

ecai

Universidade Federal de Santa
Catarina
Centro Tecnológico
Curso de Engenharia de Controle e
Automação Industrial

ufsc

Controle Fuzzy de Temperatura em Túnel de Aquecimento

*Monografia submetida à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:
EEL 5901: Projeto de Fim de Curso*

Priscila P. de P. e Souza

Florianópolis, setembro de 1996

Controle Fuzzy de Temperatura em Túnel de Aquecimento

Priscila P. de P. e Souza

Esta monografia foi julgada no contexto da disciplina
EEL 5901: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação Industrial

Banca Examinadora:

Czeslau L. Barczak
Orientador

Prof. Augusto Humberto Bruciapaglia
Responsável pela disciplina e Coordenador do Curso

Prof. Guilherme Bittencourt, Avaliador

Gilberto da Cunha, Debatedor

Nicole S. Orfali, Debatedor

SUMÁRIO

RESUMO	1
ABSTRACT	1
1. INTRODUÇÃO	1
1.1 Objetivos	1
1.2 Justificativa	1
1.3 Revisão Bibliográfica	2
1.4 Descrição do Conteúdo	4
2. CONTROLE FUZZY	5
2.1 Introdução à Lógica Fuzzy	5
2.1.1 Grau de Pertinência	5
2.1.2 Função de Pertinência	7
2.1.3 Operação de Intersecção	8
2.1.4 Operação de União	9
2.2 Controlador Fuzzy	9
2.2.1 Fuzzyficação	10
2.2.2 Avaliação de Regras	11
2.2.3 Defuzzyficação	14
2.2.3.1 Centro de Gravidade	14
2.2.3.2 Média do Máximo	15
2.2.3.3 Máximo à Direita (à Esquerda)	15
2.2.3.4 Método da Altura	16
2.2.3.5 Centróide dos Máximos	16
2.2.4 Métodos de Geração da Base de Regras	17
2.2.5 Complexidade Computacional de Inferências Fuzzy	18
3. DETALHAMENTO DO TRABALHO	19
3.1 Constituição Física do Sistema	19
3.1.1 Túnel de Aquecimento	20
3.1.1.1 Transferência de Calor	20
3.1.2 Sensor de Temperatura	21
3.1.3 O Atuador PWM	21
3.1.3.1 Modulação PWM	22
3.1.4 A Placa RFC1-AT	22
3.2 Programação da Placa - Softwares	25
3.2.1 Programação em FC110 Assembly	26
3.2.2 A Rotina de Controle Fuzzy	26
3.3 O Controle usando Lógica Difusa	29
3.3.1 Definição de Entradas e Saídas	29
3.3.2 A Base de Conhecimento	30
3.3.2.1 Funções de Pertinência	31

3.3.2.2 Base de Regras	32
3.3.2.3 A Simulação do Sistema	32
4. RESULTADOS DOS ENSAIOS	34
4.1 Materiais e Métodos	34
4.2 Ensaios Dinâmicos	34
4.2.1 Simulação X Ensaios	34
4.2.2 Respostas Dinâmicas	36
4.2.3 Controlador PID X Controlador Fuzzy	37
5. CONCLUSÕES E RECOMENDAÇÕES	39
5.1 Conclusões do Trabalho	39
5.2 Recomendações para Trabalhos Futuros	40
6. REFERÊNCIAS BIBLIOGRÁFICAS	41
ANEXO A - Programa em Assembly	
ANEXO B - Programa FPL	
ANEXO C - Programa em “C”	

Resumo

O trabalho descreve um sistema fuzzy de controle de temperatura para um túnel de aquecimento sendo que o processamento do controle é feito por uma placa instalada em um computador hospedeiro. Esta placa faz a aquisição de dados e possui um microprocessador dedicado específico para aplicações em controle difuso.

A leitura de temperatura do túnel de aquecimento é obtida através de um termopar tipo "J" e o controle é feito regulando a largura de pulsos de uma potência entregue ao forno (PWM).

Abstract

This paper describes a fuzzy temperature control system for a plastic sheet heating tunnel, using a fuzzy card installed in a host microcomputer for the control processing. This card performs the data acquisition and contains a special dedicated fuzzy microprocessor.

A "J" thermocouple is used for the temperature measurement and a PWM amplifier control the rate of electric power to the heating element.

1. Introdução

1.1 Objetivos

Com este trabalho pretende-se estudar o controle automático de temperatura usando uma placa fabricada pela Togai especialmente projetada para aplicações em controle difuso. O projeto consiste em analisar as capacidades desta placa no controle da temperatura de um túnel de aquecimento e comparar os resultados com os obtidos por meio de controle PID.

1.2 Justificativa

Este trabalho faz parte de uma linha de pesquisa do GRUCON-LHW (Grupo de Pesquisa e Treinamento em Comando Numérico e Automