real do controlador fuzzy na planta e comentários sobre posteriores trabalhos futuros que poderão vir a ser implementados na empresa com foco na Lógica Fuzzy.

## CAPÍTULO 2

### 2.1 - FUNDAMENTOS DE LÓGICA FUZZY

#### 2.1.2 – Introdução

A Lógica Fuzzy foi desenvolvida com base na teoria dos conjuntos fuzzy, proposta em meados da década de 60 pelo professor de ciências da computação Lotfi A Zadeh, da Universidade de Berkeley. A elaboração dessa teoria foi motivada, em grande parte, pela convicção de que os métodos tradicionais de análise eram inadequados para descrever sistemas nos quais as variáveis não estivessem relacionadas por equações diferenciais.

Os primeiros trabalhos de aplicação da lógica a controladores foram realizados por Assilian e Mandani em 1974. Esses trabalhos ficaram, quase todos, restritos a montagem em laboratório ou simulação computacional. Durante a década de 80 foram desenvolvidos os primeiros dispositivos usando Lógica Fuzzy em escala industrial.

Operadores humanos são capazes de controlar processos bastante complexos baseado em informações, imprecisas ou aproximadas a respeito desses processos. A estratégia adotada pelos operadores humanos é também de natureza imprecisa e geralmente possível de ser expressa em termos linguísticos. A Teoria de Conjuntos Fuzzy [7] e os Conceitos de Lógica Fuzzy [8] podem ser utilizados para traduzir em termos matemáticos a informação imprecisa expressa por um conjunto de regras linguísticas. Se um operador humano for capaz de articular sua estratégia de ação como um conjunto de regras da forma SE ENTÃO, um algoritmo passível de ser implementado em computador pode ser construído, conforme demonstrado já nos anos 80 [9,11,12]. O resultado é um sistema de inferência baseado em regras, no qual a Teoria de Conjuntos Fuzzy fornece o ferramental matemático para se lidar com as tais regras linguísticas. É importante notar que, nesse tipo de sistema, não é necessário conhecer-se o modelo matemático do processo, o que se constitui em uma vantagem sobre outros procedimentos em que é fundamental a descrição matemática do sistema a controlar.

## 2.2 -CONJUNTOS FUZZY

Nesta seção são apresentados os fundamentos da Teoria de Conjuntos Fuzzy, bem como definições, propriedades e aspectos de Lógica Fuzzy.

### 2.2.1. Fundamentos

Quando se raciocina com base na teoria clássica dos conjuntos, o conceito de pertinência de um elemento a um conjunto fica bem definido. Os elementos de um conjunto  $A$  em um determinado universo  $X$  simplesmente pertencem ou não pertencem àquele conjunto. Isto pode ser expresso pela função característica  $f_A$  :

$$f_A(x) = \begin{cases} 1 & , \text{se, e somente se, } X \in A \\ 0 & , \text{se, e somente se, } X \notin A \end{cases} \quad (2.1)$$

O conceito de nebulosidade (*fuzziness*) é introduzido generalizando-se a função característica de modo que ela possa assumir um número infinito de valores diferentes no intervalo  $[0,1]$ . Assim, um conjunto fuzzy  $A$  em  $X$  é um conjunto de pares ordenados

$$A = \{ \mu_A(x) / x \} \quad x \in X \quad (2.2)$$

onde  $\mu_A(x)$  é a função de pertinência de  $x$  em  $A$  e é definida como o mapeamento de  $X$  no intervalo fechado  $[0,1]$

$$\mu_A(x) : X \rightarrow [0,1] \quad (2.3)$$

A função de pertinência indica o quanto um elemento pertence a um dado conjunto. O **conjunto suporte** de um conjunto fuzzy  $A$  é o conjunto de elementos no universo  $X$  para os quais  $\mu_A(x) > 0$ . Assim, um conjunto fuzzy também pode ser visto como o mapeamento do conjunto suporte no intervalo  $[0,1]$ , o que implica em expressar o conjunto fuzzy por sua função de pertinência. Por exemplo, para o conjunto fuzzy  $A = \{baixo\}$ , no universo dos números positivos menores ou iguais a 100, alguns valores da função de pertinência poderiam ser:

$$\mu_A(0) = \mu_A(5) = \mu_A(10) = \mu_A(15) = \mu_A(20) = 1.0$$

$$\mu_A(25) = 0,9$$

$$\mu_A(30) = 0,8$$

$$\mu_A(45) = 0,1$$

$$\mu_A(50) = \mu_A(55) = \mu_A(60) = \dots = \mu_A(100) = 0$$

Uma **variável fuzzy** é uma variável cujos valores são rótulos (*labels*) de conjuntos fuzzy. Por exemplo, a temperatura de um dado processo poderia ser uma variável fuzzy assumindo **valores** *pequena, média, alta*, etc. Estes valores são descritos por intermédio de conjuntos fuzzy. Generalizando, os valores de uma variável podem ser sentenças em uma linguagem especificada. Neste caso, a variável é uma **variável linguística**. Para ilustrar, os valores da variável fuzzy temperatura poderiam ser expressos como *alta, não alta, muito alta, bastante alta, não muito alta, alta mas não muito alta*. Neste caso, os valores fuzzy são sentenças formadas a partir do rótulo *alta*, da negação *não*, dos conectivos *e mas*, e dos modificadores *muito* e *bastante*. Nesse contexto, a variável temperatura é uma variável linguística.

A principal função das variáveis linguísticas é fornecer uma maneira sistemática de caracterizar fenômenos complexos ou mal definidos, a partir da exploração de informações qualitativas. Em essência, a utilização do tipo de descrição linguística empregada por seres humanos, permite o tratamento de sistemas que são muito complexos para serem analisados através de métodos baseados apenas em informação quantitativa.

A dependência de uma variável linguística em relação a outra (independente) é descrita através de uma declaração condicional fuzzy (*fuzzy conditional statement*), da forma: R: SE  $\Pi^1$  ENTÃO  $\Pi^2$ , onde  $\Pi^1$  e  $\Pi^2$  são frases fuzzy da forma ( $\Pi$  é A) e A é um subconjunto fuzzy de um universo X. Pode ser atribuído a A um significado linguístico que define o valor de X. Por exemplo:

SE *z* é pequeno ENTÃO *w* é muito grande

SE *erro* é grande negativo ENTÃO *saída* é grande positiva.

Duas ou mais declarações condicionais podem ser combinadas, como, por exemplo:

$$R: \text{SE } \Pi^1 \text{ ENTAO } (\text{SE } \Pi^2 \text{ ENTAO } \Pi^3)$$

Várias declarações podem ser combinadas através do conectivo **ou**, formando assim um algoritmo fuzzy (RN) da forma:  $R^N: R^1 \text{ ou } R^2 \text{ ou } R^3 \dots\dots\dots \text{ ou } R^n$

### 2.2.2. Operações com conjuntos fuzzy

Com o intuito de apresentar algumas propriedades e operações importantes com conjuntos fuzzy [11,12], introduz-se agora os dois operadores mais utilizados, e, num estágio seguinte, as operações propriamente ditas.

#### 2.2.2.1. Operadores *max* E *min*

Os operadores *max* e *min* podem ser usados tanto didática como monicamente, dependendo do contexto. Eles podem operar sobre conjuntos ou matrizes, elemento a elemento, assim como sobre elementos singulares. Por exemplo, o mínimo ou o máximo de dois elementos, *a* e *b*, é definido por:

$$\begin{aligned} a \wedge b = \min(a, b) &= \begin{cases} a, \text{ se } a \leq b \\ b, \text{ se } a > b \end{cases} \\ a \vee b = \max(a, b) &= \begin{cases} a, \text{ se } a \geq b \\ b, \text{ se } a < b \end{cases} \end{aligned} \quad (2.4)$$

O mínimo ou máximo de dois conjuntos *A* e *B* resulta num conjunto *C* dado por:

$$\begin{aligned} C = A \wedge B &= \{\min(a, b)\} \forall a \in A, b \in B \\ C = A \vee B &= \{\max(a, b)\} \forall a \in A, b \in B \end{aligned} \quad (2.5)$$

Quando usados monicamente, os operadores implicam no mínimo (infimum) ou máximo (supremum) sobre todos os elementos de um determinado conjunto, como, por exemplo:

$$\begin{aligned} a = \wedge A = \text{Inf}A \quad a \in A \\ a = \vee A = \text{Sup}A \quad a \in A \end{aligned} \quad (2.6)$$

Os operadores podem também ser usados em um sentido funcional, com elementos ou conjuntos:

$$\begin{aligned} a = \wedge(a_1, a_2, \dots, a_m) = a_1 \wedge a_2 \wedge \dots \wedge a_m = \bigwedge_{k=1}^m a_k \\ A = \wedge(A_1, A_2, \dots, A_m) = A_1 \wedge A_2 \wedge \dots \wedge A_m = \bigwedge_{k=1}^m A_k \end{aligned} \quad (2.7)$$

Quando os elementos são funções de outra variável, as operações podem ser expressas como:

$$A = \wedge_x a(x) \quad x \in X \quad (2.8)$$

Finalmente, na manipulação de expressões envolvendo os operadores *min* e *max*, as leis são as mesmas que as leis algébricas de multiplicação e soma, respectivamente.

### 2.2.3. Variáveis linguísticas

Geralmente, o valor de uma variável linguística é um termo composto, resultado da concatenação de outros termos, os quais podem ser divididos em 4 categorias:

- **termos primários:** rótulos de conjuntos fuzzy especificados em um determinado universo (por exemplo: *alto*, *baixo*, *pequeno*, *médio*, *grande*, *zero*).
- a negação **não** e os conectivos **e** e **ou**
- **modificadores**, como *muito*, *pouco*, *levemente*, *positivo*, *negativo*
- **marcadores**, como parênteses.

Termos primários podem apresentar funções de pertinência contínuas, descontínuas ou discretizadas. Funções de pertinência contínuas são definidas por intermédio de funções analíticas. Por exemplo, a seguinte função geral pode ser usada para definir os termos primários *pequeno*, *médio* e *grande*:

$$\mu_A(x) = (1 + (a(x - c))^b)^{-1} \quad (2.18)$$

A forma de  $\mu_A(x)$  pode ser modificada através da manipulação dos três parâmetros  $a$ ,  $b$  e  $c$ . Por exemplo:

$$\begin{aligned} \mu_{pequena}(x) &= (1 + 9x^2)^{-1} \\ \mu_{médio}(x) &= (1 + 9(x - 5)^2)^{-1} \\ \mu_{grande}(x) &= (1 + 9(x - 2)^2)^{-1} \end{aligned} \quad (2.19)$$

Funções de pertinência descontínuas são compostas de segmentos contínuos lineares, resultando em formas triangulares ou trapezoidais. Funções de pertinência discretizadas consistem de conjuntos de valores discretos correspondendo a elementos discretos do universo. Por exemplo, se  $X = \{0,1,2,3,4,5,6\}$  poder-se-ia ter:

$$\begin{aligned} \mu_{pequena}(x) &= \{0,3; 0,7; 1,0; 0,7; 0,3; 0; 0\} \\ \mu_{médio}(x) &= \{0; 0; 0,3; 0,7; 1,0; 0,7; 0,3\} \\ \mu_{grande}(x) &= \{0; 0; 0; 0,3; 0,7; 1,0\} \end{aligned} \quad (2.20)$$

A negação *não* e os conectivos **e** e **ou** podem ser definidos em termos das operações de *complementação*, *intersecção* e *união*, respectivamente. Em geral, o conectivo *e* é usado com variáveis em diferentes universos:

$$\left. \begin{aligned} A &= \{\mu_A(x)/x\} \ x \in X \\ B &= \{\mu_B(y)/y\} \ y \in Y \end{aligned} \right\} \rightarrow A \text{ e } B = \{\mu_A(x) \wedge \mu_B(y)/(x, y)\} = \{\mu_{A \text{ e } B}(x, y)/(x, y)\} \quad (2.21)$$

Por exemplo: vapor é *molhado* e *quente*. Se os valores de  $A$  e  $B$  pertencerem ao mesmo universo, a interpretação semântica invalida a premissa (por exemplo, vapor é *quente* e *frio* não faz sentido). Variáveis no mesmo universo podem ser conectadas por *e* quando elas são negadas (vapor é *não quente* e *não frio*, por exemplo).

O conectivo *ou* conecta valores linguísticos de uma mesma variável, os quais estão no mesmo universo:

$$\left. \begin{aligned} A &= \{\mu_A(x)/x\} \ x \in X \\ B &= \{\mu_B(x)/x\} \end{aligned} \right\} \rightarrow A \text{ ou } B = \{\mu_A(x) \vee \mu_B(x)/(x)\} = \{\mu_{A \text{ ou } B}(x)/(x)\} \quad (2.22)$$

Por exemplo: vapor é *quente* ou *frio*; pressão é *alta* ou *baixa*. Se dois valores linguísticos conectados por *ou* pertencem a variáveis diferentes, a declaração parece ser semanticamente incorreta (por exemplo, vapor é *quente ou molhado*). Entretanto, se *ou* é usado para conectar variáveis em uma sentença do tipo SE . . . ENTÃO . . . , ele pode ser usado com duas variáveis diferentes. Por exemplo:

SE a pressão é *alta* **ou** a velocidade é *rápida* ENTÃO o controle é zero

A operação *não* é considerada como semanticamente sinônima da negação em linguagem natural:

$$A = \{\mu_A(x) / x\} \Rightarrow \text{não } A = \{(1 - \mu_A(x)) / x\} \quad (2.23)$$

Exemplo: pressão é *não alta*.

Modificadores servem para gerar um conjunto maior de valores para uma variável linguística a partir de uma coleção de termos primários. Por exemplo, usando *muito* em conjunto com *não*, e o termo primário *grande*, podem ser gerados os conjuntos fuzzy *muito grande*, *muito muito grande*, *não muito grande*, *grande* e *não muito grande*.

Modificadores são definidos como um operador mônico,  $h$ , que age sobre um conjunto fuzzy  $A$ , para produzir um novo conjunto  $B$ , ou seja  $B = h A$ , onde  $A$  e  $B$  estão no mesmo universo. Modificadores *negativo* e *positivo* são utilizados, geralmente em conjunto, quando  $x$  pode assumir tanto valores negativos como positivos.

#### 2.2.4. Funções de implicação

Uma declaração de implicação fuzzy ou declaração condicional fuzzy descreve uma **relação** entre variáveis linguísticas. Consideremos dois conjuntos fuzzy  $A$  e  $B$ , representando valores linguísticos nos universos  $X$  e  $Y$ , respectivamente. Uma declaração condicional fuzzy da forma SE  $A$  ENTÃO  $B$  é definida matematicamente como:

$$R: \text{SE } A \text{ ENTÃO } B = A \rightarrow B = A \times B \quad (2.24)$$

A **relação fuzzy**  $AxB$  denota, neste caso, a implicação  $A \rightarrow B$  no produto cartesiano dos dois universos  $XxY = \{(x, y)\} \quad x \in X, y \in Y$ .

A função de pertinência),  $\mu_R(x, Y)$  que define a *implicação* pode ser obtida através das funções de pertinência individuais  $\mu_A(x)$  e  $\mu_B(y)$  de inúmeras maneiras [8,9,10,11,12]. Assim, denotando o operador de implicação por  $f_1$ :

$$\mu_R(x, y) = f_1(\mu_A(x), \mu_B(y)) \quad x \in X, y \in Y \quad R = \{\mu_R(x, y)/(x, y)\} \quad (2.25)$$

Generalizando, se  $A^1, A^2, \dots, A^n$  são conjuntos fuzzy em  $X$ , e  $B^1, B^2, \dots, B^n$  são conjuntos fuzzy em  $Y$ , um algoritmo fuzzy é:

$R^N$ : SE  $A^1$  ENTÃO  $B^1$

**ou**

SE  $A^2$  ENTÃO  $B^2$

**ou**

SE  $A^n$  ENTÃO  $B^n$

Neste caso, o conectivo **ou** (*else*) serve para conectar sentenças linguísticas. Sua implementação, denotada por  $f_2$ , depende da função de implicação  $f_1$  empregada. Assim, para o algoritmo fuzzy tem-se:

$$\begin{aligned} \mu_R^N(x, y) &= f_2(\mu_{R^1}(x, y), \mu_{R^2}(x, y), \dots, \mu_{R^n}(x, y)) \\ &= f_2[f_1(\mu_{A^1}(x), \mu_{B^1}(y)), f_1(\mu_{A^2}(x), \mu_{B^2}(y)), \dots, f_1(\mu_{A^n}(x), \mu_{B^n}(y))] \end{aligned} \quad (2.26)$$

As equações acima foram desenvolvidas para variáveis simples  $A$  e  $B$ , mas podem ser estendidas para casos de mais variáveis. Normalmente as variáveis **antecedentes** são combinadas através do conectivo **e**, implementado pelos operadores **min** ou **produto**. As **funções de implicação** mais utilizadas em aplicações de engenharia são **min** e **produto**. Para sentenças com um *antecedente* apenas (do tipo SE  $A$  ENTÃO  $B$ ), tem-se:

$$\begin{aligned} \text{min :} \quad & \mu_R(x, y) = \min(\mu_A(x), \mu_B(y)) = \mu_A(x) \wedge \mu_B(y) \\ \text{produto:} \quad & \mu_R(x, y) = \mu_A(x) \cdot \mu_B(y) \end{aligned} \quad (2.27)$$

Em ambos os casos acima, a combinação de  $n$  declarações condicionais fuzzy é efetuada através do conectivo **ou**, implementado por **max**. Para a implicação **min**, por exemplo, tem-se:

$$\mu_{R^n}(x, y) = \bigvee_{j=1}^n (\mu_{A^j}(x) \wedge \mu_{B^j}(y)) \quad (2.28)$$

Muitas outras funções de implicação têm sido propostas na literatura, conforme pode ser verificado nas referências.

### 2.2.5. Regra de inferência composicional

Duas relações fuzzy

$R^1$ : SE  $A$  ENTÃO  $B$

$R^2$ : SE  $B$  ENTÃO  $C$

podem ser compostas de forma a resultar em  $R^{12}$  : SE  $A$  ENTÃO  $C$ .

A composição  $R^{12} = R^1 \circ R^2$  pode ser definida ou por uma regra do tipo **max-min**:

$$\mu_{R^{12}}(x, z) = \bigvee_y (\mu_{R^1}(x, y) \wedge \mu_{R^2}(y, z)) \quad (2.29)$$

ou por uma do tipo **max-produto**:  $\mu_{R^{12}}(x, z) = \bigvee_y (\mu_{R^1}(x, y) \cdot \mu_{R^2}(y, z))$

Quando conjuntos fuzzy discretos são empregados, as operações acima são equivalentes ao produto interno de duas matrizes, com a multiplicação e a soma substituídas pelas operações *min* (no primeiro caso) e *max*, respectivamente. É razoável que se use uma das duas definições acima em correspondência com a função de implicação adotada. Assim, com as funções de implicação envolvendo somente operadores *max* e *min*, emprega-se a regra composicional *max-min*. No caso de implicações que envolvam operadores aritméticos, pode-se usar a regra *max-produto*.

A questão que se coloca agora é: dada uma relação entre 2 variáveis fuzzy, qual o conseqüente para um dado antecedente? Considerem-se as duas

$$\begin{aligned} \text{variáveis:} \quad A &= \{\mu_A(x) / x\} \quad x \in X \\ B &= \{\mu_b(y) / y\} \quad y \in Y \end{aligned} \quad (2.30)$$

e a relação entre elas  $R = \{\mu_R(x, y)/(x, y)\} \quad x \in X, y \in Y$  (2.31)

Se o antecedente  $A'$  for dado por  $A' = \{\mu_{A'}(x)/x\} \quad x \in X$ , o conseqüente  $B$ , inferido através da regra de inferência *max-min*, será [2]:

$$B' = A' \circ R = \left\{ \bigvee_x (\mu_{A'}(x) \wedge \mu_R(x, y)) / y \right\} \quad x \in X, y \in Y$$
 (2.32)

### 2.2.6. Sistema de inferência fuzzy

Com base nas informações apresentadas, é possível construir o Sistema de Inferência Fuzzy mostrado na Figura 2.1, onde estão identificadas as funções de cada bloco.

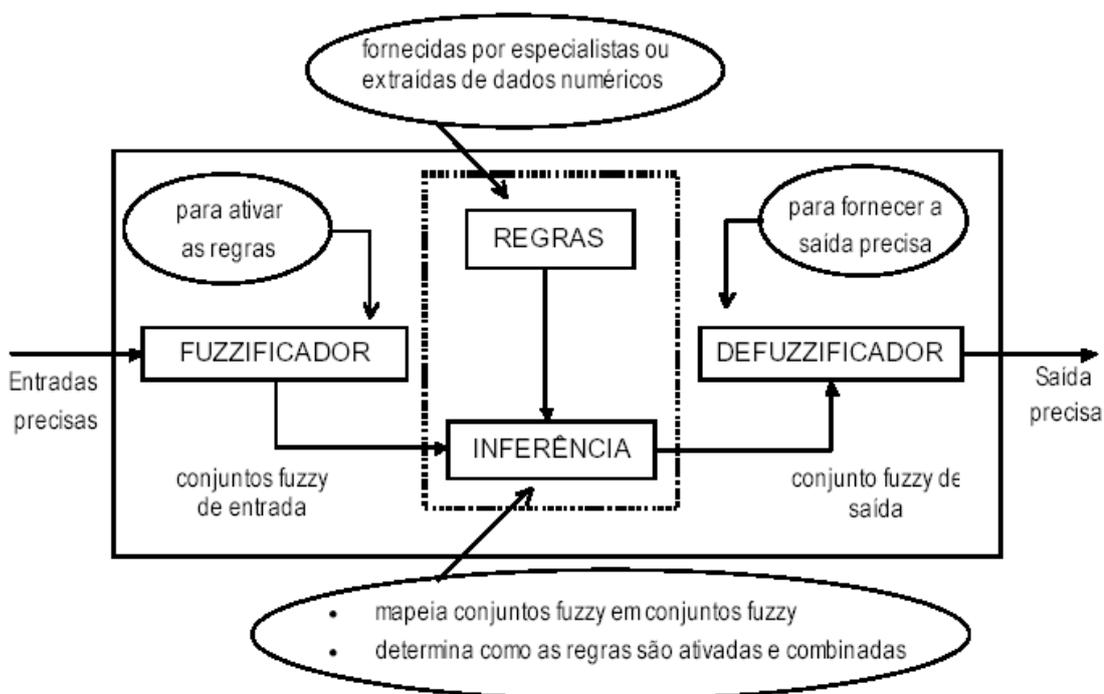


Figura 2.1 – Sistema de inferência fuzzy.

Como geralmente os dados de entrada são valores precisos, resultados de medições ou observações (conjuntos de dados, por exemplo), é necessário efetuar-se um *mapeamento* destes dados precisos para os conjuntos fuzzy de entrada relevantes, o que é realizado no estágio de **fuzzificação**. Neste estágio ocorre também a *ativação* das regras relevantes para uma dada situação. Uma vez obtido o conjunto fuzzy de saída através do processo de inferência, no estágio de **defuzzificação** é efetuada uma *interpretação* dessa informação. No

estágio de **inferência** ocorrem as operações com conjuntos fuzzy propriamente ditas: combinação dos *antecedentes* das regras, *implicação* e *regra de inferência composicional*. Os conjuntos fuzzy de entrada, relativos aos antecedentes das regras, e o de saída, referente ao conseqüente, podem ser definidos previamente ou, alternativamente, gerados automaticamente a partir dos dados.

### 2.3 – CONCLUSÃO

Neste capítulo foram abordados os fundamentos da teoria de conjuntos fuzzy e Lógica Fuzzy. Os conjuntos e Lógica Fuzzy provém a base para a geração de técnicas poderosas para a solução de problemas, com uma vasta aplicabilidade, especialmente, nas áreas de controle e tomada de decisão.

No próximo capítulo será abordado a teoria de controle fuzzy, apresentando todas as etapas de implementação, bem como uma análise da contribuição de cada regra de controle, influência das funções de pertinência e funções de implicação.

## CAPÍTULO 3

### 3.1 - CONTROLE FUZZY

#### 3.1.1 -Introdução

Em um controlador fuzzy baseado em regras, a estratégia de controle é descrita por intermédio de regras linguísticas que conectam, de modo impreciso, várias situações com as ações a serem tomadas. As regras são expressas por declarações condicionais do tipo SE..... ENTÃO e implementadas como implicações lógicas através da utilização das propriedades e técnicas abordadas no capítulo a anterior. A estrutura básica de um controlador deste tipo pode ser vista na Figura 3.1.

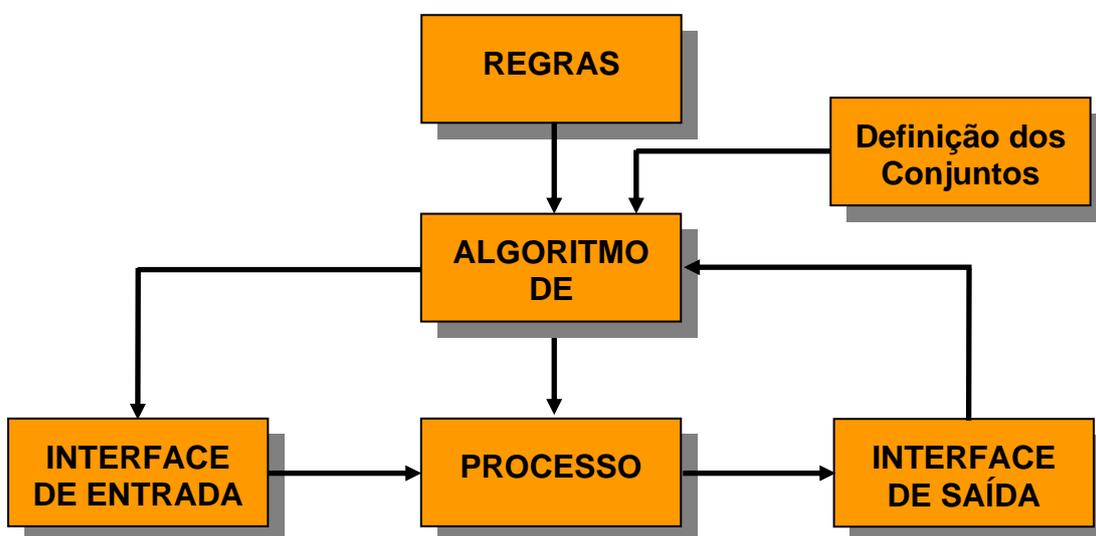


Fig.3.1 – Estrutura básica de um controlador fuzzy

A **interface de saída** adquire informações (determinísticas) a respeito do processo e as traduz para a linguagem de conjuntos fuzzy. Nesse, podem estar incluídos conversores A/D e D/A, fatores de escala, procedimentos de quantização, etc., conforme será descrito mais adiante. A **interface de entrada** converte a saída fuzzy do controlador para um valor determinístico a ser fornecido ao processo. Nesse processo se incluem métodos de defuzzificação, fatores de escala, integradores, conversores, etc. Também faz parte da estrutura a *definição dos conjuntos fuzzy* usados para representar as regras, incluindo-se aí também as definições dos universos usados para as variáveis de entrada e saída do

controlador. O algoritmo de controle usa as regras disponíveis e produz, para um determinado estado do processo, a decisão sobre a próxima entrada do processo. A estrutura apresentada é de caráter geral e aplicável a qualquer processo controlável por essas técnicas. O que fornece ao controlador uma identidade é o conjunto de regras que lhe são fornecidas; estas constituem a quantidade de informação necessária para executar uma determinada tarefa de controle com o processo em questão. Isto é importante de ser enfatizado pois a essência do controlador reside nas regras de controle; a teoria de conjuntos fuzzy apenas fornece os meios para se traduzir as regras em termos matemáticos e para se inferir uma decisão a partir delas.

### 3.2. INTERFACE COM O PROCESSO

Nesta seção considera-se a definição das variáveis de controle e itens como escalonamento, quantização e métodos de defuzzificação, necessários para se estabelecer a conexão do controlador fuzzy com um processo não-fuzzy.

#### 3.2.1. Variáveis de Controle

Em controladores fuzzy de caráter geral como o apresentado aqui as variáveis de entrada são o **erro**, gerado a partir da diferença entre o sinal de referência e a saída do processo, e a **variação do erro**, normalmente gerada a partir do erro. A variável de saída do controlador é a **variação no controle**. A opção por uma *saída incremental*, ao invés de *absoluta*, é mais condizente com o raciocínio empregado por operadores humanos e, além disso, proporciona uma economia em termos do universo da saída.

Estabelecendo uma ligação com as seções anteriores, as variáveis fuzzy *erro* e *variação do erro* são subconjuntos fuzzy em seus respectivos universos. Elas poderiam ser definidas, por exemplo, como correspondentes aos valores reais medidos das entradas. Entretanto, um procedimento mais geral faz uso de universos discretos e normalizados, definidos como, por exemplo:

$$\{-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6\} \quad (3.1)$$

Nos controladores fuzzy do tipo do aqui considerado, os conjuntos fuzzy – correspondentes às variáveis fuzzy – são definidos por conjuntos suporte de forma triangular, simétricos, com valores de pertinência  $\{0,3 \ 0,7 \ 1,0 \ 0,7 \ 0,3\}$ . Na Tabela 2 a seguir estão representados 7 conjuntos fuzzy – denominados PG (Positivo Grande), PM (positivo Médio), PP (Positivo Pequeno), ZE (Zero), NG (Negative Grande), NM (Negative Médio), NP (Negativo Pequeno) – e os valores de pertinência associados a cada um desses conjuntos, considerando um universo discreto e finito de 13 elementos conforme definido acima.

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
PG	0	0	0	0	0	0	0	0	0	0	0,3	0,7	1,0
PM	0	0	0	0	0	0	0	0	0,3	0,7	1,0	0,7	0,3
PP	0	0	0	0	0	0	0,3	0,7	1,0	0,7	0,3	0	0
ZE	0	0	0	0	0,3	0,7	1,0	0,7	0,3	0	0	0	0
NP	0	0	0,3	0,7	1,0	0,7	0,3	0	0	0	0	0	0
NM	0,3	0,7	1,0	0,7	0,3	0	0	0	0	0	0	0	0
NG	1,0	0,7	0,3	0	0	0	0	0	0	0	0	0	0

Tabela 2- Universo

Alternativamente pode ser empregada uma representação gráfica, conforme a figura a seguir:

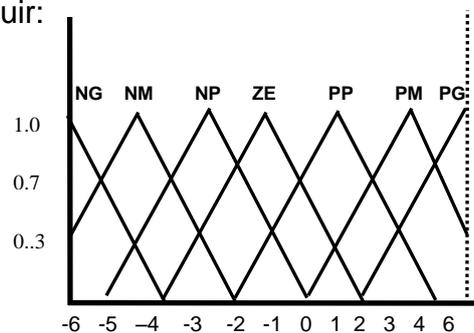


Fig.3.2 – Representação Gráfica do Universo

### 3.2.2. Escalonamento e Quantização

Conforme foi tratado na seção anterior, os universos aqui considerados para as variáveis fuzzy são: discretos, finitos e normalizados. Assim, torna-se necessário utilizar fatores de escala para fazer corresponder, aos valores reais e medidos do erro e variação do erro, valores normalizados. Além disto, deve ser

efetuada uma quantização, resultando nas seguintes entradas para o controlador (em um instante  $i$ , por exemplo):

$$\begin{aligned} e_i &= (G_e \times \text{erro})_{\text{quantizado}} \\ ce_i &= (CG_e \times \text{variação do erro})_{\text{quantizado}} \end{aligned} \quad (3.3)$$

onde  $GE$  e  $GCE$  são fatores de escala. O procedimento de escalonamento e quantização está representado na figura abaixo, onde, como um exemplo,  $GE$  ( $GCE$ ) = 0,5:

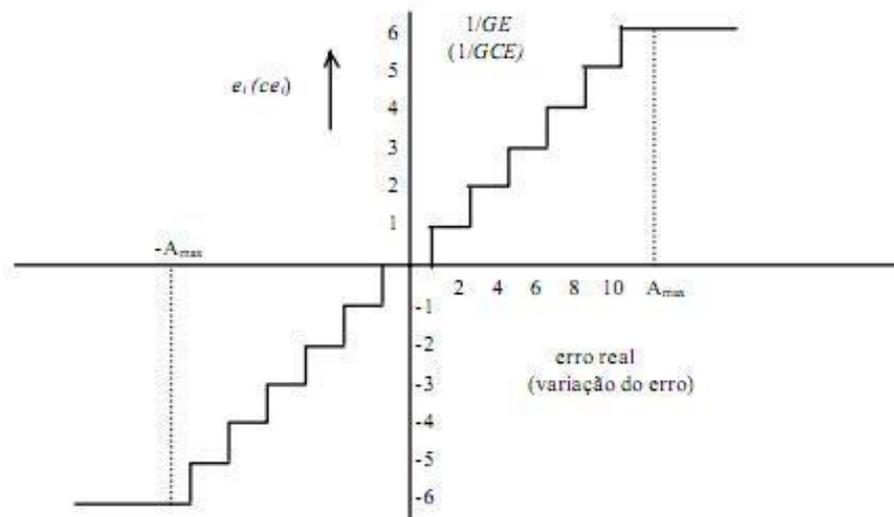


Fig.3.3 – Procedimento de Quantização

Os fatores de escala  $GE$  e  $GCE$  determinam a forma da característica mostrada na figura. Um aumento em  $GE$ , por exemplo, significa que um número menor de valores reais poderá ser mapeado para o universo escolhido. A tolerância do erro é dada por  $1/2GE$ . Além disso, valores que caírem fora da região entre  $A_{\max}$  e  $-A_{\max}$  serão mapeados para  $+6$  e  $-6$ , respectivamente. Isto significa que o controle será mais preciso na *janela* delimitada por aqueles valores. A escolha dos fatores de escala é parte do procedimento de sintonia do controlador e estes podem ser expressos por constantes ou por funções (do erro, por exemplo). Se o universo da saída do controlador também for normalizado, a exemplo das variáveis de entrada, um fator de escala  $GO$  é usado, após a

defuzzificação (cf. seção seguinte), para mapear valores quantizados para valores reais compatíveis com a entrada do processo. Esse fator de escala também deve ser ajustado de acordo com o comportamento da resposta.

### 3.2.3. Métodos de Defuzzificação

A saída do controlador fuzzy é um subconjunto fuzzy  $U_i$  do universo da saída. Como o processo requer um sinal não-fuzzy em sua entrada, deve-se fazer uma interpretação daquele conjunto fuzzy. Esta interpretação, denominada **defuzzificação**, pode ser feita através de vários métodos, mas os mais consagrados são média dos máximos e centro de gravidade. No primeiro, a saída determinística (não fuzzy), denominada  $u_{MOM}$  é obtida tomando-se a média entre os dois elementos extremos no universo que correspondem aos maiores valores da função de pertinência.

Com o centro de gravidade, a saída  $u_{COG}$  é o valor no universo que divide a área sob a curva da função de pertinência em duas partes iguais. Quando os universos utilizados são discretos, o resultado da defuzzificação tem de ser arredondado para o valor inteiro mais próximo no universo. O arredondamento para o valor inteiro imediatamente abaixo impossibilitará que se obtenha o valor mais alto do universo quando o conjunto suporte está localizado no extremo do universo – {0,3; 0,7; 1}, por exemplo. Na realidade, com o centro de gravidade nunca se consegue obter os valores extremos do universo, devido a própria natureza do método. Isto pode dar origem a uma resposta mais lenta do que a obtida com a média dos máximos.

## 3.3. EXPRESSÃO DE REGRAS DE CONTROLE EM TERMOS MATEMÁTICOS

Embora a terminologia a seguir seja voltada para sistemas de controle, o desenvolvimento apresentado é totalmente válido para qualquer *sistema de inferência fuzzy* com entradas determinísticas. A estratégia de controle é descrita por um conjunto de declarações condicionais ou regras linguísticas. Conforme mencionado em seções anteriores, duas entradas são consideradas para cada

saída: o erro  $E$ , e a *variação do erro*  $CE$ . A saída é a *variação na saída do controlador*  $U$ . O conjunto de regras é, então, da forma:

$R^N$ :            SE  $E = E^1$  e  $CE = CE^1$  ENTÃO  $U = U^1$

**ou**

SE  $E = E^2$  e  $CE = CE^2$  ENTÃO  $U = U^2$

**ou**

.

.

.

**ou**

SE  $E = E^n$  e  $CE = CE^n$  ENTÃO  $U = U^n$

onde  $E^j$ ,  $CE^j$  e  $U^j$  são subconjuntos fuzzy dos universos  $E = \{e\}$ ,  $CE = \{ce\}$  e  $U = \{u\}$ .

Cada regra  $R^j$  pode ser expressa, através de sua função de pertinência, como:

$$\mu_{R^j}(e, ce, u) = f_1[\mu_{E^j}(e) \wedge \mu_{CE^j}(ce), \mu_{U^j}(u)] \quad (3.4)$$

A combinação de todas as  $n$  regras é expressa matematicamente por:

$$\mu_{R^N}(e, ce, u) = f_2[f_1(\mu_{E^1}(e) \wedge \mu_{CE^1}(ce), \mu_{U^1}(u)), f_1(\mu_{E^2}(e) \wedge \mu_{CE^2}(ce), \mu_{U^2}(u)), \dots, \dots f_1(\mu_{E^n}(e) \wedge \mu_{CE^n}(ce), \mu_{U^n}(u))] \quad (3.5)$$

Desta forma, a estratégia de controle é representada por uma *matriz*  $\mu_{R^N}$ . O controlador é então solicitado a fornecer, a partir do valores do *erro* e da *variação do erro*, e utilizando-se da *regra de inferência composicional*, uma saída apropriada. Assim, em um instante  $i$ , quando as entradas determinísticas são  $e_i$  e  $ce_i$ , o conjunto fuzzy da saída será dado por:

$$U_i = (E_i \wedge CE_i) \circ R^N \quad (3.6)$$

onde  $E_i$  e  $CE_i$  são os subconjuntos fuzzy das entradas e  $U_i$  é o subconjunto fuzzy da saída. Usando a *regra de inferência max-min*, tem-se:

$$\mu_{U_i}(u) = \bigvee_e \bigvee_{ce} [\mu_{E_i}(e) \wedge \mu_{CE_i}(ce) \wedge (\mu_{R^N}(e, ce, u))] \quad (3.7)$$

Os cálculos podem ser simplificados se considerarmos que os subconjuntos fuzzy das entradas consistem de valores reais não-fuzzy (*singletons*), isto é:

$$\begin{aligned} \mu_{E_i}(e) &= 1 \text{ se } e = e_i & \mu_{E_i}(e) &= 0 \text{ se } e \neq e_i \\ \mu_{CE_i}(ce) &= 1 \text{ se } ce = ce_i & \mu_{CE_i}(ce) &= 0 \text{ se } ce \neq ce_i \end{aligned} \quad (3.8)$$

Com isto,

$$\begin{aligned} \mu_{U_i}(u) &= \bigvee_e \bigvee_{ce} [1 \wedge 1 \wedge (\mu_{R^N}(e, ce, u))] \text{ para } e = e_i \text{ e } ce = ce_i \\ &= \bigvee_e \bigvee_{ce} [0 \wedge 0 \wedge (\mu_{R^N}(e, ce, u))] \text{ para } e \neq e_i \text{ e } ce \neq ce_i \end{aligned} \quad (3.9)$$

Ou seja:

$$\begin{aligned} \mu_{U_i}(u) &= \mu_{R^N}(e, ce, u) \text{ para } e = e_i \\ &= 0 \text{ nos outros casos} \end{aligned} \quad (3.10)$$

Portanto, o conjunto fuzzy que representa a saída do controlador é dado por:

$$\begin{aligned} \mu_{R^N}(e, ce, u) &= f_2[f_1(\mu_{E^1}(e_1) \wedge \mu_{CE^1}(ce_1), \mu_{U^1}(u)), f_1(\mu_{E^2}(e_2) \wedge \mu_{CE^2}(ce_2), \mu_{U^2}(u)), \\ &\dots, f_1(\mu_{E^n}(e_n) \wedge \mu_{CE^n}(ce_n), \mu_{U^n}(u))] \end{aligned} \quad (3.11)$$

Exemplo: controlador fuzzy com duas entradas: *erro* e, *variação do erro* e uma saída. Dados:

- 13 conjuntos fuzzy para cada variável, todos eles "triangulares", simétricos, definidos em universos discretos e finitos [-6,+6] segundo as funções de pertinência a seguir:

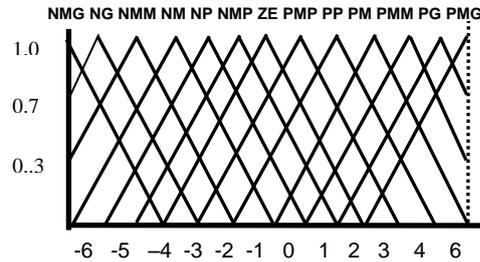


Fig.3.4 – Representação Gráfica do Universo de 13 Predicados

- ⇒ concatenação dos antecedentes (conectivo **e**): **min**
- ⇒ implicação (operador  $f_1$ ): **min**
- ⇒ conectivo **ou** (operador  $f_2$ ): **max**
- ⇒ regras: SE  $E = NP$  e  $CE = PP$  ENTÃO  $U = NP$   
SE  $E = ZE$  e  $CE = NMP$  ENTÃO  $U = ZE$
- ⇒ entradas (após escalonamento e quantização):  $e_i = -1$  e  $ce_i = 0$

$$\left. \begin{array}{l} \mu_{NS}(e_i) = 0,7 \\ \mu_{PS}(ce_i) = 0,3 \end{array} \right\} \Rightarrow \mu_{NS}(e_i) \wedge \mu_{PS}(ce_i) = 0,3 \\
 \left. \begin{array}{l} \mu_{ZO}(e_i) = 0,7 \\ \mu_{NVS}(ce_i) = 0,7 \end{array} \right\} \Rightarrow \mu_{ZO}(e_i) \wedge \mu_{NVS}(ce_i) = 0,7$$

$$\Rightarrow \mu_{U_i}(u) = \vee [0,3 \wedge \mu_{NS}(u); 0,7 \wedge \mu_{ZO}(u)]$$

Ou seja, faz-se a **união** de:

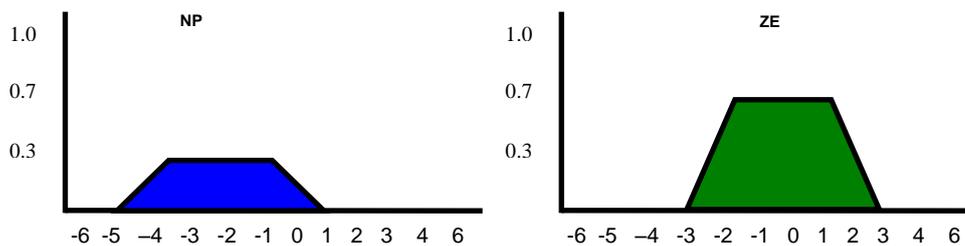


Fig.3.5 – Representação Gráfica do Predicados NP e ZE

Conjunto fuzzy da saída do controlador:

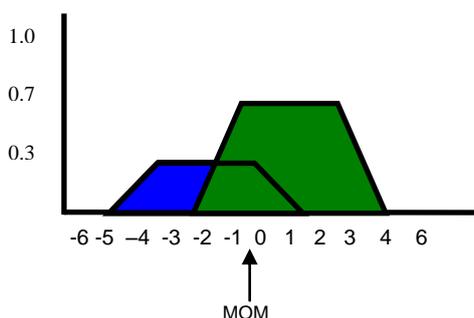


Fig.3.6 – Representação Gráfica da saída do controlador

A **defuzzificação** pela média dos máximos, por exemplo, proporciona uma saída determinística  $u_i = 0$

*Observação:*

Como, neste exemplo, há 13 conjuntos fuzzy para o *erro* e 13 para a *variação do erro*, o máximo número possível de regras seria 169 ( $13 \times 13$ ). Considerou-se, para fins do exemplo, a existência de apenas 2 regras. Alternativamente, poder-se-ia considerar que, das  $n$  ( $\leq 169$ ) regras existentes, somente as duas explicitadas são as ativadas pelo *erro* e *variação do erro* no instante  $i$  considerado.

### 3.4. ASPECTOS DE IMPLEMENTAÇÃO

Nesta seção é apresentada uma maneira simples de se tratar as regras, com vistas à implementação do algoritmo de controle. São também comentados aspectos de ordem prática que devem ser levados em consideração quando da implementação e utilização do algoritmo.

#### 3.4.1. Armazenamento de Regras

O exemplo visto na seção anterior pode ser resolvido de uma maneira mais eficiente se o espaço de estado  $E \times CE$  for representado por uma *grade* e se for

definido o *centro da regra* ( $e_R, ce_R$ ) nesse espaço como o ponto onde  $\mu_{E^j}(e) = \mu_{CE^j}(ce) = 1$ . Com os universos e valores de pertinência usados no exemplo acima, a região de influência de cada regra é determinada por um quadrado de 5 unidades de lado e seu grau de influência dentro deste quadrado é representado por:

$$\begin{array}{ccccc}
 & & ce_R & & \\
 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3 \\
 & 0,3 & 0,7 & 0,7 & 0,7 & 0,3 \\
 e_R & 0,3 & 0,7 & 1 & 0,7 & 0,3 \\
 & 0,3 & 0,7 & 0,7 & 0,7 & 0,3 \\
 & 0,3 & 0,3 & 0,3 & 0,3 & 0,3
 \end{array} \quad (3.12)$$

Com isto, é possível utilizar um simples número para representar o conjunto fuzzy associado ao valor lingüístico de uma variável em uma regra. A ação do controlador pode também ser representada da mesma forma; os valores numéricos na grade representam a posição no universo em que o valor de pertinência é 1. Assim, o armazenamento de regras torna-se bastante simples; conhecidos o conjunto suporte e o centro da regra, torna-se fácil *recuperar* os conjuntos fuzzy correspondentes.

A expressão  $\mu_{E^j}(e_i) \wedge \mu_{CE^j}(ce_i)$ , que determina a *região de influência* de uma regra, descreve o quanto as entradas do controlador são compatíveis com os *antecedentes* de uma regra  $j$ . Esta expressão é normalmente chamada de **degree of fulfilment** da regra  $j$  no instante  $i$ , ou  $DOF_i^j$ .

Se o operador **produto** fosse usado para  $DOF$  (em outras palavras, o conectivo **e** de concatenação dos antecedentes), a distribuição em torno de ( $e_R, ce_R$ ) seria:

$$\begin{array}{ccccc}
 & & ce_R & & \\
 & 0,09 & 0,21 & 0,3 & 0,21 & 0,09 \\
 & 0,21 & 0,49 & 0,7 & 0,49 & 0,21 \\
 e_R & 0,3 & 0,7 & 1 & 0,7 & 0,3 \\
 & 0,21 & 0,49 & 0,7 & 0,49 & 0,21 \\
 & 0,09 & 0,21 & 0,3 & 0,21 & 0,09
 \end{array} \quad (3.13)$$

Como valores pequenos de  $DOF$  são obtidos com este operador, um procedimento razoável consiste de se estabelecer um limiar abaixo do qual a contribuição da regra é desconsiderada. Para um limiar de 0,3, por exemplo, a região de influência de cada regra excluiria os pontos onde  $DOF = 0,09$  e  $DOF = 0,21$ . Voltando ao exemplo, a posição no espaço de estado no instante  $i$ , dada por  $(e_i, ce_i)$ , e as duas regras, especificadas pelos seus centros  $(e_{R_1}, ce_{R_1})$  e  $(e_{R_2}, ce_{R_2})$ , estão representadas na Figura 3.7 :

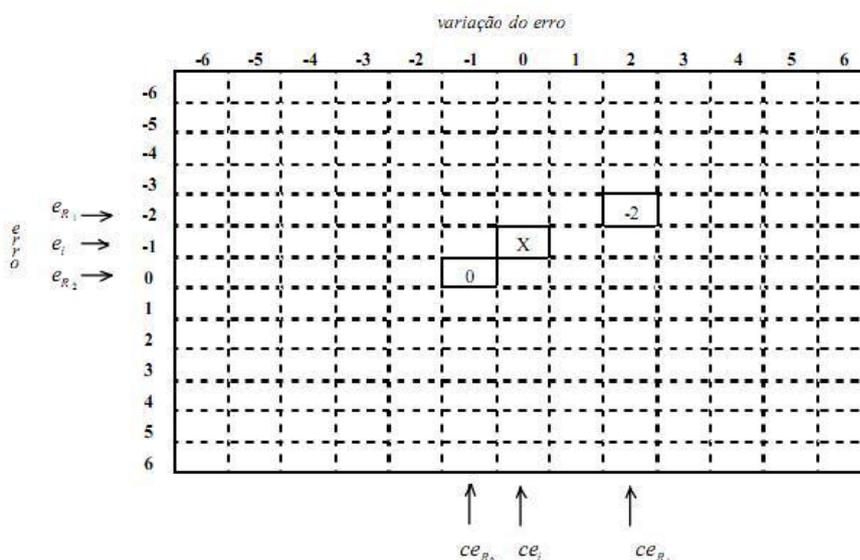


Fig.3.7 – Representação do Espaço de Estado do Erro e Variação Erro

A regra centrada em  $(-2,2)$  contribui menos do que a centrada em  $(0,-1)$  para o cálculo da ação a ser tomada pelo controlador quando o estado presente é  $(-1,0)$ . Isto foi indicado pelos valores  $0,3\Lambda\mu_{NP}(u)$  e  $0,7\Lambda\mu_{ZE}(u)$  obtidos na expressão para  $\mu_{U_i}(u)$ .

### 3.4.2. Contribuição de cada Regra

Com base no que foi visto, verifica-se que, para um ponto  $(e_i, ce_i)$  no espaço de estado, as regras que contribuem para o cálculo da ação do controlador são aquelas cujos centros estão a uma certa distância de  $(e_i, ce_i)$ . Se conjuntos suporte de tamanhos diferentes forem empregados, os quadrados tornam-se paralelogramos e a distância de uma regra do ponto  $(e_i, ce_i)$  não será

uniforme em todas as direções, tornando a implementação menos simples. Dependendo do método de defuzzificação empregado, menos regras podem contribuir para o cálculo da ação do controlador. Se COG for utilizado, todas as regras dentro de uma certa distância contribuem e devem ser incluídas nos cálculos. As regras mais próximas de  $(e_i, ce_i)$  terão uma contribuição maior do que as mais distantes, evidentemente. No entanto, se MOM for empregado, somente as regras mais próximas de  $(e_i, ce_i)$  contribuirão, e apenas elas precisam ser incluídas nos cálculos. Isto se deve ao fato de que, no método MOM, somente os picos no conjunto fuzzy da saída são relevantes. Esse método efetivamente seleciona as regras com maior *DOF*, ou seja, aquelas mais próximas de  $(e_i, ce_i)$ . Em virtude desta característica, é possível criar algoritmos muito simples em que conjuntos fuzzy não são usados explicitamente no cálculo da ação do controlador.

### 3.4.3. Influência das Funções de Pertinência

A região de influência de uma regra dependerá da abrangência, em termos do universo considerado, dos conjuntos fuzzy das entradas. Portanto, o tamanho do conjunto suporte certamente influi no cálculo da ação de controle. No caso de conjuntos com a mesma abrangência, mas com funções de pertinência de diferentes formas, a forma dos conjuntos dos antecedentes de cada regra afeta a determinação de *DOF*, o que terá um efeito no conseqüente (ou saída) de cada regra, e no resultado final. A forma do conjunto do conseqüente de cada regra afeta o conjunto fuzzy da saída diretamente. No entanto, a saída determinística do controlador dependerá do método de fuzzificação. Com MOM, a forma dos conjuntos dos antecedentes não tem importância, desde que sejam simétricos e de forma aproximadamente *triangular*. Só é necessário saber (através de *DOF*) a distância da regra de  $(e_i, ce_i)$ ; as magnitudes absolutas dos valores da função de pertinência não são importantes; apenas as magnitudes em relação às de outra regra interessam. Similarmente, a forma dos conjuntos do conseqüente também não é importante. No caso do método COG, a forma dos conjuntos suporte é importante mesmo quando há simetria. A área sob a curva é afetada diretamente pelo conseqüente de cada regra e indiretamente por *DOF*.

#### 3.4.4. Influência das Funções de Implicação

Diversas funções de implicação têm sido propostas na literatura, na maioria dos casos levando em consideração aspectos de *lógica* [12,13,14,15]. Discussões sobre o uso de diferentes funções de implicação e sua eventual vinculação com métodos de defuzzificação podem ser encontradas em [16]. Do ponto de vista prático (ou de *engenharia*), as funções mais adequadas são as já citadas ***min*** e ***produto***. Este aspecto é exaustivamente explorado em [17], cuja leitura é fortemente recomendada.

#### 3.4.5. Espaço de Fase

Um modo interessante de compreender um controlador fuzzy consiste na análise da correspondência entre as suas regras e o espaço de fase para o fenômeno sob controle. O espaço de fase nos permite ter uma visão sobre a dinâmica do sistema [5]. Seja um típico um controlador PI, com todas as variáveis linguísticas (erro  $e$ , variação do erro  $\Delta e$ , variação no sinal de controle  $\Delta u$ ) assumindo os sete valores habituais {NG -negativo grande; NM negativo médio; NP negativo pequeno; ZE -zero; PP -positivo pequeno; PM -positivo médio; PG -positivo grande}. O espaço de fase desse controlador pode ser visto na Figura 3.7. Para grandes erros e variações positivas do erro (o erro está a crescer, e o ponto de funcionamento está a divergir da referência) é preciso aplicar uma ação de controle negativo grande (NG) para inverter a situação. No extremo oposto verifica-se igualmente uma situação de divergência (o erro é grande e negativo, e está a aumentar negativamente), pelo que é necessário uma forte ação de controle positivo (PG) para que o ponto de funcionamento retome uma aproximação ao ponto especificado. Assim, o primeiro e terceiro quadrantes são zonas característica de tendência para a instabilidade. Pelo contrário, o segundo e quarto quadrantes correspondem genericamente a situações em que o sistema é tendencialmente estável e está a reduzir erro em relação ao ponto especificado de funcionamento. No segundo quadrante o erro é negativo mas varia positivamente (está a diminuir de amplitude, portanto); no quarto quadrante o erro é positivo mas está a diminuir (variação negativa). Portanto, a magnitude do controle a aplicar será pequena. Sobre a diagonal que cruza o segundo e quarto

quadrantes pode-se então escrever sob a forma de uma regra, se o erro é *NG* e a *variação do erro* é *PG* então a *variação no controle* deve ser *ZE*. No extremo superior do primeiro quadrante, pelo contrário. Tem-se uma regra do tipo se o erro é *PG* e a *variação do erro* é *PG* então a *variação no controle* deve ser *NG*. Portanto, de uma forma gradual, entre aquela diagonal e o canto superior direito, o consequente da regra aplicável deverá em princípio variar de *ZE* a *NG*, passando pelos valores intermédios *NP* e *NM*. Da mesma forma, entre a diagonal e o canto inferior esquerdo, será de esperar que o consequente da regra aplicável varie gradualmente entre *ZE* e *PG*. Isto se exprime, mais classicamente sob forma tabular, como veremos na seção seguinte.

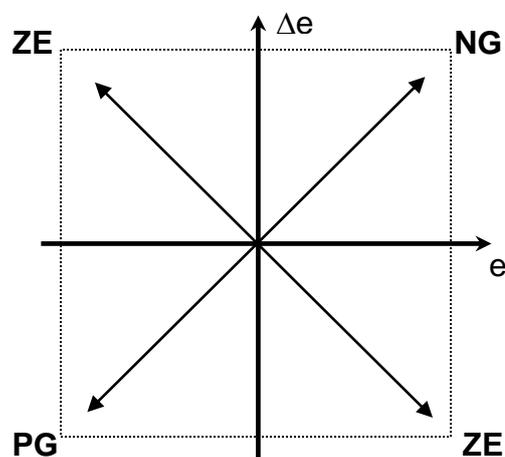


Fig.3.8 – Espaço de Fase para as variáveis do Controlador

### 3.5. CONCLUSÃO

O comportamento e o desempenho de sistemas de inferência fuzzy em geral, e de controladores fuzzy em particular, dependem de vários aspectos relacionados a sua estrutura e implementação, tais como: número de conjuntos fuzzy associados a cada variável, formas das funções de pertinência, funções de implicação, operadores para **e** e **ou**, método de defuzzificação e, se for o caso, técnicas de quantização e fatores de escala.

Nesse capítulo a lógica fuzzy foi abordada visando especificamente a sua aplicação a sistemas de controle. Essa teoria oferece uma alternativa atraente

para o desenvolvimento de controladores que sejam usados em plantas, cujos parâmetros variam ou em plantas difíceis de serem descritas ou modeladas em termos precisos. Muito desses sistemas apresentam características ruins se são controlados por métodos convencionais, mas podem ter a sua operação otimizada pelo uso de controladores fuzzy. Em geral, para projetar um controlador fuzzy, não é necessário ter profundo conhecimento matemático sobre o sistema, mas dominar intuitivamente o seu funcionamento. Isso pode ser conseguido pela experiência e/ou por observação.

No próximo capítulo será demonstrado como será realizado o modelamento do controlador fuzzy do sistema de controle de temperatura do aquecedor de óleo térmico da planta de Anodo Verde da Albrás.

## CAPÍTULO 4

### 4.1 -PROJETO E IMPLEMENTAÇÃO DO CONTROLE FUZZY DO AQUECEDOR DE ÓLEO TÉRMICO (HTM)

#### 4.1.1 – Introdução

As explicações dadas nos capítulos anteriores auxiliam a compreensão dos conceitos e do raciocínio necessário para o uso do Controle *Fuzzy*, que terá aplicação prática neste capítulo. O processo de fabricação de anodos foi também mencionado anteriormente, todavia, serão abordados aqui somente a parte de controle do sistema de aquecimento de óleo térmico (HTM) da planta de Anodo Verde da Albras.

#### 4.1.2- Características do sistema de aquecimento de óleo térmico

O sistema para aquecimento de Fluido Térmico marca DENNEX Figuras 4.1 e 4.2 , consiste basicamente de 12 aquecedores de passagem montados em série e interligados, possuindo uma potencia instalada de 2000KW com tensão de alimentação de 480 volts trifásico projetado para uma temperatura de 300 graus Celsius e fluido térmico marca Mobitherm 605. O conjunto completo é constituído de:

- Unidade de aquecimento multitubular, modelo MTHU-12, com aquecimento elétrico por meio de resistências com potencia unitária de 166 KW tipo tubulares blindadas, Figura 4.3 ;
- Tanque de expansão com volume de 3000 litros, construído conforme código ASME .Sec VIII ,para temperatura de trabalho 300 C° e pressão de projeto 0,3 Bar , representação no supervisório Figura 4.4;
- Tanque de dreno com volume de 8000 litros, construído conforme código ASME .Sec VIII ,para temperatura de trabalho 300 C° e pressão de projeto 0,3 Bar;
- Painéis de força e comando (Figura 4.5)para controlar a potência instalada de 2000KW 480 volts em regime de modulação continua por meio de 4 tiristores SCR de alta potencia montados em configuração

antiparalelo independentes atuando sobre duas fases: R e T, de alimentação elétrica de fabricação ADELCO, esquema elétrico Figura 4.6;

- Um tanque de selo térmico.



Figura 4.1- Unidade de Aquecimento MTHU-12

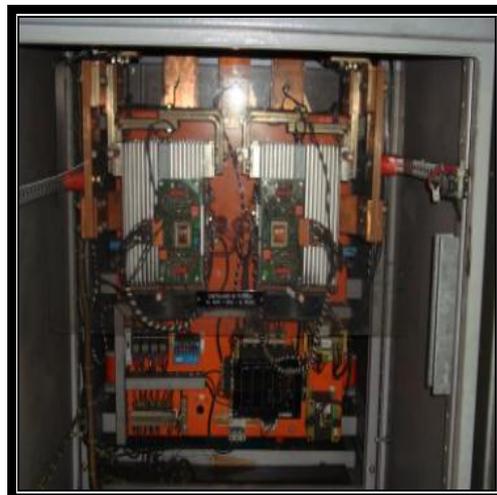


Figura 4.2- Unidade de Potência ADELCO



Figura 4.3- Resistência 166 KW tipo tubulares blindadas



Figura 4.4- Fluxo do Sistema de Aquecimento.



Figura 4.5- Painel de Força da Unidade Tiristorizada.

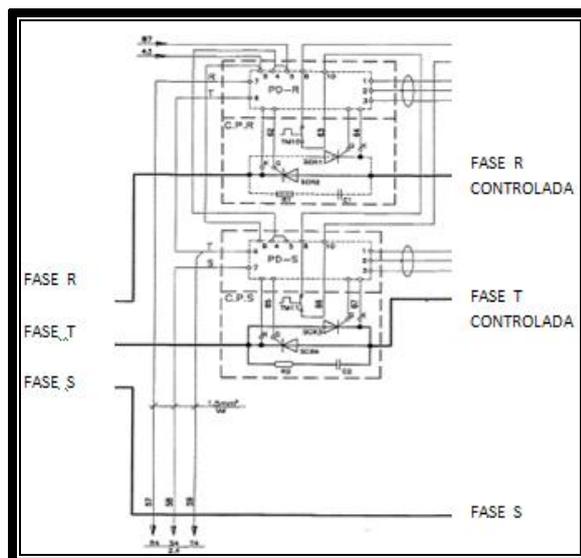


Figura 4.6- Esquema Elétrico da ligação do Tiristor

#### 4.1.3 - Controlador Lógico Programável (CLP) utilizado na implementação controle fuzzy na planta da Albrás

Uma grande parte do controle de fabricação de anodos da Albras é feita por Controladores Lógicos Programáveis (CLP), que possuem rotinas definidas para cada etapa do processo. A segunda seção do capítulo faz uma pequena abordagem a respeito do CLP, com o intuito de facilitar a compreensão de como o algoritmo fuzzy foi elaborado, e em seguida, serão abordados todos os passos que foram traçados na elaboração do controlador *Fuzzy*.

Para a execução de um controle automático em uma unidade industrial [18], necessita-se basicamente de um dispositivo que execute três passos: faça a leitura dos sinais de entrada enviados, por exemplo, por sensores, por chaves seletoras ou por medidores de vazão, pressão ou temperatura; realize o processamento desses dados de acordo com uma lógica pré-determinada; envie os resultados aos dispositivos de saída (Figura 4.8), como por exemplo: lâmpadas, relés de controle, contadores de partida de motores, etc.

Uma das maneiras de realizar o controle é utilizando-se um Controlador Lógico Programável. O CLP, Figura 4.7[18] é um equipamento modular

programável que pode ser configurado de acordo com cada aplicação. Basicamente ele é composto por uma fonte de alimentação, por um “cartão” que possui um processador responsável pela execução dos programas e por um chassi (backplane) que possui conectores (slots), onde outros cartões podem ser conectados. A comunicação entre o CLP e esses cartões é realizada através do chassi.

De acordo com cada aplicação são adicionados cartões específicos. Assim, se há a necessidade de ler variáveis analógicas, a exemplo da temperatura, corrente ou pressão, adiciona-se ao “chassi” do CLP um cartão de entradas e saídas analógicas. Por outro lado, se os dados a serem lidos são variáveis digitais, como por exemplo, recebendo um dos dois eventos possíveis de uma chave seletora “liga/desliga” adiciona-se um cartão de entradas digitais.

A memória do CLP é do tipo não-volátil. Assim, em uma queda de energia não são perdidos os programas armazenados em sua memória bem como os valores de estado anteriores.

Para a programação de um CLP, necessita-se de um computador onde os programas são inicialmente desenvolvidos e depois transferidos para o CLP, através de porta e cabos seriais apropriados. Após a transferência dos programas, o CLP se torna independente do computador, não havendo mais a necessidade de sua utilização.

Existem diversas formas se programar um CLP, tais como linguagem estruturada, o diagrama de blocos, listas de instrução e linguagem Ladder, sendo esta última a mais utilizada.

A princípio, a modelagem inicial do programa de controle fuzzy para o Aquecedor de Óleo Térmico foi elaborada em linguagem C++, sendo que foram realizadas várias simulações, tentando representar teoricamente a dinâmica de funcionamento do sistema. Foram feitos diversos programas, com o intuito de definir qual seria a base de regras e funções de pertinências mais convenientes para a realidade do Aquecedor de Óleo Térmico. Pelo fato do CLP ter de memória limitada, e estar em operação constante no processo, torna-se muito arriscado fazer intervenções no mesmo para fazer alterações na lógica de controle do processo, pois se uma determinada ação que não estava prevista ocorra, a planta pode entrar em colapso, vindo a interromper a produção[18].



Figura 4.7 – Chassi com Processador ,  
Cartão de Entrada Analógica, Cartões de  
Entrada e Saídas digitais do CLP

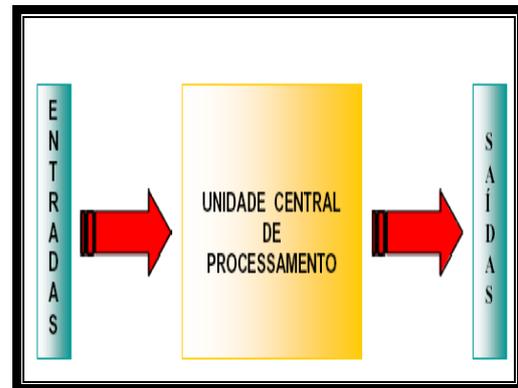


Figura 4.8- Estrutura Básica do CLP

#### 4.1.3 .1- Controlador Lógico Programável (CLP) linguagem ladder

Neste tópico mostramos as etapas da composição da linguagem ladder que foi implementada no CLP oriunda da linguagem C++ que foi criada a partir do compilador fuzzy. No CLP serão imputados os blocos correspondentes a linguagem criada pelo gerador de algoritmo fuzzy como segue : *bloco inicializador de variável* Figura 4.9 –*Erro* e Figura 4.10- *Variação do Erro* ; *bloco fuzyficador de variável* Figura 4.11- *Erro* e Figura 4.12- *Variação do Erro*; *bloco de aplicação de regras* Figura 4.13 ; *bloco máximo/mínimos* Figura 4.14 e finalizamos com o *bloco defuzyficador* Figura 4.15.

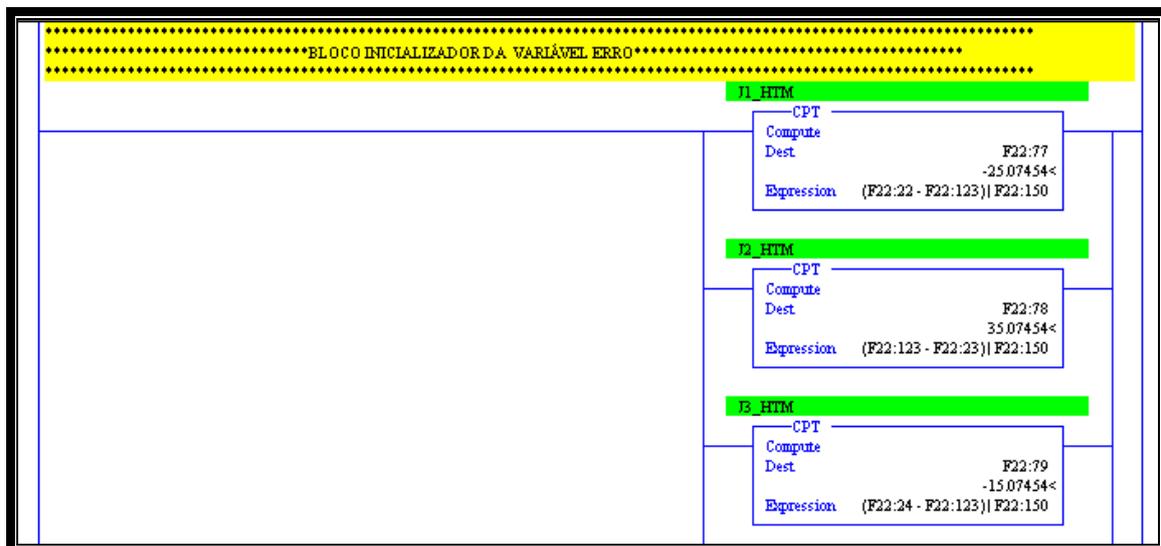


Figura 4.9 – Bloco inicializador da variável Erro.

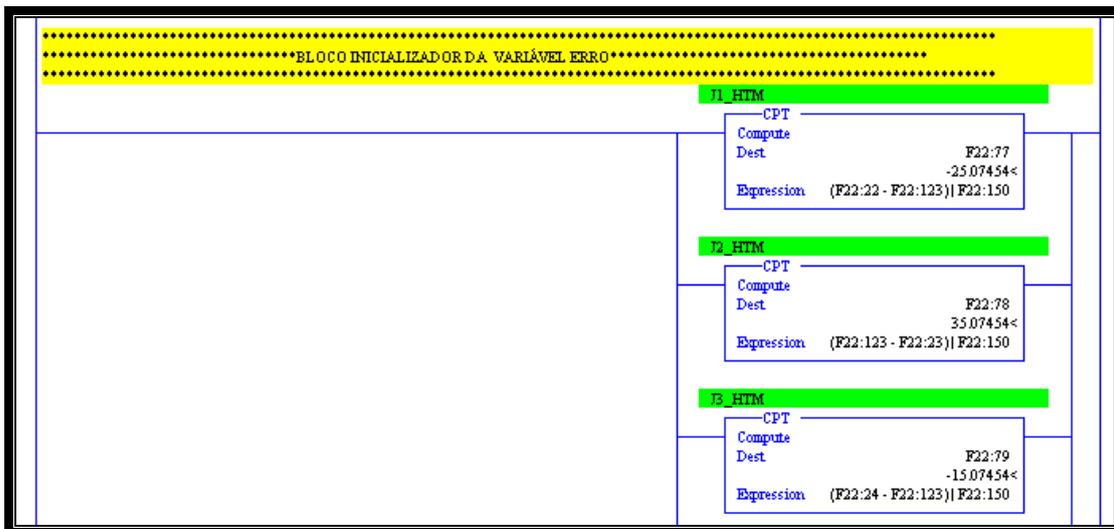


Figura 4.10- Bloco inicializador da variável Variação do Erro.

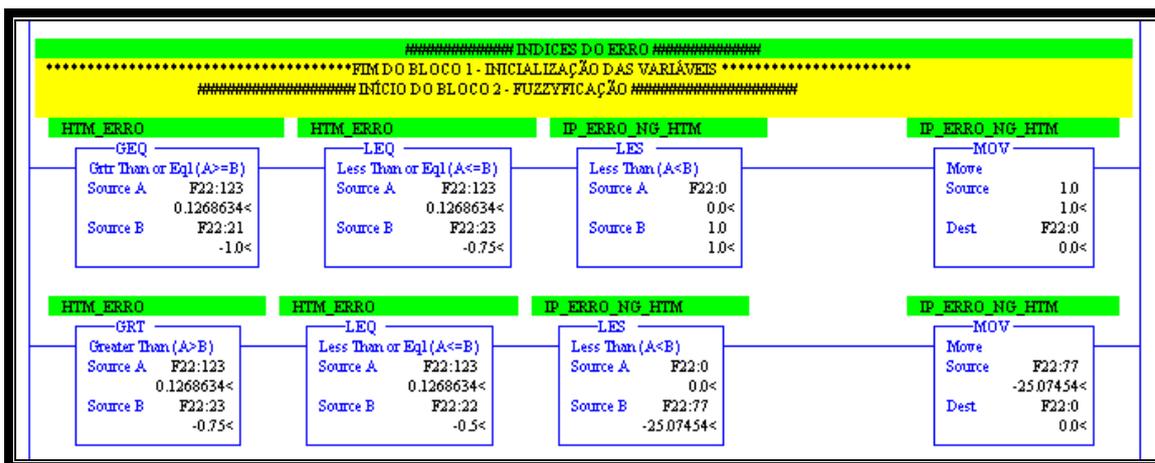


Figura 4.11- Bloco fuzzyficador da variável Erro.

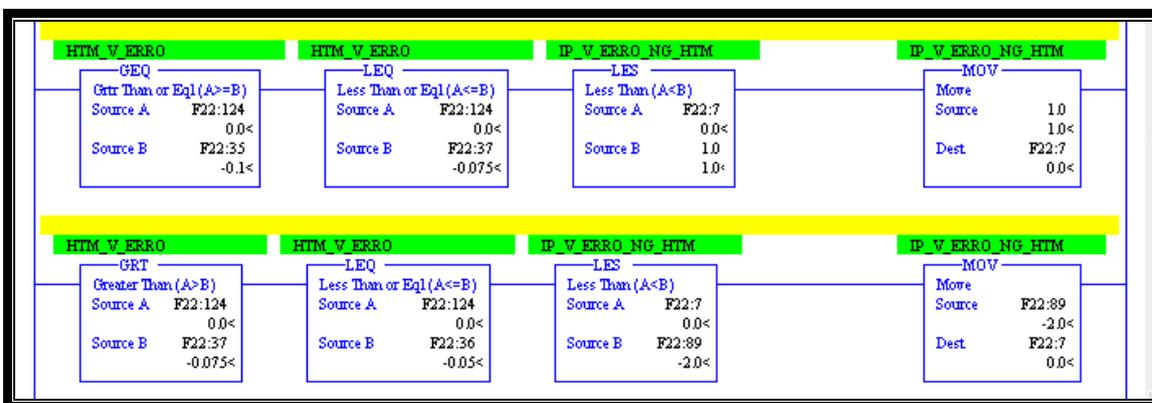


Figura 4.12- Bloco fuzzyficador da variável Variação do Erro.

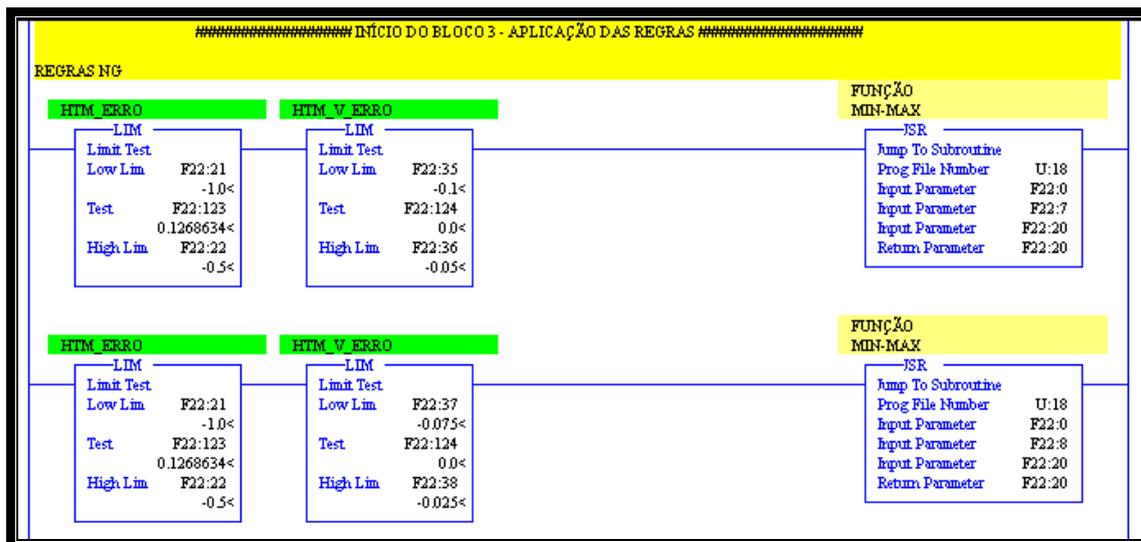


Figura 4.13- Bloco de aplicação das regras fuzzy.

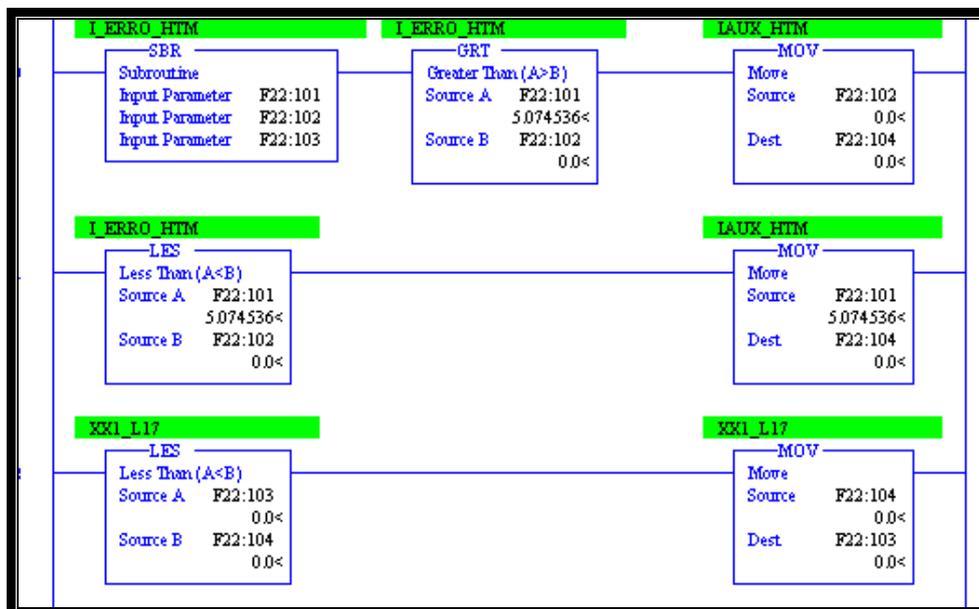


Figura 4.14- Bloco de máximo / mínimos fuzzy.

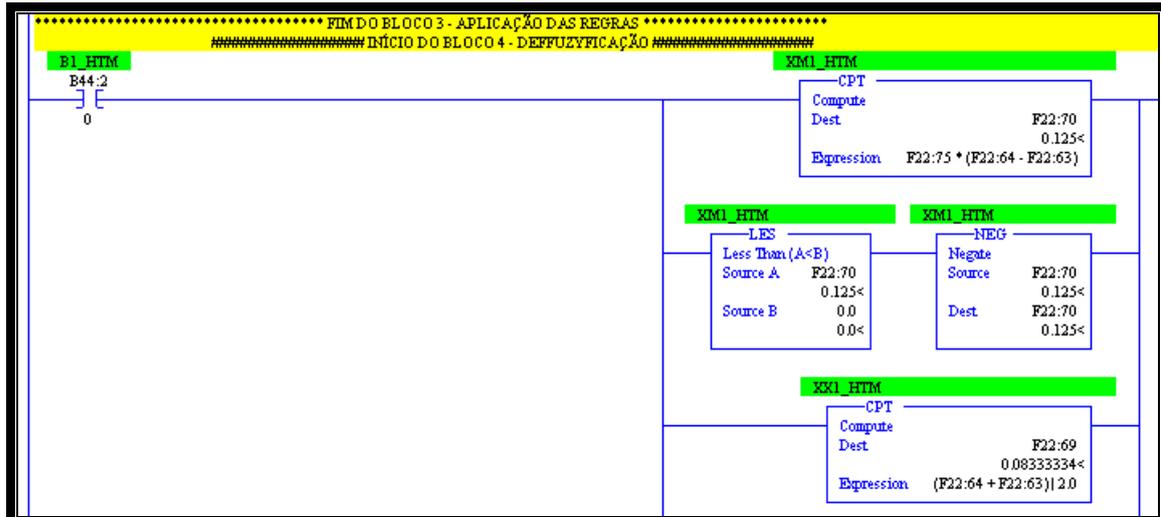


Figura 4.15- Bloco defuzificador.

#### 4.1.4- Gerador de algoritmos fuzzy

Para o desenvolvimento do algoritmo fuzzy foi utilizado um gerador de código C++ [6]. Este compilador gera códigos em linguagem C++ ou em linguagem Matlab®, para utilização em sistemas computacionais baseados em Windows. Após a geração dos códigos os mesmos serão transformados em linguagem Ladder para então o CLP processar o algoritmo fuzzy e gerar as respostas de saída que serão responsáveis pela ação de controle da temperatura do aquecedor. O compilador possui telas na quais você pode entrar com as grandezas que estarão envolvidas no processo, conjunto de regras a ser utilizada e a ultima tela que fará a geração do código em linguagem C++ o qual será transformado em linguagem Ladder no CLP. Junto com as Figuras : 4.16, 4.17 ,4.18 e 4.19 representam as fases usados na geração do código do compilador fuzzy , segue o código gerado em linguagem C++ defuzificador áreas sobrepostas.

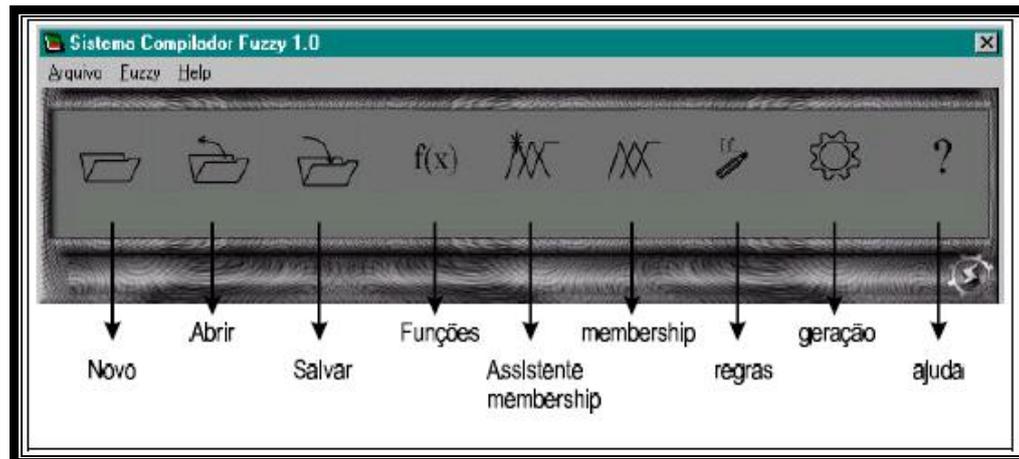


Figura 4.16 – Tela inicial do compilador fuzzy.

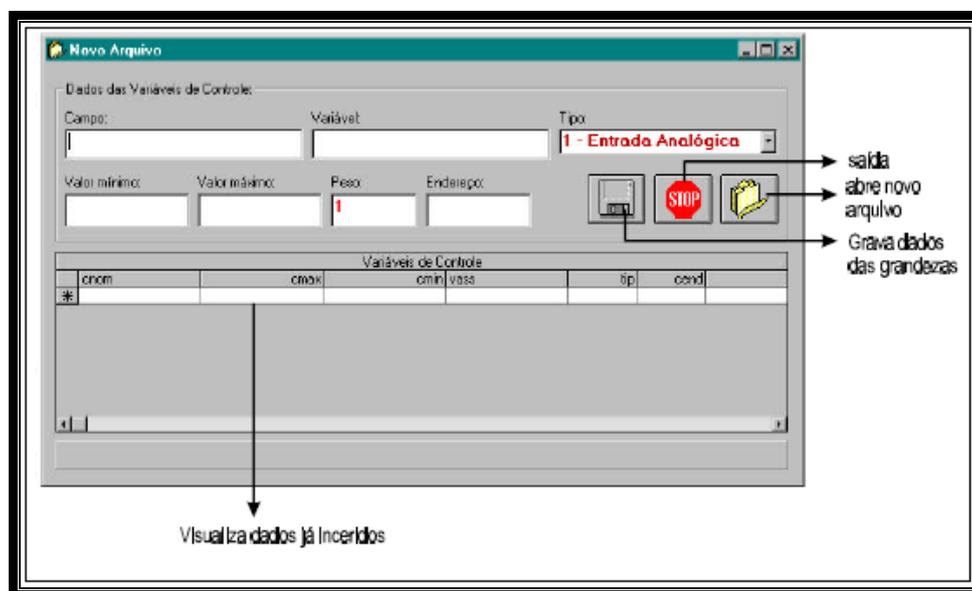


Figura 4.17- Entrada das grandezas de atuação.

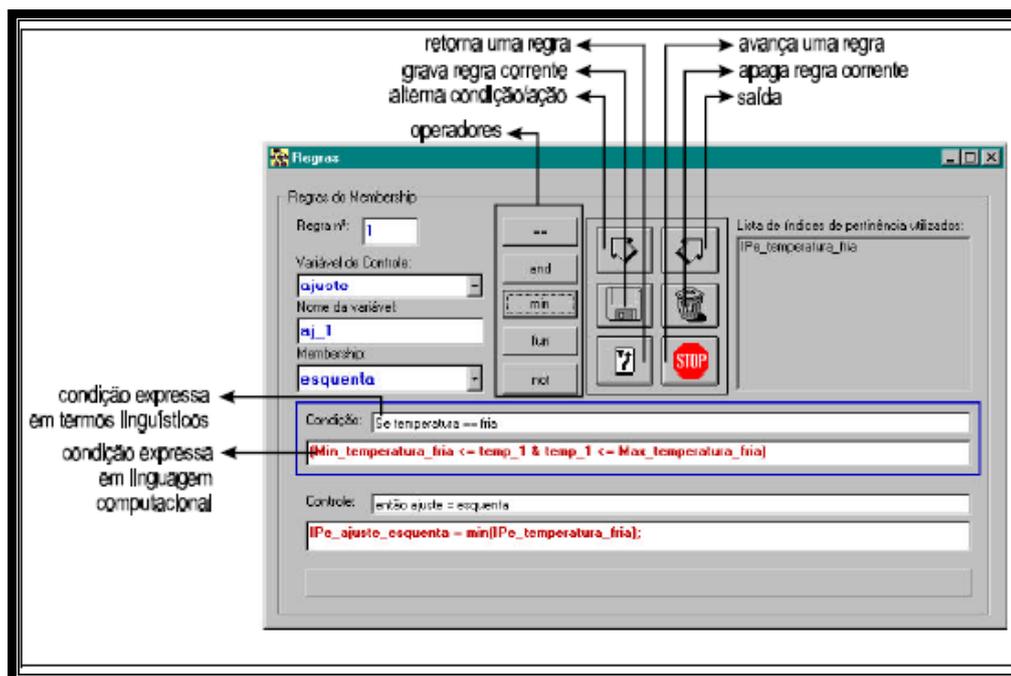


Figura 4.18- Tela de entrada de regras.

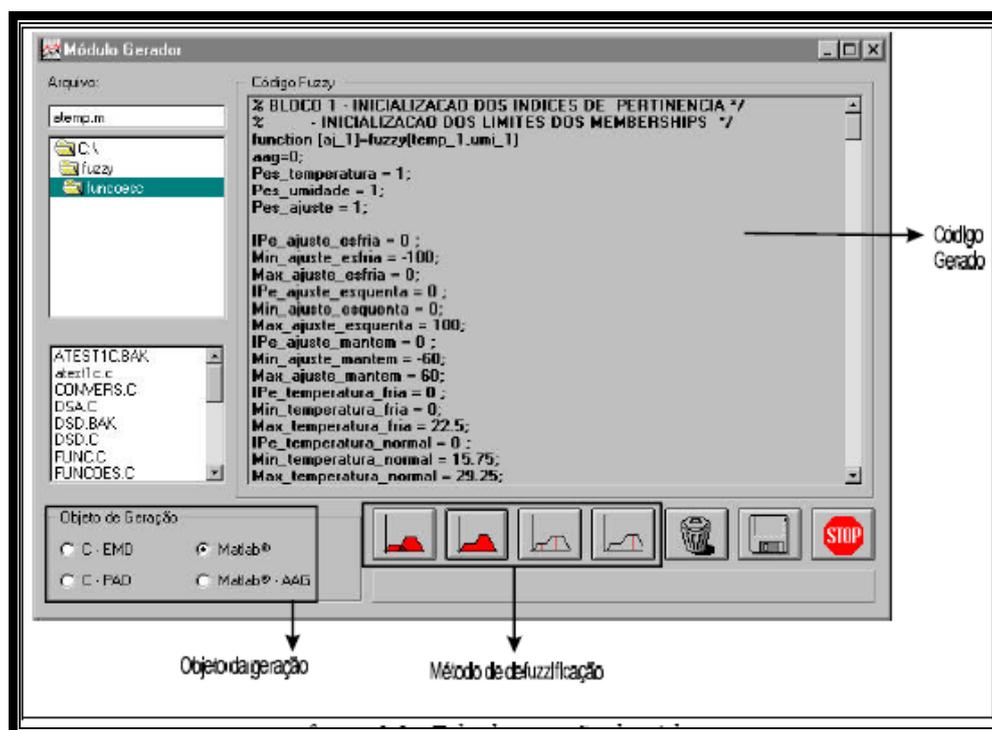


Figura 4.19- Tela de geração de códigos.

#### 4.1.5- Modelagem e análise do controlador fuzzy

Em técnicas convencionais de controle, a intensidade do sinal de controle é calculada a partir de um conjunto de sinais de entrada, em função de um modelo matemático constituído por um conjunto de equações descrevendo o processo [19]. Mas exprimir desse modo a experiência humana (e que valiosa ela é, em tantas circunstâncias!) é uma tarefa quase impossível, e por isso muitos processos de controle industrial têm de continuar a depender da intervenção de um operador. A lógica difusa veio proporcionar um quadro de desenvolvimento de controladores que refletem de forma muito próxima esse tipo de experiência humana porque os controladores fuzzy baseiam o seu princípio de funcionamento em regras, na maioria das vezes com uma correspondência linguística estabelecida pela teoria dos conjuntos fuzzy. Porém, isto não significa que o desenvolvimento de controladores fuzzy seja uma tarefa muito simples, sendo que não há forma de se dispensar um bom conhecimento teórico sobre controle e procedimentos de engenharia no desenvolvimento. Um controlador fuzzy organiza-se, em termos gerais, de acordo com o representado na Figura 4.20 [5]. Os blocos mais importantes são:

- ✓ **Módulo fuzzyficador:** executa as funções de leitura ou medida de grandezas e normalização de valores, e posteriormente a transformação dos valores numéricos em instâncias de variáveis linguísticas com descrição difusa e função ou valores de pertinência correspondentes;
- ✓ **Base de conhecimentos:** inclui as definições das funções de pertinência para cada variável de estado e de controle, bem como a coleção de regras (difusas) aplicáveis;
- ✓ **Máquina de inferência:** mecanismo pelo qual são determinadas as possíveis reações determinadas por cada regra, em resposta a um determinado estímulo, e combinadas essas reações de forma a obter uma descrição global (difusa) da resposta do controlador;

- ✓ **Módulo defuzzyficador:** converte a resposta difusa num valor numérico, e procede à conversão de escala desse valor para o domínio dos sinais de controle.

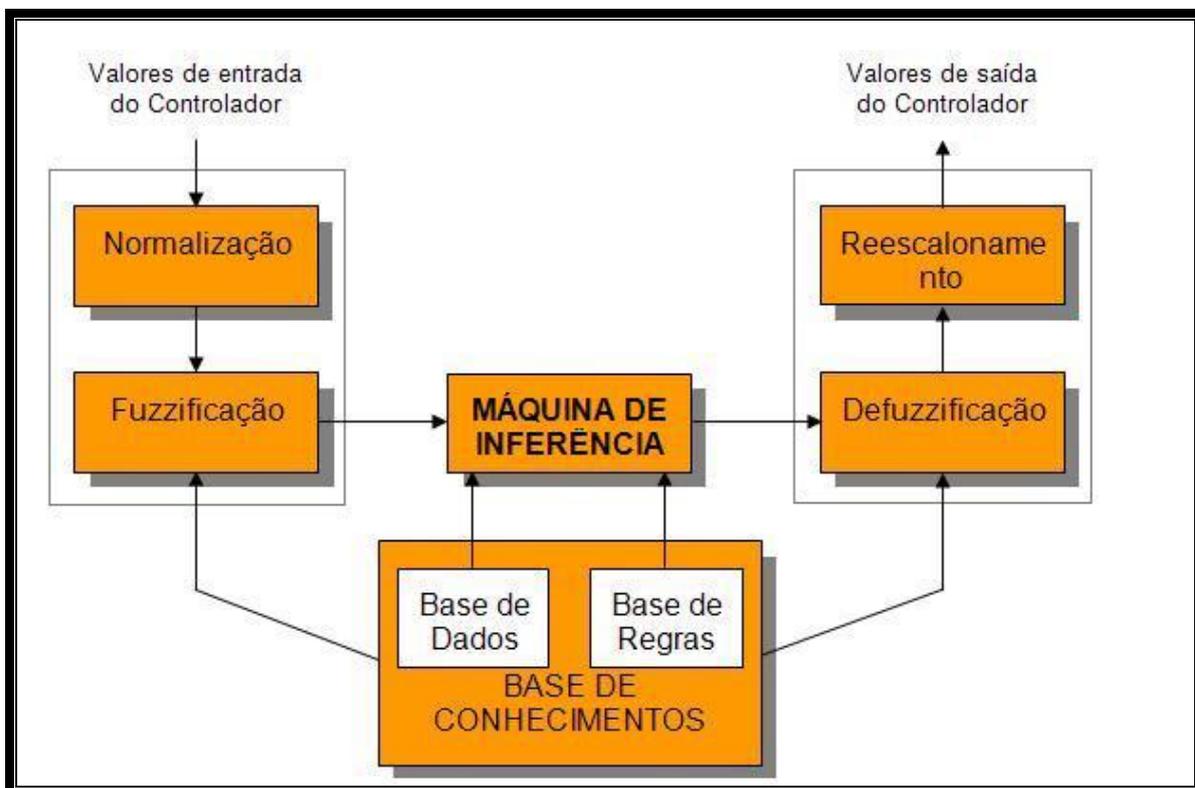


Figura 4.20– Diagrama esquemático da estrutura de um Controlador Fuzzy.

#### 4.1.5.1- Fuzzificação

No controle fuzzy proposto as variáveis de entrada do sistema são definidas como: Erro de temperatura (Erro entre a temperatura de referência (SP) e a temperatura média( $T_m$ ) )  $e_o$  e o incremento do Erro  $\Delta e_o$ , de acordo com a Figura 12 Estes valores de entradas são obtidos através de um módulo conversor analógico digital do CLP (Controlador Lógico Programável) e devem ser fuzzificados para que possam ser manipulados no universo fuzzy [23].

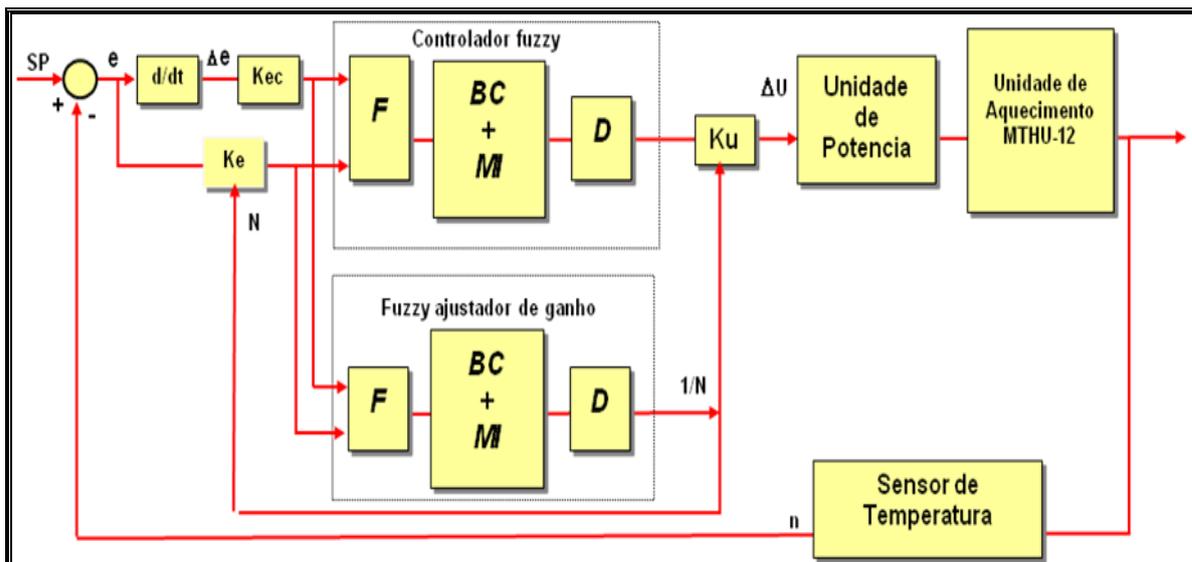


Figura 4.21- Diagrama esquemático do Controlador Fuzzy do Aquecedor de óleo térmico.

A variável de controle do sistema é a temperatura do óleo térmico, que *deve ser ajustada de tal modo a garantir que a sinal de tensão que trabalhe em torno do set point com o menor erro possível, garantindo uma estabilidade na energia de mistura da pasta anódica.* O sistema de potência pode ser traduzido como o elemento final de controle que atuará na liberação de energia para o aquecedor.

Em seguida são equacionados o Erro e o seu incremento para um instante de tempo  $k$  e um instante anterior  $k-1$  e a saída  $\Delta U(k)$ .

$$e_o = SP(k) - n(k) \quad (1)$$

$$\Delta e_o(k) = e_o(k) - e_o(k-1) \quad (2)$$

$$\Delta U(k) = U(k) + U(k-1) \quad (3)$$

onde:

$SP(k)$  é o *set point*, ou seja, a temperatura de referência, especificada pela engenharia de processo (geralmente em torno de  $260\text{ C}^\circ$ ), e  $n(k)$  é a temperatura média no instante  $k$ . Os Blocos são F (Fuzificador), BC (Base de Conhecimentos,

MI (Máquina de Inferência), D (Defuzificador), A/D (Conversor Analógico/Digital) e S (Derivador).

A faixa de trabalho do Erro  $e_o$  e seu incremento  $\Delta e_o$  são estabelecidos com base na escala de temperatura de trabalho do aquecedor que transfere temperatura para o óleo, que o sistema de potencia (tensão de 480V e corrente de 0 a 2000 A). Já a variável de controle  $\Delta U(k)$  tem a sua escala definida pelo sinal de 0 a 10 V, que é liberado para placa de disparo do tiristor do sistema de potência.

Como resultado das regras de controle se obtém os valores do sinal de controle  $\Delta U(k)$ , e como as operações são realizadas no universo fuzzy, estes valores deverão ser defuzzificado para que o CLP possa tomar ação corretiva adequada na temperatura de processo, de forma a manter a temperatura em torno da referência. A princípio, no programa feito via linguagem C++ estabeleceu-se um conjunto de 49 regras de controle, dando a ação de controle uma excursão bem fina em torno do *set point*. Estabeleceu-se somente sete predicados (Tabela 3 para cada variável fuzzy, gerando uma base de regras de 49 leis de controle.

Tabela 3 - Variáveis Lingüísticas

Positivo Grande	PG
Positivo Médio	PM
Positivo Pequeno	PP
Zero	ZE
Negativo Pequeno	NP
Negativo Médio	NM
Negativo Grande	NG

#### 4.1.5.2- Funções de pertinências

As variáveis fuzzy são definidas assumindo valores de um grau de pertinência para cada elemento do universo fuzzy de discurso. Existem vários tipos de funções de pertinência, todavia para reduzir o esforço computacional, e devido a limitação de instruções matemáticas na CPU do CLP foram utilizadas funções triangulares, conforme a Figuras: 4.22 e 4.23, onde o sinal de variável de processo temperatura foi normalizado para 1 com o seguinte procedimento : multiplicou-se o sinal de temperatura e *set-point* por 10 e calculou-se erro ( erro=

**set-point** – temperatura) e dividiu-se o mesmo pelo **set-point**, assim conclui-se a normalização do Erro -1 a 1; e seu incremento está constituído pela excursão de  $-0,1$  a  $0,1$ , enquanto a variável de controle é excursionada de  $-1$  a  $1$ . O valor da função de pertinência varia de 0 a 1.

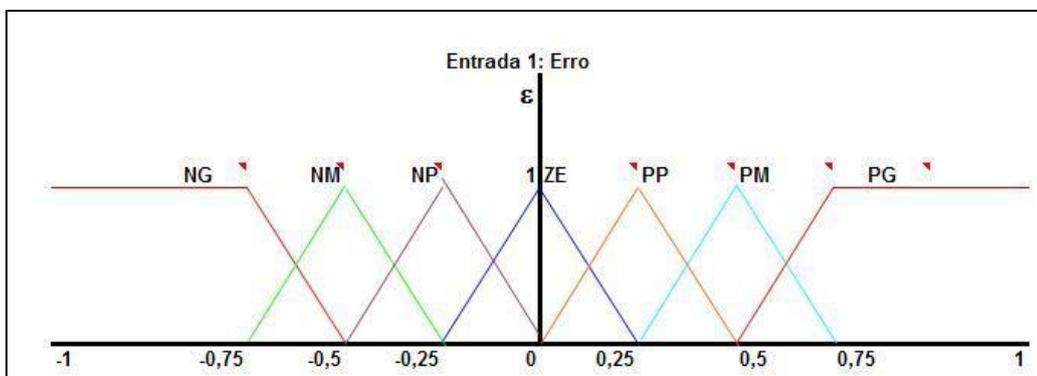


Figura 4.22 – Função de Pertinência Triangular do Erro.

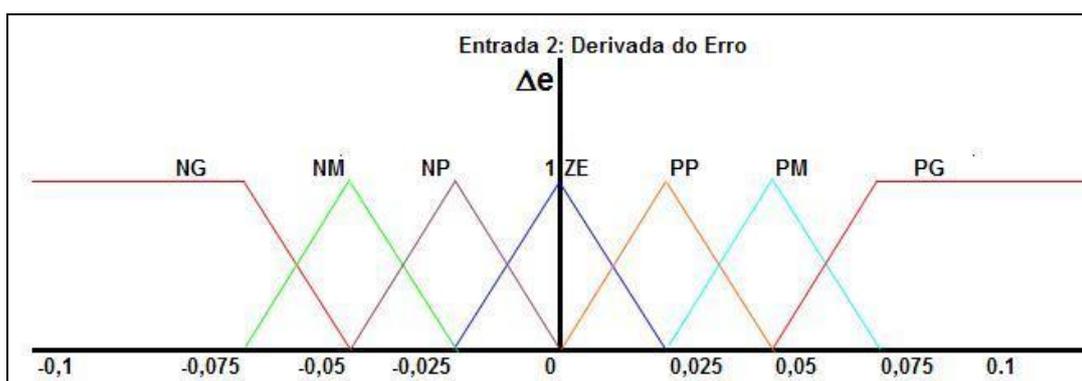


Figura 4.23 – Função de Pertinência Triangular do Incremento do Erro.

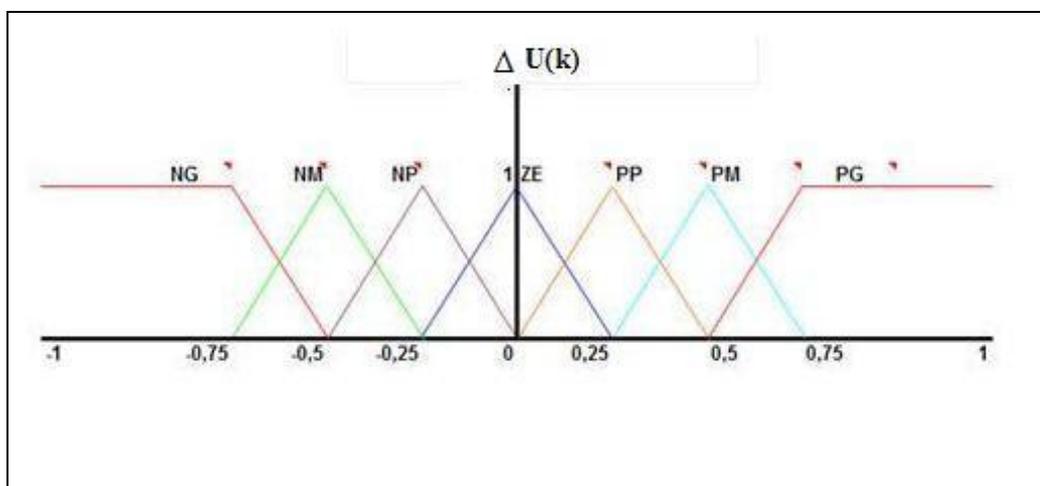


Figura 4.24 – Função de Pertinência Triangular de Saída

A função de pertinência triangular da variável linguística está determinada por três números. Por exemplo, na Figura 4.24, o predicado Zero da variável Erro Z:  $\{-0,25, 0, 0,25\}$ . As funções de pertinência da extrema direita e esquerda, por não terem forma triangular, assumem dois números.

As funções de pertinências para o Fator de ajuste N são: Contrair Grande (CG); Contrair Pequeno (CP); OK; Amplia Pequeno (AP); Amplia Grande (AG);

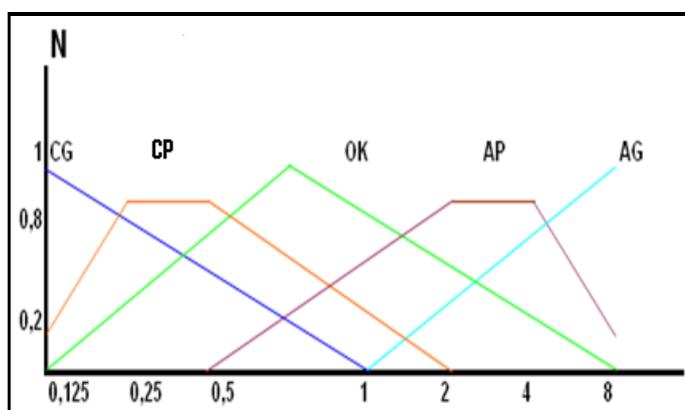


Figura 4.25- Fator de ajuste N

#### 4.1.5.3- Análise dinâmica da variável do erro

O erro produzido pela diferença entre *Set Point* e a temperatura média tem uma forma de onda ondulante e convergente [32]. A característica convergente da temperatura logo depois de produzir o erro, pode ser vista na Figura 4.25, onde C1, C2, C3.. denotam os pontos de cruzamento da temperatura e ao longo do set point, enquanto os pontos m1, m2, m3... são os valores máximos; e A1, A2, A3... denotam intervalos de referência para explicar o processo. As polaridades de  $e_0$  e  $\Delta e_0$  nos intervalos de referência são também indicados nessa figura.

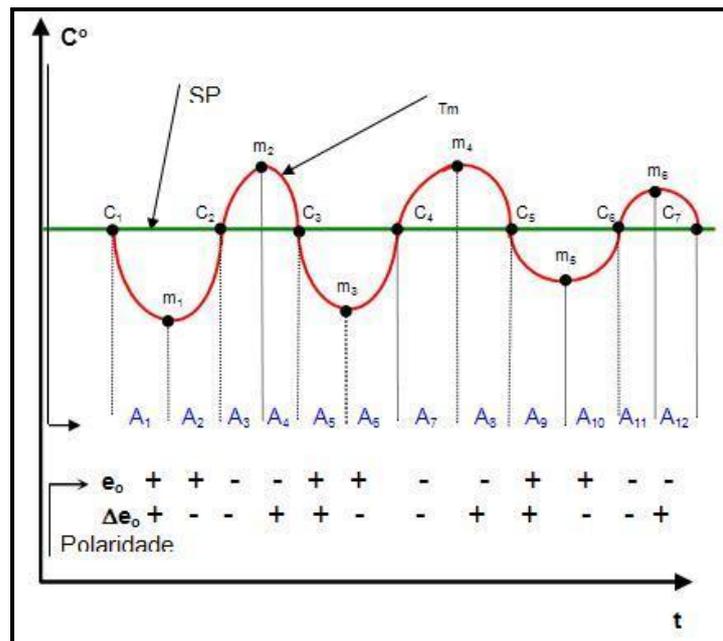


Figura 4.26- Análise dinâmica do sinal erro e sua derivação

- a) Nas áreas A1, A5, A9 da Figura 4.25. , o sinal do erro  $e_o$  é positivo crescente, bem como o sinal da variação do erro  $\Delta e_o$ . Isto significa que o sinal de temperatura está abaixo do set point (SP), logo é preciso aumentar o sinal de referencia (0 a 10 V) para placa de disparo do sistema de potencia para que a temperatura aumente e passe a trabalhar próxima ao SP. Nesse caso a temperatura atual do processo deve receber um incremento, ou seja a ação de controle deve ser positiva para que a temperatura possa aumentar.
- b) Nas áreas A2, A6, A10, o sinal do erro  $e_o$  é positivo e a sua variação  $\Delta e_o$  é negativa. Apesar do erro ser positivo, a ação de controle está fazendo com que o erro  $e_o$  decresça gradualmente. As observações nas áreas A3, A7, A11 e A4, A8, A12 são semelhantes aquelas listadas nos itens a) e b), respectivamente, pois pode-se perceber que há uma simetria entre o segundo e quarto quadrante da tabela 3. Sendo que o primeiro e terceiro quadrante possuem sinais opostos.

#### 4.1.5.4 - Dedução das regras de controle fuzzy

As regras de Controle Fuzzy estão baseadas na experiência dos especialistas de engenharia de controle para se obter boas características do modelo. Nas Tabelas 4 e 5, pode estar incluídas todas as regras linguísticas usadas no controle fuzzy. Algumas observações usadas para determinar as regras de controle são indicadas a seguir:

Tabela 4 - Regras linguísticas do controlador fuzzy.

e / Δe	NG	NM	NP	ZE	PP	PM	PG
NG	PG	PG	PG	PG	PM	PP	ZE
NM	PG	PG	PM	PM	PP	ZE	ZE
NP	PG	PM	PM	PP	ZE	ZE	NP
ZE	PM	PP	PP	ZE	NP	NP	NM
PP	PP	ZE	ZE	NP	NM	NM	NG
PM	ZE	ZE	NP	NM	NM	NG	NG
PG	ZE	NP	NM	NG	NG	NG	NG

Tabela 5 - Regras linguísticas do fator de ajuste N.

e / Δe	NG	NP	ZE	PP	PG
NG	CG	CP	OK	CP	CG
NP	CP	OK	AP	OK	CP
ZE	OK	AP	AG	AP	OK
PP	CP	AP	OK	AP	CP
PG	CG	CP	OK	CP	CG

#### 4.1.5.5 - Máquina de inferência

Os métodos de defuzzyficação mencionados anteriormente, atuam sobre uma área de pertinência preparada a partir da Tabela 6, conhecida como máquina de inferência. Existem vários métodos de inferência. Os mais conhecidos são:

- ✓ Método de Mandani (min-max);
- ✓ Método de Larsen (produto máximo-produto soma);
- ✓ Método de Sugeno (a etapa de decisão entrega um valor fixo de saída ponderada das entradas);

- ✓ Método de Tsukamoto (similar ao de Sugeno variando a fórmula de saída);

Agora exemplificaremos a aplicação das regras de Controle Fuzzy que estão baseadas na experiência dos especialistas de engenharia de controle para se obter boas características do modelo. Algumas observações usadas para determinar as regras de controle são indicadas a seguir:

Na Figura 4.27 inclui-se a variável **P**, que define os intervalos do universo de discurso da variável  $e_o$ . Cada valor de entrada  $e_o$  dá um valor de pertinência não nulo em algum desses intervalos.

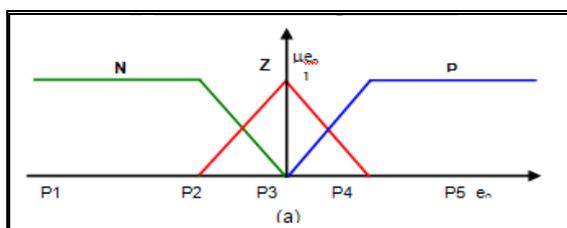


Figura 4.27-Variável erro

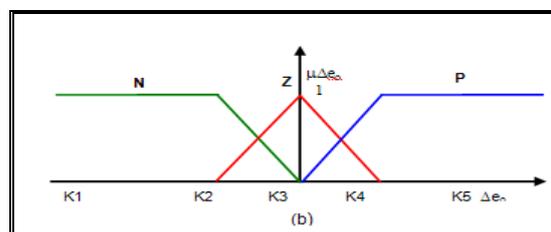


Figura 4.28 – Variável variação do erro

Pela Figura 4.27 percebe-se que:

Se  $e_o \leq P2$  e  $e_o > P1$  e  $\mu_N(e_o) < 1$ , então  $\mu_N(e_o) = 1$

Se  $e_o \leq P3$  e  $e_o > P2$  e  $\mu_Z(e_o) < (e_o + P2)/P2$ , então  $\mu_Z(e_o) = (e_o + P2)/P2$

Se  $e_o \leq P3$  e  $e_o > P2$  e  $\mu_N(e_o) < (-e_o)/P2$ , então  $\mu_N(e_o) = (-e_o)/P2$

Se  $e_o \leq P4$  e  $e_o > P3$  e  $\mu_P(e_o) < e_o/P4$ , então  $\mu_P(e_o) = e_o/P4$

Se  $e_o \leq P4$  e  $e_o > P3$  e  $\mu_Z(e_o) < (P4 - e_o)/P4$ , então  $\mu_Z(e_o) = (P4 - e_o)/P4$

Se  $e_o \leq P5$  e  $e_o > P4$  e  $\mu_P(e_o) < 1$ , então  $\mu_P(e_o) = 1$

Na Figura 4.28 inclui-se a variável **K**, que define os intervalos do universo de discurso da variável  $\Delta e_o$ . Cada valor de entrada  $\Delta e_o$  dá um valor de pertinência não nulo em algum desses intervalos. Usando a técnica anterior, tem-se:

Se  $\Delta e_o \leq K2$  e  $\Delta e_o > K1$  e  $\mu_N(\Delta e_o) < 1$ , então  $\mu_N(\Delta e_o) = 1$

Se  $\Delta e_o \leq K3$  e  $\Delta e_o > K2$  e  $\mu_Z(\Delta e_o) < (\Delta e_o + K2)/K2$ , então  $\mu_Z(\Delta e_o) = (\Delta e_o + K2)/K2$

Se  $\Delta e_o \leq K3$  e  $\Delta e_o > K2$  e  $\mu_N(\Delta e_o) < (-\Delta e_o)/K2$ , então  $\mu_N(\Delta e_o) = (-\Delta e_o)/K2$

Se  $\Delta e_o \leq K4$  e  $\Delta e_o > K3$  e  $\mu_P(\Delta e_o) < (\Delta e_o)/K4$ , então  $\mu_P(\Delta e_o) = (\Delta e_o)/K4$

Se  $\Delta e_o \leq K4$  e  $\Delta e_o > K3$  e  $\mu_Z(\Delta e_o) < (K4 - \Delta e_o)/P4$ , então  $\mu_Z(\Delta e_o) = (K4 - \Delta e_o)/K4$

Se  $\Delta e_o \leq K5$  e  $\Delta e_o > K4$  e  $\mu_P(\Delta e_o) < 1$ , então  $\mu_P(\Delta e_o) = 1$

Tabela 6 – Regras linguísticas

$\Delta e_o$	$e_o$	N	Z	P
N		P	P	Z
Z		P	Z	N
P		Z	N	N

Após a realização dos cálculos dos índices de pertinência, faz-se a aplicação das regras que foram estipuladas na Tabela 6. Quanto mais regras forem definidas, mais preciso será a ação de controle. A regra consiste basicamente em verificar em quais intervalos **P** e **K** as variáveis de entrada se localizam, fazendo-se todas as combinações possíveis para que sempre exista uma ação de controle a ser tomada, mesmo que esta seja nula. Aqui realiza-se a aplicação do método de Mandani. Calcula-se a função mínimo entre os índices de pertinência das entradas e finalmente, calcula-se o máximo entre o índice de pertinência da saída com o resultado da função mínimo das entradas. Isto pode ser visto para uma regra especificamente, como segue:

Se  $(e_o \leq P1$  e  $e_o \geq P3)$  E  $(\Delta e_o \leq K1$  e  $\Delta e_o > K3)$  então

$$\text{retorno} = \text{MIN}(\mu_N(e_o), (\mu_N(\Delta e_o)))$$

$$\mu(U_{pp}) = \text{MAX}(\text{retorno}, \mu(U_{pp}))$$

#### 4.1.5.6 - Defuzificação

A *defuzificação* é a aplicação contrária à *fuzificação*, isto é, no fim de um controle fuzzy temos as saídas como conjuntos fuzzy, mas essas saídas não têm significado no mundo real.

Por isso, não basta chegar à descrição difusa de uma resposta **U**, face a uma entrada **e**. É preciso defuzzificar essa resposta num sinal específico. Não há uma forma única de efetuar esta operação, nem existe uma justificativa teórica

que privilegie uma forma em definitivo. Os vários métodos que analisados no capítulo anterior têm sido mais ou menos utilizados na prática, e têm prevalecido razões de ordem pragmática na sua escolha em particular, a eficiência ou rapidez de cálculo em aplicações de tempo real. O Método que o foi usado nessa aplicação foi o do centro de área Figura 4.29.

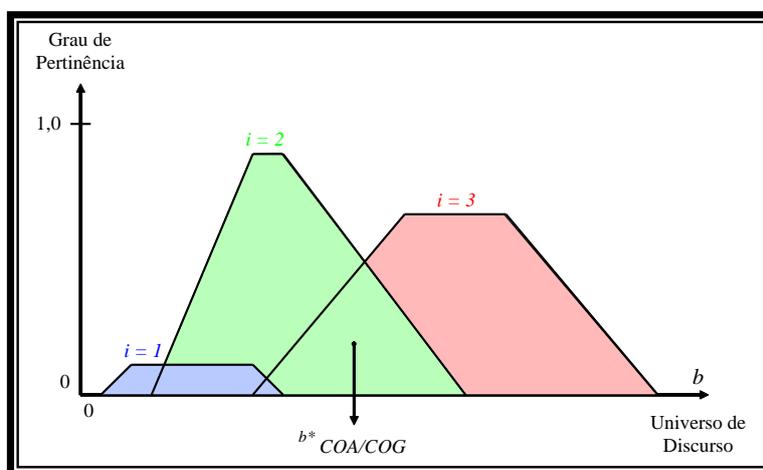


Figura 4.29- Defuzzificação por centro de gravidade.

$$u_{CG} = \frac{\sum_{i=1}^N u_i \cdot \mu_u(u_i)}{\sum_{i=1}^N \mu_u(u_i)} \quad (4.1)$$

#### 4.1.5.7 - Conclusão

Neste capítulo foi abordado a implementação do controlador fuzzy para o Aquecedor de Óleo Térmico (HTM) , dando uma descrição detalhada do sistema de aquecimento , bem como a inserção da Lógica Fuzzy em linguagem ladder no CLP- 5/80 da Rockwell Automation, que nada mais é do que uma CPU dedicada para executar uma seqüência de tarefas pré-programadas. Também foi abordada a forma que foi elaborada o controlador inteligente fuzzy, que controla o a temperatura do Aquecedor de Óleo Térmico da fábrica de Anodo Verde da Albras . Os resultados do desempenho desse controlador fuzzy, podem ser visto no próximo capítulo.

## CAPÍTULO 5

### 5.1 - RESULTADOS EXPERIMENTAIS NA PLANTA DA ALBRAS

#### 5.1.1 – Considerações gerais

Cada sistema de processo da área de Anodo Verde possui o seu sistema de aquecimento de óleo térmico, que opera independentemente em cada planta e que possui as mesmas características de potência. A motivação de se produzir um controlador fuzzy para o sistema de aquecimento, partiu da necessidade da troca do controlador em função da remoção do painel de controle de instrumentação antigo, onde encontrava-se um controlador PID fabricante Watlow, que era responsável pelo controle de temperatura e que foi substituído por um sistema supervisorio (Figura 5.1) interligado ao CLP de controle de processo das plantas de Anodo Verde, na Figura 5.2 temos a sala de controle.



Figura 5.1 - Sistema de controle de temperatura no supervisorio



Figura 5.2 – Sala de controle do sistema supervisório

## 5.2 – CONTROLE FUZZY NO SISTEMA DE AQUECIMENTO DE ÓLEO

O modelamento do controlador fuzzy foi apresentado nos capítulos anteriores, e de acordo com o que foi discutido, as variáveis que o controlador necessita para poder gerar uma saída de controle são: o erro e a sua derivada. O sistema controla a temperatura de processo que está setada em 260 graus Celsius, na Figura 5.3, temos a malha de controle do sistema de aquecimento de óleo térmico. O programa desenvolvido para implementar a estratégia de controle fuzzy foi desenvolvido e implementado em linguagem de C++, e uma biblioteca para implementação de algoritmos fuzzy [6], conforme já descrito no Capítulo 4 (seção 4.1.4), desta dissertação. Subsequentemente, o algoritmo de controle fuzzy foi transcrito para linguagem ladder e implementado em um CLP 5/80, de fabricação da empresa Rockwell Automation [33]. Na figura 5.3 temos a malha de controle do sistema de aquecimento de óleo térmico

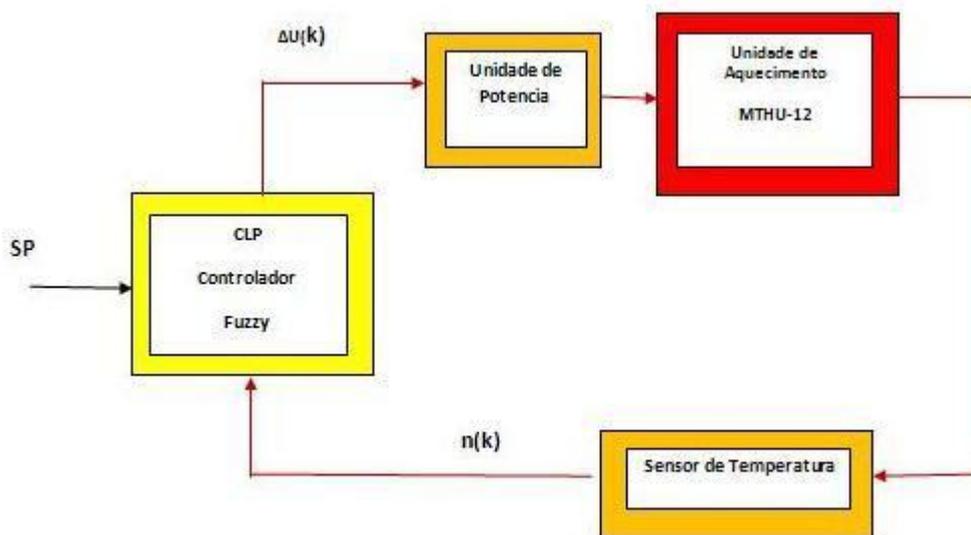


Figura 5.3 – Malha de controle do sistema de aquecimento

Na Figura 5.4 é exibida uma situação em que o sistema controlado pela lógica fuzzy sofre mudança no set point. O *set point* pena verde foi reduzido de 259 graus para 243 graus, produzindo-se um degrau de -16 graus e, pouco depois, passando para uma temperatura de trabalho de 260 graus. Percebe-se que o controlador consegue rastrear os novos pontos de operação. Quando o *set point* aumenta, o controlador fuzzy faz um incremento no sinal de controle pena azul fazendo a temperatura de saída do processo pena vermelha passe a trabalhar em um novo ponto estabelecido.

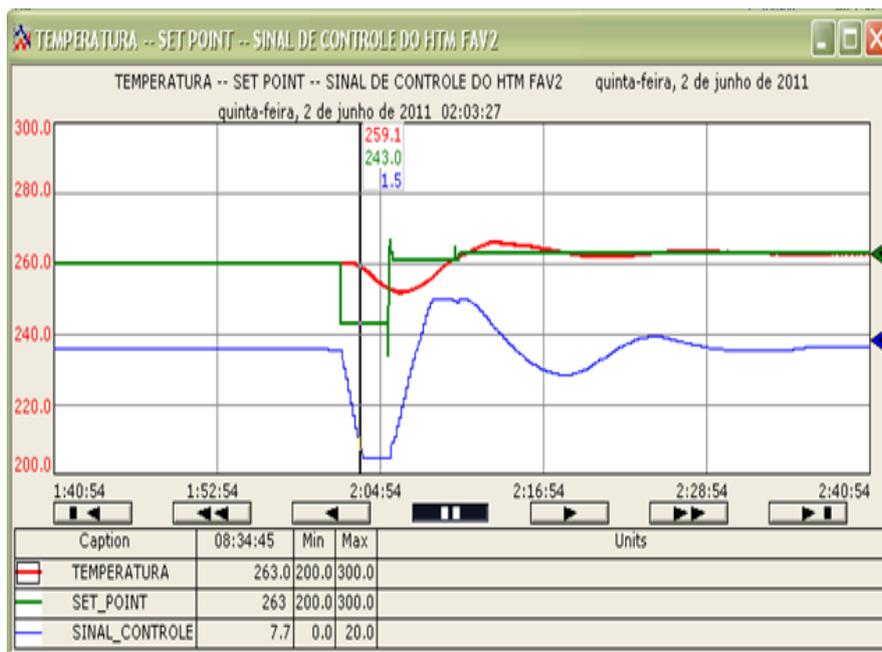


Figura 5.4- Rastreamento do controlador para novos pontos de operação desejados .

Na Figura 5.5 apresentamos a velocidade de resposta a aplicação de um degrau de  $17\text{ C}^{\circ}$ , observamos que o controlado fuzzy obteve um tempo de resposta excelente para aplicação no processo da planta de Anodo Verde.



Figura 5.5 - Velocidade de resposta do Controlador Fuzzy .

Quando sistema encontra-se em regime permanente de trabalho, o sinal de incremento sofre pequenas variações com a finalidade de manter a temperatura de processo pena vermelha em torno do *set point* 260 graus Celsius pena verde como está demonstrado na Figura 5.6.

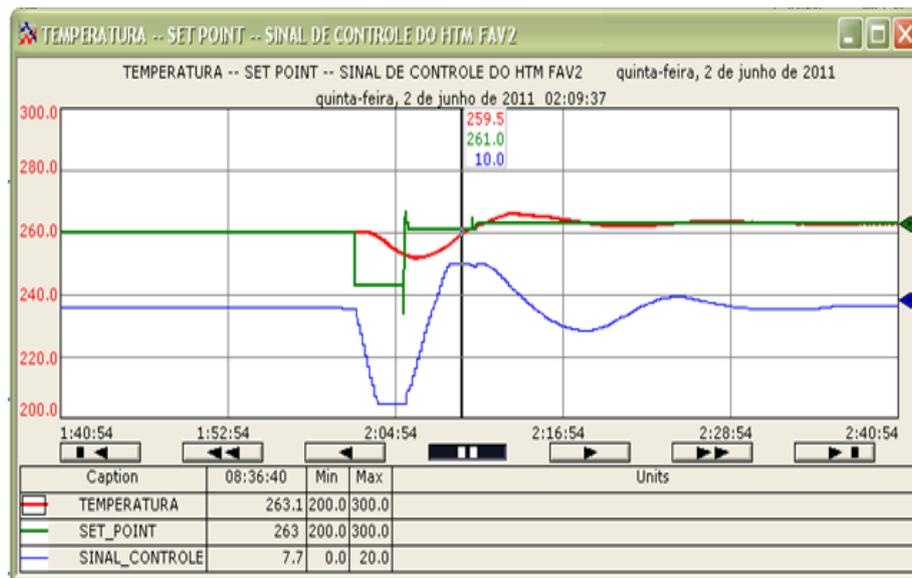


Figura 5.6– Alcance do ponto de acomodação do Controlador Fuzzy em torno do *Set-point*.

Na situação a seguir temos uma mudança da variável de processo pena vermelha assumindo valores acima do *set-point*. O controlador fuzzy reage decrementando o sinal  $\Delta U(k)$  para o modulo de potência do aquecedor de óleo térmico como é demonstrado Figura 5.7.

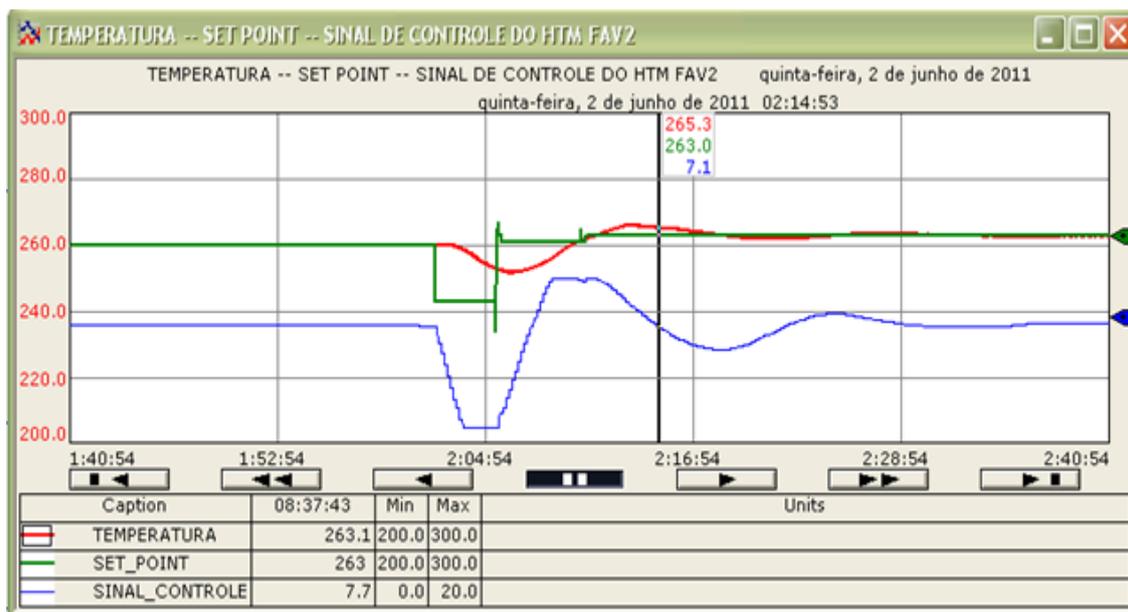


Figura 5.7 – Controlador Reagindo a uma Mudança da Variável de Processo .

O controlador fuzzy atinge o ponto de acomodação em torno do *set point*, como podemos observar na Figura 5.8 o sinal  $\Delta U(k)$  pena azul produzido pelo controlador tem característica suave , significando que há um mínimo esforço de trabalho de controle , logo impactando em um menor gasto de energia para atingir a temperatura da variável de processo pena vermelha 263 c° igual ao *set-point* pena verde.

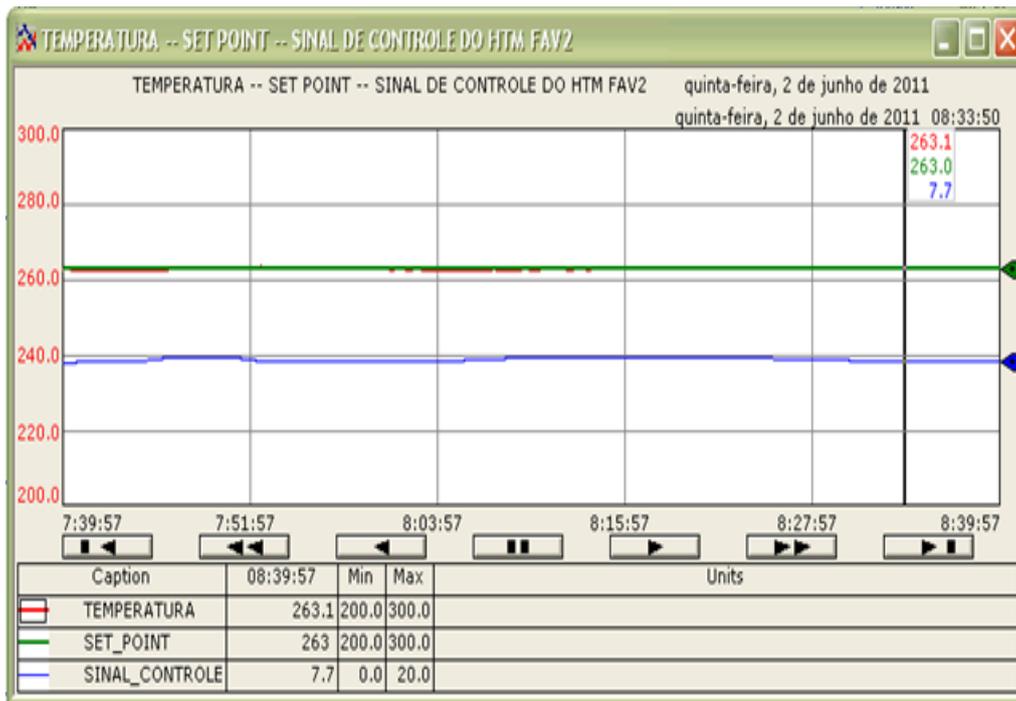


Figura 5.8 – Controlador Fuzzy em estado permanente de acomodação .

### 5.3-COMPARAÇÃO DOS CONTROLADORES FUZZY E PID

Na seção a seguir faremos uma comparação do controlador fuzzy implementado com o controlador que operava antes, controlador PID, faremos uma análise das principais características de desempenho dos controladores:

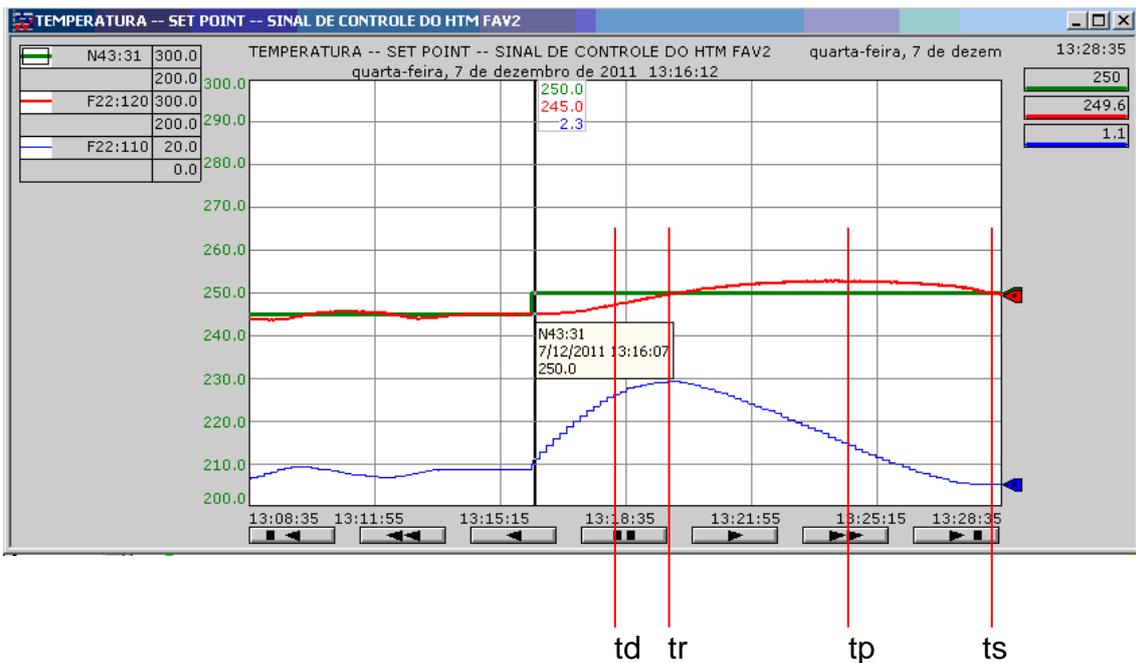


Figura 5.9- Resposta a um sinal degrau controlador Fuzzy

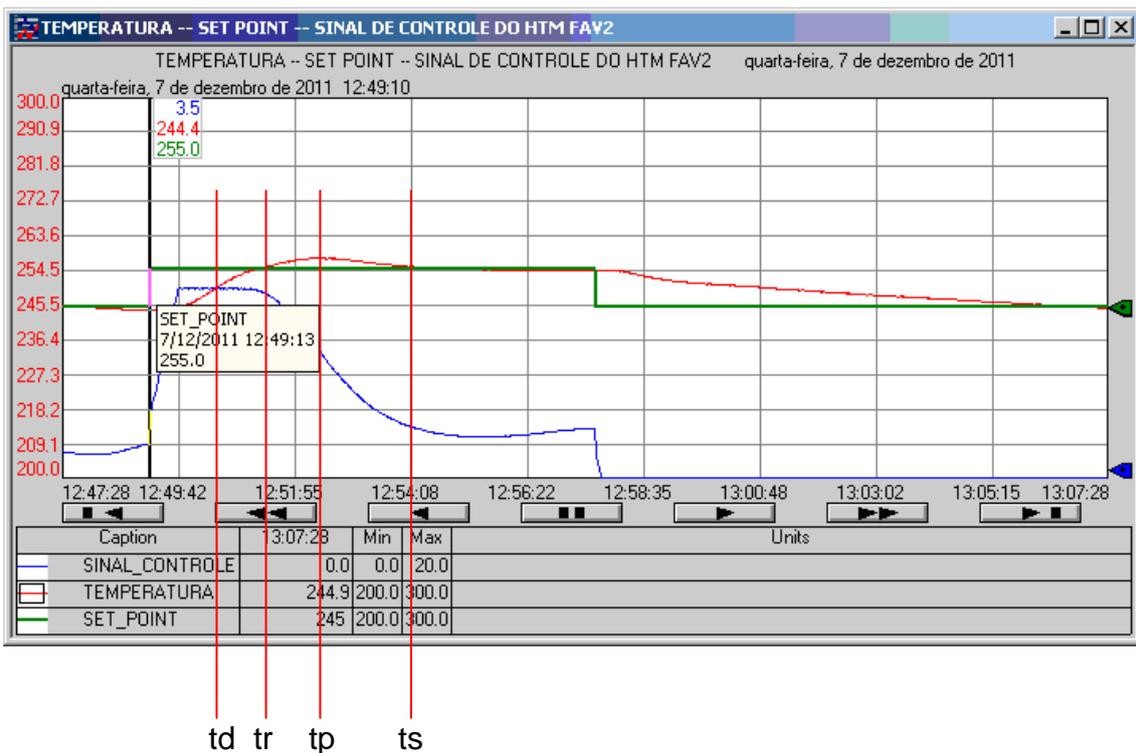


Figura 5.10- Resposta a um sinal degrau controlador Watlow PID.

Legenda :

- td - Tempo de atraso de resposta;
- tr - Tempo de subida;
- tp - Instante de pico ou máxima resposta;
- ts - Tempo de assentamento.

Tabela 7- Características dos controladores PID e Fuzzy.

Parâmetros	Controlador FUZZY Aquecedor	Controlador PID Aquecedor
td	137 seg.	83 seg.
tr	225 seg.	112 seg.
tp	293 seg.	180 seg.
ts	12 minutos	5 minutos

Nas Figuras 5.9 e 5.10 estão representadas as respostas dos controladores : Fuzzy e PID . Na Tabela 7, queremos registrar as características apresentadas por cada tipo de controlador que foi aplicado nas plantas de Anodo Verde da Albras.

Em geral a análise do desempenho ou características de um sistema é realizada com base na resposta deste sistema a uma excitação qualquer . Neste trabalho utilizamos uma excitação degrau para estudar e diferenciar bem o comportamento dinâmico dos controladores fuzzy e PID.

O parâmetro tempo de atraso de resposta (td) , que representa o intervalo no qual o sistema atinge pela primeira vez 50% do seu valor final demonstrou vantagem, ou seja uma menor valor do **controlador PID (83s)** comparado ao **controlador fuzzy (137s)**, haja visto que esse parâmetro retrata o atraso na ação do próprio controlador.

O tempo de subida (tr) revela o tempo que o sistema leva para passar de 0 a100% do seu valor final. Percebemos que o controlador *PID* (112 s) possui uma velocidade maior de resposta do a do controlador fuzzy (225s), na presença de um transitório aplicado ao sistema .

O tempo de pico, também conhecido como sobre-sinal(ou overshoot), relaciona-se com a estabilidade do sistema.Comparando o controlador PID(**180 s**) com o controlador fuzzy (**293 s**) , pela velocidade

de resposta do controlador PID era de se esperar que o tempo do controlador fuzzy fosse maior que o PID.

Já o tempo de assentamento ( ou acomodação) que corresponde ao intervalo que o sistema leva até que a resposta caia dentro de uma faixa de valores centrada no valor final do regime permanente , demonstrou-se uma melhor eficiência do controlador PID (5 min.) em relação ao controlador fuzzy ( 12 min.).

#### 5.4 – CONCLUSÃO

Neste capítulo, apresentou-se e discutiu os resultados da implementação de um controlador PI Fuzzy aplicado ao o processo de fabricação de anodo na fábrica da Albrás. Os resultados obtidos nos testes comparados do controlador fuzzy com o controlador PID que antes era utilizado, de certa forma não alcançou as mesmas características do controlador PID, porém para o processo da planta de Anodo Verde o desempenho do controlador Fuzzy atende com eficiência as necessidades do processo de produção de anodos verde , pois com os resultados obtidos demonstrou-se que o controlador Fuzzy possui um ponto de acomodação satisfatório para o processo da planta de Anodo Verde.

## CAPÍTULO 6

### 6.1 -CONCLUSÃO

A aplicação real de uma de um controlador Fuzzy na planta de Anodo Verde, mais precisamente no controle da temperatura de óleo térmico, cuja etapa faz parte do processo de produção que completa a cadeia de fabricação de alumínio primário, permitiu um grande aprendizado relacionado a aplicação das técnicas de projeto de controlador Fuzzy. Foi observado a aplicabilidade desse controle de maneira real. Sabe-se que os resultados obtidos não demonstram a superioridade do controlador Fuzzy em relação ao controlador PID, pois as características que foram coletadas nos testes percebe-se um melhor desempenho do controlador PID. O controlador Fuzzy implementado apesar de não mostrar nítida superação em termos de curva de desempenho em relação ao controlador PID convencional já existente, o mesmo atendeu com eficiência os pré requisitos do processo de fabricação de anodos, pois o mesmo possui uma boa acomodação em torno do *set point*, que é fator importante nesse processo, onde a temperatura participa de maneira direta na formação da pasta anódica que irá formar o bloco que será usado como pólo positivo na redução. Com certeza existe possibilidade de melhoria das características do controlador Fuzzy implementado que pode ser feita através de uma melhor sintonia do mesmo, como otimizando seu banco de regras e alterando suas funções de pertinência para se obter um melhor resultado. Com o conhecimento das técnicas de controle fuzzy abriu-se uma janela, provocando o interesse da equipe de engenharia da Albras a adentrar em um universo diversificado de controle processo que a mesma possui. Todos os experimentos executados foram seguidos de planejamento e preparação e autorizados pelos gestores da Albras, pois é lógico que na implementação havia possibilidade de erro mínima que fosse, caso a resposta do controlador não fosse satisfatória ao processo da planta de Anodo Verde, o que poderia ocorrer rejeição de blocos anódicos por não atendimento da temperatura necessária para o processo de fabricação de anodo verde.

## 6.2 – TRABALHOS FUTUROS

A Albras – Alumínio Brasileiro S.A. é um dos maiores parques industriais da região norte do Brasil, possuindo o que há de mais avançado em tecnologia de software e automação. O processo de produção do alumínio ocorre em várias etapas, desde a produção do anodo, que o pólo positivo no processo de eletrólise, até a etapa final que é a fundição (lingotamento). Especificamente, a fábrica de anodo possui uma série de equipamentos, e muitos deles operam automaticamente. Um exemplo é o sistema de filtro de mangas da britagem de crosta, que teve o seu sistema de controle alterado por um controlador inteligente. Podemos citar alguns desafios a serem vencidos com a aplicação das técnicas de projeto de controle Fuzzy :

- Controle de velocidade dos motores dos exaustores dos filtros de mangas do forno de cozimento, cujo foco é a economia de energia,
- Controle de altura dos blocos anódicos, que permitirá uma melhor qualidade na fabricação de blocos, projeto em fase de estudo;
- Oportunidade de melhoria da sintonia do controlador Fuzzy aqui dissertado, que poderá ser escrito em forma de artigo.
- Sugere-se a aplicação pelo método takage-sugeno para projetos de controlador Fuzzy .

## REFERÊNCIAS

- [1] PEDRYCZ, W. (1989). **Fuzzy Control and Fuzzy Systems**. - John Wiley and Sons Inc, NY.
- [2] LEE, C.C. (1990). **Fuzzy Logic in Control Systems: Fuzzy Logic Controller, part I and II**, *IEEE Trans. on Systems, Man and Cybernetics*, vol 20, pp 404-435.
- [3] ALBERTOS, P. (1992). **Fuzzy Controllers - All Techniques in Control** - Pergamon Press.
- [4] ZADEH, L.A., (1965). "**Fuzzy Sets**". *Information and Control*, V. 8: 338-353.
- [5] Miranda,V(2003). "**Uma Abordagem sobre Redes Neurais, Lógica Difusa e Algoritmos Evolucionários**". Convênio de Cooperação Internacional Brasil/Portugal, NESC-UFPa, 47-77.
- [6] MELLO, L.(1999), "**Gerador de Algoritmos Fuzzy**". Dissertação de Mestrado, Cap. 3, EFEI(Universidade Federal de Itajubá).
- [7] 2. Zadeh, L.A., (1973). "**Outline of a New Approach to the Analysis of Complex Systems and Decision Processes**". *IEEE Transactions on Systems Man & Cybernetics*, Vol.3: 28-44.
- [8] MAMDANI, E.H., (1974). "**Application of Fuzzy Algorithms for Control of Simple Dynamic Plant**". *Proceedings of the IEE (Control and Science)*, V. 121: 298-316.
- [9] MAMDANI, E.H. & Assillan, S., (1975). "**An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller**". *International Journal of Man-Machine Studies*, Vol. 7: 1-13.

- [10] MAMDANI, E.H., (1977). "**Applications of Fuzzy Logic to Approximate Reasoning using Linguistic Synthesis**", IEEE Trans. Computers, V. 126: 1182-1191.
- [11] KAUFMANN, A., (1975). **Introduction to the Theory of Fuzzy Subsets. Academic Press.**
- [12] BALDWIN, J.F. & Pilsworth, B.W. (1980). "**Axiomatic Approach to Implication for Approximate Reasoning with Fuzzy Logic**". **Fuzzy Sets and Systems**, Vol. 3: 193-219.
- [13] YAGER, R., (1980). "**An Approach to Inference in Approximate Reasoning**" Int. Journal of Man-Machine Studies, Vol. 13: 323-338.
- [14] MIZUMOTO, M., (1982). "**Fuzzy Inference using max-min composition in the Compositional Rule of Inference**". In: Approximate Reasoning in Decision Analysis, M.M. Gupta, E. Sanchez (Ed.), North-Holland: 67-76.
- [15] WHALEN, T. & Schott, B., (1983). "**Issues in Fuzzy Production Systems**", Int. Journal of Man-Machine Studies, Vol. 19: 57-71.
- [16] LEMBESSIS, E. & Tanscheit, R., (1991). "**The Influence of Implication Operators and Defuzzification Methods on the Deterministic Output of a Fuzzy Rule based Controller**". Proceedings of the 4th IFSA Congress, PP. 109-114 (Engineering), Bruxelles, Bélgica.
- [17] MENDEL, J.M., (1995). "**Fuzzy logic systems for engineering: a tutorial**", Proc. IEEE, vol. 83(3): 345-377.
- [18] SILVA, C. Edson., (2004) "**Controlador Inteligente em Misturadores de Pasta Anódica**", Artigo.

- [19] LUJÁN, W., P., & Morais, C., (1999), "**Logica Fuzzy para Control de Velocidad de Motores asincronos**". Artigo, USP(Universidade de São Paulo).
- [20] AZEVEDO, H., R(2003). "**Conjuntos Fuzzy Aplicado ao Controle de Processo**". Tutorial, Universidade Federal de Uberlândia.
- [21] HOLMBLAD, I.P. & Ostergaard, J.J., (1981). "**Fuzzy Logic Control: Operator Experience applied in Automatic Process Control**". FLS Review, F.L.Smith & Co., Copenhagen, Denmark, n. 45: 11-16.
- [22] KICKERT, W.J.M. & Van Nauta Lemke, H.R., (1976). "**The Application of a Fuzzy Controller In a Warm Water Plant**". **Automatica**. Vol. 12: 301-308.
- [23] PAPPIS, C.P. & Mamdani, E.H., (1977). "**A Fuzzy Logic Controller for a Traffic Junction**". IEEE Transactions on Systems, Man and Cybernetics, V. 7: 625-629.
- [24] LEMBESSIS, E., (1984). "**Dynamic Learning Behaviour of a Rule-based Self Organising Controller**". Ph. D. Thesis, University of London.
- [25] PROCYK, T. J. & Mamdani, E. H. (1979). "**A Linguistic Self-Organising Process Controller**". **Automatica**, V. 15: 15-30.
- [26] YAMAZAKI, T. (1982). "**An Improved Algorithm for a Self-Organising Controller**". Ph. D. Thesis, Univ. London.
- [27] SCHARF, E.M. & Mandic, N.J., (1985). "**The Application of a Fuzzy Controller to the Control of a Multi-degree of Freedom Robot-Arm**". In: **Industrial Applications of Fuzzy Control**, M. Sugeno (Ed.), North-Holland: 41-61.
- [28] TAKAGI, T. & Sugeno, M., (1985). "**Fuzzy Identification of Systems and its Applications to Modelling and Control**". IEEE Trans. on Systems, Man & Cybernetics, Vol. IS: 116-132.

[29]SUGENO, M. & Kang, G.T., (1986). "**Fuzzy Modelling and Control of Multilayer Incinerator**". Fuzzy Sets and Systems, Vol. 18: 329-346.

[30] ALUMINIO, ALBRAS (2001). "**Fundamentos do Processo de Produção de Alumínio**". Parte I, 171-184.

[31] ANTUNES, M., P(2005)., "**Implementação de Algoritmos Fuzzy em Controladores Lógicos Programáveis**". Relatório de Iniciação Científica PIBIC 2000/2001, EFEI.

[32] ACOSTA, G (2004)., "**Taller de Controlador Basado en Lógica Difusa**". ARDECA- UNCPBA, Olavarría, Argentina.

[33] ROCKWELL AUTOMATION,(2006)."**Catalog Numbers 1785-L80E-Serie F**".

## ANEXO 1

### CÓDIGO FONTE DA IMPLEMENTAÇÃO DO CONTROLE FUZZY, EM LINGUAGEM C++

#### Código em C++ Defuzificador Áreas Sobrepostas

```
#define max(a,b) ((a)<(b)?(b):(a))
#define men(a,b) ((a)<(b)?(a):(b))
#define abs(a) ((a)>(0)?(a): (-a))

/* CONTROLADOR FUZZY PARA AQUECEDOR DE ÓLEO TÉRMICO*/
/* FUNÇÃO : CONTROLE DE TEMPERATURA DAS PLANTAS DE ANODO VERDE
DA ALBRAS*/
/* DATA: 15/04/2011 */
/* CRIADOR: JOSÉ GUILHERME C. ALMEIDA*/
/*COMPILADOR FUZZY 1.0 – EFEI –ESCOLA FEDERAL DE ENGENARIA DE
ITAJUBA*/

/* BLOCO 1 - INICIALIZACAO DOS INDICES DE PERTINENCIA */
/* - INICIALIZACAO DOS LIMITES DOS MEMBERSHIPS */

fuzzy(float *ex,float *Upx,float *v_ex)
{
float e; // erro
float Up; // incremento de controle
float v_e; // derivadda do erro
float Pes_Erro = 1;
float Pes_Ajuste = 1;
float Pes_variação do erro = 1;
```

float IPe\_Ajuste\_Ng = 0 ; // Índice de Pertinência Negativo Grande  
float Min\_Ajuste\_Ng = -1; // limite Min. da variável ling. Negativo Grande  
float Max\_Ajuste\_Ng = -0,5; // limite Max. da variável ling. Negativo Grande  
float IPe\_Ajuste\_Nm = 0 ; // Índice de Pertinência Negativo Médio  
float Min\_Ajuste\_Nm = -0,75; // limite Min. da variável ling. Negativo Médio  
float Max\_Ajuste\_Nm = -0,25; // limite Max. da variável ling. Negativo Médio  
float IPe\_Ajuste\_Np = 0 ; // Índice de Pertinência Negativo Pequeno  
float Min\_Ajuste\_Np = -0,5; // limite Min. da variável ling. Negativo Pequeno  
float Max\_Ajuste\_Np = 0; // limite Max. da variável ling. Negativo Pequeno  
float IPe\_Ajuste\_Pg = 0 ; // Índice de Pertinência Positivo Grande  
float Min\_Ajuste\_Pg = 0,5; // limite Min. da variável ling. Positivo Grande  
float Max\_Ajuste\_Pg = 1; // limite Max. da variável ling. Positivo Grande  
float IPe\_Ajuste\_Pm = 0 ; // Índice de Pertinência Positivo Médio  
float Min\_Ajuste\_Pm = 0,25; // limite Min. da variável ling. Positivo Médio  
float Max\_Ajuste\_Pm = 0,75; // limite Max. da variável ling. Positivo Médio  
float IPe\_Ajuste\_Pp = 0 ; // Índice de Pertinência Positivo Pequeno  
float Min\_Ajuste\_Pp = 0; // limite Min. da variável ling. Positivo Pequeno  
float Max\_Ajuste\_Pp = 0,5; // limite Max. da variável ling. Positivo Pequeno  
float IPe\_Ajuste\_Ze = 0 ; // Índice de Pertinência Zero  
float Min\_Ajuste\_Ze = -0,25; // limite Min. da variável ling. Zero  
float Max\_Ajuste\_Ze = 0,25; // limite Max. da variável ling. Zero  
float IPe\_Erro\_Ng = 0 ; // Índice de Pertinência Negativo Grande  
float Min\_Erro\_Ng = -1; // limite Min. da variável ling. Negativo Grande  
float Max\_Erro\_Ng = -0,5; // limite Max. da variável ling. Negativo Grande  
float IPe\_Erro\_Nm = 0 ; // Índice de Pertinência Negativo Médio  
float Min\_Erro\_Nm = -0,75; // limite Min. da variável ling. Negativo Médio  
float Max\_Erro\_Nm = -0,25; // limite Max. da variável ling. Negativo Médio  
float IPe\_Erro\_Np = 0 ; // Índice de Pertinência Negativo Pequeno  
float Min\_Erro\_Np = -0,5; // limite Min. da variável ling. Negativo Pequeno  
float Max\_Erro\_Np = 0; // limite limite Max., da variável ling. Negativo Pequeno  
  
float IPe\_Erro\_Pg = 0 ; // Índice de Pertinência Positivo Grande  
float Min\_Erro\_Pg = 0,5; // limite Min. da variável ling. Positivo Grande

```
float Max_Erro_Pg = 1; // limite Max. da variável ling. Positivo Grande
float IPe_Erro_Pm = 0 ; // Índice de Pertinência Positivo Médio
float Min_Erro_Pm = 0,25; // limite Min. da variável ling. Positivo Médio
float Max_Erro_Pm = 0,75; // limite Max.. da variável ling. Positivo Médio
float IPe_Erro_Pp = 0 ;// Índice de Pertinência Positivo Pequeno
float Min_Erro_Pp = 0; // limite Min. da variável ling. Positivo Pequeno
float Max_Erro_Pp = 0,5; // limite Max. da variável ling. Positivo Pequeno
float IPe_Erro_Ze = 0 ;// Índice de Pertinência Zero
float Min_Erro_Ze = -0,25; // limite Min. da variável ling. Zero
float Max_Erro_Ze = 0,25; // limite Max. da variável ling. Zero
float IPe_variação do erro_Ng = 0 ;// Índice de Pertinência Negativo Grande
float Min_variação do erro_Ng = -0,1; // limite inf. da variável ling. Negativo Grande
float Max_variação do erro_Ng = -0,05; // limite inf. da variável ling. Negativo Grande
float IPe_variação do erro_Nm = 0 ; //Índice de Pertinência Negativo Médio
float Min_variação do erro_Nm = -0,075; // limite Min.da variável ling. Negativo Médio
float Max_variação do erro_Nm = -0,025; //limite Max.da variável ling.Negativo Médio
float IPe_variação do erro_Np = 0 ;// Índice de Pertinência Negativo Pequeno
float Min_variação do erro_Np = -0,05; // limite Min. variável ling. Negativo Pequeno
float Max_variação do erro_Np = 0; // limite Max., variável ling. Negativo Pequeno
float IPe_variação do erro_Pg = 0 ;// Índice de Pertinência Positivo Grande
float Min_variação do erro_Pg = 0,05; // limite Min. da variável ling. Positivo Pequeno
float Max_variação do erro_Pg = 0,1; // limite Max. da variável ling. Positivo Pequeno
float IPe_variação do erro_Pm = 0 ;// Índice de Pertinência Positivo Médio
float Min_variação do erro_Pm = 0,025; //limite Min.da variável ling.Positivo Médio
float Max_variação do erro_Pm = 0,075; //limite Max.. variável ling. Positivo Médio
float IPe_variação do erro_Pp = 0 ;// Índice de Pertinência Positivo Pequeno
float Min_variação do erro_Pp = 0; // limite Min. da variável ling. Positivo Pequeno
float Max_variação do erro_Pp = 0,05; // limite Max. variável ling. Positivo Pequeno
float IPe_variação do erro_Ze = 0 ;// Índice de Pertinência Zero
float Min_variação do erro_Ze = -0,025; //limite Min. variável ling. Zero

float Max_variação do erro_Ze = 0,025; //limite Max. variável ling. Zero
float soma;
```

```
int flag,flag1;
float x1,x2,y1,y2,xmin,xmax,ymax,ymax1,yant;
float a1,a2;
float m1,m2;
float xx1,xm1,xx2,xm2,xx,xm;
float indice,ipert;
float x;
unsigned int base,resp;
int porta;
int canal= 0;
float retorno,matriz[50];
int elementos,count;
int cont,temp1,temp2,temp3,temp4;
float saidaana1,saidaana2;
```

```
e= *ex;
Up= *Upx;
v_e= *v_ex;
```

```
/* FIM DO BLOCO 1 */
```

```
printf("Entrada: e = ");
gets(pausa);
sscanf(pausa,"%f",&e);
printf("Entrada: v_e = ");
gets(pausa);
sscanf(pausa,"%f",&v_e);
```

/\* BLOCO 2 - CALCULO DOS INDICES DE PERTINENCIA \*/

if (e > -1 & e <= -0,75 )

if (IPe\_Erro\_Ng < ((-0,75-e)\*((1)-(1)))/((-0,75)-(-1))+1))

IPe\_Erro\_Ng = ((-0,75-e)\*((1)-(1)))/((-0,75)-(-1))+1);

if (e > -0,75 & e <= -0,5 )

if (IPe\_Erro\_Nm < ((e-(-0,75))\*((1)-(0)))/((-0,5)-(-0,75))+0))

IPe\_Erro\_Nm = ((e-(-0,75))\*((1)-(0)))/((-0,5)-(-0,75))+0);

if (e > -0,5 & e <= -0,25 )

if (IPe\_Erro\_Np < ((e-(-0,5))\*((1)-(0)))/((-0,25)-(-0,5))+0))

IPe\_Erro\_Np = ((e-(-0,5))\*((1)-(0)))/((-0,25)-(-0,5))+0);

if (e > -0,75 & e <= -0,5 )

if (IPe\_Erro\_Ng < ((-0,5-e)\*((1)-(0)))/((-0,5)-(-0,75))+0))

IPe\_Erro\_Ng = ((-0,5-e)\*((1)-(0)))/((-0,5)-(-0,75))+0);

if (e > -0,25 & e <= 0 )

if (IPe\_Erro\_Ze < ((e-(-0,25))\*((1)-(0)))/((0)-(-0,25))+0))

IPe\_Erro\_Ze = ((e-(-0,25))\*((1)-(0)))/((0)-(-0,25))+0);

if (e > -0,5 & e <= -0,25 )

if (IPe\_Erro\_Nm < ((-0,25-e)\*((1)-(0)))/((-0,25)-(-0,5))+0))

IPe\_Erro\_Nm = ((-0,25-e)\*((1)-(0)))/((-0,25)-(-0,5))+0);

if (v\_e > -0,1 & v\_e <= -0,075 )

if (IPe\_variação do erro\_Ng < ((-0,075-v\_e)\*((1)-(1)))/((-0,075)-(-0,1))+1))

IPe\_variação do erro\_Ng = ((-0,075-v\_e)\*((1)-(1)))/((-0,075)-(-0,1))+1);

if (v\_e > -0,075 & v\_e <= -0,05 )

if (IPe\_variação do erro\_Nm < ((v\_e-(-0,075))\*((1)-(0))/((-0,05)-(-0,075)))+(0))  
IPe\_variação do erro\_Nm = ((v\_e-(-0,075))\*((1)-(0))/((-0,05)-(-0,075)))+(0);

if (v\_e > -0,075 & v\_e <= -0,05 )  
if (IPe\_variação do erro\_Ng < ((-0,05-v\_e)\*((1)-(0))/((-0,05)-(-0,075)))+(0))  
IPe\_variação do erro\_Ng = ((-0,05-v\_e)\*((1)-(0))/((-0,05)-(-0,075)))+(0);

if (v\_e > -0,05 & v\_e <= -0,025 )  
if (IPe\_variação do erro\_Np < ((v\_e-(-0,05))\*((1)-(0))/((-0,025)-(-0,05)))+(0))  
IPe\_variação do erro\_Np = ((v\_e-(-0,05))\*((1)-(0))/((-0,025)-(-0,05)))+(0);

if (v\_e > -0,025 & v\_e <= 0 )  
if (IPe\_variação do erro\_Ze < ((v\_e-(-0,025))\*((1)-(0))/((0)-(-0,025)))+(0))  
IPe\_variação do erro\_Ze = ((v\_e-(-0,025))\*((1)-(0))/((0)-(-0,025)))+(0);

if (v\_e > -0,05 & v\_e <= -0,025 )  
if (IPe\_variação do erro\_Nm < ((-0,025-v\_e)\*((1)-(0))/((-0,025)-(-0,05)))+(0))  
IPe\_variação do erro\_Nm = ((-0,025-v\_e)\*((1)-(0))/((-0,025)-(-0,05)))+(0);

if (e > 0 & e <= 0,25 )  
if (IPe\_Erro\_Pp < ((e-(0))\*((1)-(0))/((0,25)-(0)))+(0))  
IPe\_Erro\_Pp = ((e-(0))\*((1)-(0))/((0,25)-(0)))+(0);

if (e > -0,25 & e <= 0 )  
if (IPe\_Erro\_Np < ((0-e)\*((1)-(0))/((0)-(-0,25)))+(0))  
IPe\_Erro\_Np = ((0-e)\*((1)-(0))/((0)-(-0,25)))+(0);

if (v\_e > -0,025 & v\_e <= 0 )  
if (IPe\_variação do erro\_Np < ((0-v\_e)\*((1)-(0))/((0)-(-0,025)))+(0))  
IPe\_variação do erro\_Np = ((0-v\_e)\*((1)-(0))/((0)-(-0,025)))+(0);

if (v\_e > 0 & v\_e <= 0,025 )  
if (IPe\_variação do erro\_Pp < ((v\_e-(0))\*((1)-(0))/((0,025)-(0)))+(0))

$I_{Pe\_variação\ do\ erro\_Pp} = ((v\_e - 0) * ((1) - (0)) / ((0,025) - (0))) + (0);$

if ( $v\_e > 0,025 \ \& \ v\_e \leq 0,05$  )

if ( $I_{Pe\_variação\ do\ erro\_Pm} < ((v\_e - 0,025) * ((1) - (0)) / ((0,05) - (0,025))) + (0)$ )

$I_{Pe\_variação\ do\ erro\_Pm} = ((v\_e - 0,025) * ((1) - (0)) / ((0,05) - (0,025))) + (0);$

if ( $v\_e > 0 \ \& \ v\_e \leq 0,025$  )

if ( $I_{Pe\_variação\ do\ erro\_Ze} < ((0,025 - v\_e) * ((1) - (0)) / ((0,025) - (0))) + (0)$ )

$I_{Pe\_variação\ do\ erro\_Ze} = ((0,025 - v\_e) * ((1) - (0)) / ((0,025) - (0))) + (0);$

if ( $v\_e > 0,025 \ \& \ v\_e \leq 0,05$  )

if ( $I_{Pe\_variação\ do\ erro\_Pp} < ((0,05 - v\_e) * ((1) - (0)) / ((0,05) - (0,025))) + (0)$ )

$I_{Pe\_variação\ do\ erro\_Pp} = ((0,05 - v\_e) * ((1) - (0)) / ((0,05) - (0,025))) + (0);$

if ( $v\_e > 0,05 \ \& \ v\_e \leq 0,075$  )

if ( $I_{Pe\_variação\ do\ erro\_Pg} < ((v\_e - 0,05) * ((1) - (0)) / ((0,075) - (0,05))) + (0)$ )

$I_{Pe\_variação\ do\ erro\_Pg} = ((v\_e - 0,05) * ((1) - (0)) / ((0,075) - (0,05))) + (0);$

if ( $v\_e > 0,05 \ \& \ v\_e \leq 0,075$  )

if ( $I_{Pe\_variação\ do\ erro\_Pm} < ((0,075 - v\_e) * ((1) - (0)) / ((0,075) - (0,05))) + (0)$ )

$I_{Pe\_variação\ do\ erro\_Pm} = ((0,075 - v\_e) * ((1) - (0)) / ((0,075) - (0,05))) + (0);$

if ( $v\_e > 0,075 \ \& \ v\_e \leq 0,1$  )

if ( $I_{Pe\_variação\ do\ erro\_Pg} < ((v\_e - 0,075) * ((1) - (1)) / ((0,1) - (0,075))) + (1)$ )

$I_{Pe\_variação\ do\ erro\_Pg} = ((v\_e - 0,075) * ((1) - (1)) / ((0,1) - (0,075))) + (1);$

if ( $e > 0,25 \ \& \ e \leq 0,5$  )

if ( $I_{Pe\_Erro\_Pm} < ((e - 0,25) * ((1) - (0)) / ((0,5) - (0,25))) + (0)$ )

$I_{Pe\_Erro\_Pm} = ((e - 0,25) * ((1) - (0)) / ((0,5) - (0,25))) + (0);$

if ( $e > 0 \ \& \ e \leq 0,25$  )

if ( $I_{Pe\_Erro\_Ze} < ((0,25 - e) * ((1) - (0)) / ((0,25) - (0))) + (0)$ )

$I_{Pe\_Erro\_Ze} = ((0,25 - e) * ((1) - (0)) / ((0,25) - (0))) + (0);$

```
if (e > 0,25 & e <= 0,5 )
```

```
if (IPe_Erro_Pp < ((0,5-e)*((1)-(0))/((0,5)-(0,25)))+(0))
```

```
IPe_Erro_Pp = ((0,5-e)*((1)-(0))/((0,5)-(0,25)))+(0);
```

```
if (e > 0,5 & e <= 0,75 )
```

```
if (IPe_Erro_Pg < ((e-(0,5))*((1)-(0))/((0,75)-(0,5)))+(0))
```

```
IPe_Erro_Pg = ((e-(0,5))*((1)-(0))/((0,75)-(0,5)))+(0);
```

```
if (e > 0,75 & e <= 1 )
```

```
if (IPe_Erro_Pg < ((e-(0,75))*((1)-(1))/((1)-(0,75)))+(1))
```

```
IPe_Erro_Pg = ((e-(0,75))*((1)-(1))/((1)-(0,75)))+(1);
```

```
if (e > 0,5 & e <= 0,75 )
```

```
if (IPe_Erro_Pm < ((0,75-e)*((1)-(0))/((0,75)-(0,5)))+(0))
```

```
IPe_Erro_Pm = ((0,75-e)*((1)-(0))/((0,75)-(0,5)))+(0);
```

```
/* FIM DO BLOCO 2 */
```

```
/* BLOCO 3 - APLICACAO DAS REGRAS */
```

```
if ((Min_Erro_Ng <= e & e <= Max_Erro_Ng) & (Min_variação do erro_Ng <= v_e &  
v_e <= Max_variação do erro_Ng))
```

```
{
```

```
matriz[0,1]=IPe_Erro_Ng,IPe_variação do erro_Ng;
```

```
elementos=2;
```

```
min (elementos,matriz,&retorno);
```

```
IPe_Ajuste_Pg = max(IPe_Ajuste_Pg ,retorno);
```

```
}
```

```
if ((Min_Erro_Ng <= e & e <= Max_Erro_Ng) & (Min_variação do erro_Nm <= v_e &
v_e <= Max_variação do erro_Nm))
{
matriz[0,1]=Ipe_Erro_Ng,Ipe_variação do erro_Nm;
elementos=2;
min (elementos,matriz,&retorno);
Ipe_Ajuste_Pg = max(Ipe_Ajuste_Pg ,retorno);
}
```

```
if ((Min_Erro_Ng <= e & e <= Max_Erro_Ng) & (Min_variação do erro_Np <= v_e &
v_e <= Max_variação do erro_Np))
{
matriz[0,1]=Ipe_Erro_Ng,Ipe_variação do erro_Np;
elementos=2;
min (elementos,matriz,&retorno);
Ipe_Ajuste_Pg = max(Ipe_Ajuste_Pg ,retorno);
}
```

```
if ((Min_Erro_Ng <= e & e <= Max_Erro_Ng) & (Min_variação do erro_Ze <= v_e &
v_e <= Max_variação do erro_Ze))
{
matriz[0,1]=Ipe_Erro_Ng,Ipe_variação do erro_Ze;
elementos=2;
min (elementos,matriz,&retorno);
Ipe_Ajuste_Pg = max(Ipe_Ajuste_Pg ,retorno);
}
```

```
if ((Min_Erro_Ng <= e & e <= Max_Erro_Ng) & (Min_variação do erro_Pp <= v_e &
v_e <= Max_variação do erro_Pp))
{
```

```
matriz[0,1]=Ipe_Erro_Ng,Ipe_variação do erro_Pp;  
elementos=2;  
min (elementos,matriz,&retorno);  
Ipe_Ajuste_Pm = max(Ipe_Ajuste_Pm ,retorno);  
}
```

```
if ((Min_Erro_Ng <= e & e <= Max_Erro_Ng) & (Min_variação do erro_Pm <= v_e &  
v_e <= Max_variação do erro_Pm))  
{  
matriz[0,1]=Ipe_Erro_Ng,Ipe_variação do erro_Pm;  
elementos=2;  
min (elementos,matriz,&retorno);  
Ipe_Ajuste_Pp = max(Ipe_Ajuste_Pp ,retorno);  
}
```

```
if ((Min_Erro_Ng <= e & e <= Max_Erro_Ng) & (Min_variação do erro_Pg <= v_e &  
v_e <= Max_variação do erro_Pg))  
{  
matriz[0,1]=Ipe_Erro_Ng,Ipe_variação do erro_Pg;  
elementos=2;  
min (elementos,matriz,&retorno);  
Ipe_Ajuste_Ze = max(Ipe_Ajuste_Ze ,retorno);  
}
```

```
if ((Min_Erro_Nm <= e & e <= Max_Erro_Nm) & (Min_variação do erro_Ng <= v_e &  
v_e <= Max_variação do erro_Ng))  
{  
matriz[0,1]=Ipe_Erro_Nm,Ipe_variação do erro_Ng;  
elementos=2;  
min (elementos,matriz,&retorno);
```

```
IPe_Ajuste_Pg = max(IPe_Ajuste_Pg ,retorno);  
}
```

```
if ((Min_Erro_Nm <= e & e <= Max_Erro_Nm) & (Min_variação do erro_Nm <= v_e &  
v_e <= Max_variação do erro_Nm))  
{  
matriz[0,1]=IPe_Erro_Nm,IPe_variação do erro_Nm;  
elementos=2;  
min (elementos,matriz,&retorno);  
IPe_Ajuste_Pg = max(IPe_Ajuste_Pg ,retorno);  
}
```

```
if ((Min_Erro_Nm <= e & e <= Max_Erro_Nm) & (Min_variação do erro_Np <= v_e &  
v_e <= Max_variação do erro_Np))  
{  
matriz[0,1]=IPe_Erro_Nm,IPe_variação do erro_Np;  
elementos=2;  
min (elementos,matriz,&retorno);  
IPe_Ajuste_Pm = max(IPe_Ajuste_Pm ,retorno);  
}
```

```
if ((Min_Erro_Nm <= e & e <= Max_Erro_Nm) & (Min_variação do erro_Ze <= v_e &  
v_e <= Max_variação do erro_Ze))  
{  
matriz[0,1]=IPe_Erro_Nm,IPe_variação do erro_Ze;  
elementos=2;  
min (elementos,matriz,&retorno);  
IPe_Ajuste_Pm = max(IPe_Ajuste_Pm ,retorno);  
}
```

```
if ((Min_Erro_Nm <= e & e <= Max_Erro_Nm) & (Min_variação do erro_Pp <= v_e &
v_e <= Max_variação do erro_Pp))
{
matriz[0,1]=Ipe_Erro_Nm,Ipe_variação do erro_Pp;
elementos=2;
min (elementos,matriz,&retorno);
Ipe_Ajuste_Pp = max(Ipe_Ajuste_Pp ,retorno);
}
```

```
if ((Min_Erro_Nm <= e & e <= Max_Erro_Nm) & (Min_variação do erro_Pm <= v_e &
v_e <= Max_variação do erro_Pm))
{
matriz[0,1]=Ipe_Erro_Nm,Ipe_variação do erro_Pm;
elementos=2;
min (elementos,matriz,&retorno);
Ipe_Ajuste_Ze = max(Ipe_Ajuste_Ze ,retorno);
}
```

```
if ((Min_Erro_Nm <= e & e <= Max_Erro_Nm) & (Min_variação do erro_Pg <= v_e &
v_e <= Max_variação do erro_Pg))
{
matriz[0,1]=Ipe_Erro_Nm,Ipe_variação do erro_Pg;
elementos=2;
min (elementos,matriz,&retorno);
Ipe_Ajuste_Ze = max(Ipe_Ajuste_Ze ,retorno);
}
```

```
if ((Min_Erro_Np <= e & e <= Max_Erro_Np) & (Min_variação do erro_Ng <= v_e &
v_e <= Max_variação do erro_Ng))
```

```
{  
matriz[0,1]=Ipe_Erro_Np,Ipe_variação do erro_Ng;  
elementos=2;  
min (elementos,matriz,&retorno);  
Ipe_Ajuste_Pg = max(Ipe_Ajuste_Pg ,retorno);  
}
```

```
if ((Min_Erro_Np <= e & e <= Max_Erro_Np) & (Min_variação do erro_Nm <= v_e &  
v_e <= Max_variação do erro_Nm))  
{  
matriz[0,1]=Ipe_Erro_Np,Ipe_variação do erro_Nm;  
elementos=2;  
min (elementos,matriz,&retorno);  
Ipe_Ajuste_Pm = max(Ipe_Ajuste_Pm ,retorno);  
}
```

```
if ((Min_Erro_Np <= e & e <= Max_Erro_Np) & (Min_variação do erro_Np <= v_e &  
v_e <= Max_variação do erro_Np))  
{  
matriz[0,1]=Ipe_Erro_Np,Ipe_variação do erro_Np;  
elementos=2;  
min (elementos,matriz,&retorno);  
Ipe_Ajuste_Pm = max(Ipe_Ajuste_Pm ,retorno);  
}
```

```
if ((Min_Erro_Np <= e & e <= Max_Erro_Np) & (Min_variação do erro_Ze <= v_e &  
v_e <= Max_variação do erro_Ze))  
{  
matriz[0,1]=Ipe_Erro_Np,Ipe_variação do erro_Ze;  
elementos=2;
```

```
min (elementos,matriz,&retorno);
Ipe_Ajuste_Pp = max(Ipe_Ajuste_Pp ,retorno);
}
```

```
if ((Min_Erro_Np <= e & e <= Max_Erro_Np) & (Min_variação do erro_Pg <= v_e &
v_e <= Max_variação do erro_Pg))
{
matriz[0,1]=Ipe_Erro_Np,Ipe_variação do erro_Pg;
elementos=2;
min (elementos,matriz,&retorno);
Ipe_Ajuste_Np = max(Ipe_Ajuste_Np ,retorno);
}
```

```
if ((Min_Erro_Np <= e & e <= Max_Erro_Np) & (Min_variação do erro_Pp <= v_e &
v_e <= Max_variação do erro_Pp))
{
matriz[0,1]=Ipe_Erro_Np,Ipe_variação do erro_Pp;
elementos=2;
min (elementos,matriz,&retorno);
Ipe_Ajuste_Ze = max(Ipe_Ajuste_Ze ,retorno);
}
```

```
if ((Min_Erro_Np <= e & e <= Max_Erro_Np) & (Min_variação do erro_Pm <= v_e &
v_e <= Max_variação do erro_Pm))
{
matriz[0,1]=Ipe_Erro_Np,Ipe_variação do erro_Pm;
elementos=2;
min (elementos,matriz,&retorno);
Ipe_Ajuste_Ze = max(Ipe_Ajuste_Ze ,retorno);
}
```

```
/* FIM DO BLOCO 3 */
```

```
/* INICIO DO BLOCO 4 - DEFUZZYFICACAO*/
```

```
xx = 0;
```

```
xm = 0;
```

```
x1 = -0,75;
```

```
x2 = -1;
```

```
y1 = 1;
```

```
y2 = 1;
```

```
indice= IPe_Ajuste_Ng;
```

```
if (indice > 0)
```

```
{
```

```
xm1 = abs(indice*(x2-x1));
```

```
xx1 = (x2+x1)/2;
```

```
xx = (xm*xx + xm1*xx1)/(xm+xm1);
```

```
xm = xm+xm1;
```

```
}
```

```
x1 = -0,5;
```

```
x2 = -0,75;
```

```
y1 = 0;
```

```
y2 = 1;
```

```
indice= IPe_Ajuste_Ng;
```

```
if (indice > 0)
```

```
{
```

```
x = x1 - indice*(x1-x2);
```

```
xm1 = indice*(-2*x2+x1+x)/2;
```

```
m1 = abs((x-x2)*indice);
```

```

m2=abs((x1-x)*indice/2);
a2=x+((x1-x)/3);
a1=x2+(x-x2)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
x1 =-0,75;
x2 =-0,5;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Nm;

```

```

if (indice > 0)
{
x = indice*(x2-x1)+ x1;
m1 = abs((x-x1)*indice/2);
m2=abs((x2-x)*indice);
a1=x1+(2*(x-x1)/3);
a2=x+(x2-x)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
x1 =-0,25;
x2 =-0,5;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Nm;

```

```

if (indice > 0)
{
x = x1 - indice*(x1-x2);
xm1 = indice*(-2*x2+x1+x)/2;
m1 = abs((x-x2)*indice);
m2=abs((x1-x)*indice/2);
a2=x+((x1-x)/3);
a1=x2+(x-x2)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
x1 =-0,5;
x2 =-0,25;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Np;

```

```

if (indice > 0)
{
x = indice*(x2-x1)+ x1;
m1 = abs((x-x1)*indice/2);
m2=abs((x2-x)*indice);
a1=x1+(2*(x-x1)/3);
a2=x+(x2-x)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}

```

```
x1 =0;
x2 =-0,25;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Np;
```

```
if (indice > 0)
{
x = x1 - indice*(x1-x2);
xm1 = indice*(-2*x2+x1+x)/2;
m1 = abs((x-x2)*indice);
m2=abs((x1-x)*indice/2);
a2=x+((x1-x)/3);
a1=x2+(x-x2)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
```

```
x1 =0,5;
x2 =0,75;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Pg;
```

```
if (indice > 0)
{
x = indice*(x2-x1)+ x1;
m1 = abs((x-x1)*indice/2);
m2=abs((x2-x)*indice);
a1=x1+(2*(x-x1)/3);
```

```

a2=x+(x2-x)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
x1 =0,75;
x2 =1;
y1 =1;
y2 =1;
indice= IPe_Ajuste_Pg;

```

```

if (indice > 0)
{
xm1 = abs(indice*(x2-x1));
xx1 = (x2+x1)/2;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
x1 =0,25;
x2 =0,5;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Pm;

```

```

if (indice > 0)
{
x = indice*(x2-x1)+ x1;
m1 = abs((x-x1)*indice/2);
m2=abs((x2-x)*indice);
a1=x1+(2*(x-x1)/3);

```

```

a2=x+(x2-x)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
x1 =0,75;
x2 =0,5;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Pm;

```

```

if (indice > 0)
{
x = x1 - indice*(x1-x2);
xm1 = indice*(-2*x2+x1+x)/2;
m1 = abs((x-x2)*indice);
m2=abs((x1-x)*indice/2);
a2=x+((x1-x)/3);
a1=x2+(x-x2)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
x1 =0;
x2 =0,25;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Pp;

```

```

if (indice > 0)
{
x = indice*(x2-x1)+ x1;
m1 = abs((x-x1)*indice/2);
m2=abs((x2-x)*indice);
a1=x1+(2*(x-x1)/3);
a2=x+(x2-x)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
x1 =0,5;
x2 =0,25;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Pp;

```

```

if (indice > 0)
{
x = x1 - indice*(x1-x2);
xm1 = indice*(-2*x2+x1+x)/2;
m1 = abs((x-x2)*indice);
m2=abs((x1-x)*indice/2);
a2=x+((x1-x)/3);
a1=x2+(x-x2)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
x1 =-0,25;

```

```
x2 =0;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Ze;
```

```
if (indice > 0)
{
x = indice*(x2-x1)+ x1;
m1 = abs((x-x1)*indice/2);
m2=abs((x2-x)*indice);
a1=x1+(2*(x-x1)/3);
a2=x+(x2-x)/2;
xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
```

```
x1 =0,25;
x2 =0;
y1 =0;
y2 =1;
indice= IPe_Ajuste_Ze;
```

```
if (indice > 0)
{
x = x1 - indice*(x1-x2);
xm1 = indice*(-2*x2+x1+x)/2;
m1 = abs((x-x2)*indice);
m2=abs((x1-x)*indice/2);
a2=x+((x1-x)/3);
a1=x2+(x-x2)/2;
```

```

xm1=m1+m2;
xx1=((a1*m1)+(a2*m2))/xm1;
xx = (xm*xx + xm1*xx1)/(xm+xm1);
xm = xm+xm1;
}
printf("resposta Ajuste = %f \n", xx);
gets (pausa);

}
/* FIM DO SISTEMA */

```

```

min(int elementos,float matriz[],float *retorno)
{
//numero de elementos
//matriz = matriz dos indices de pertinencia
//retorno = endereco onde retornara o valor
int i;
*retorno=9999999;
for(i=0;i<elementos;i++)
if(matriz[i]<*retorno)
*retorno=matriz[i];
}

```

```

leportaana(int n,unsigned int base,float *leitura)
{
unsigned int dadoh, dado1;
outportb(base+11,0x01); //seleciona modo de controle de registro
//01 - software trigger
//0x10 - pacer trigger
//outportb(base+10,n); seleciona porta de leitura
outportb(base+9,0); // ajusta ganho de entrada
outportb(base+12,0);

```

```

    outportb(base+10,n);
    dadoh=0xff;
    do{
    dadoh=inportb(base+5);
    }while(dadoh>0x0f);
    dado1=inportb(base+4);
    *leitura=(dadoh << 8)+ (dado1);
    *leitura=(*leitura/409.5)-5;
    }
    geraana(int canal,unsigned int base,float *valor)
    {
    int f;
    unsigned int x;
    *valor=*valor*819;
    x=*valor;
    if(canal==1)
    {
    //for(f=0;f<1;f++)
    //{
    outportb(base+4,x & 0x00ff);
    outportb(base+5,(x & 0x0f00) >> 8);
    //}
    }
    if(canal==2)
    {
    outportb(base+6,x & 0x00ff);
    outportb(base+7,(x & 0x0f00) >> 8);
    }
    }
}

```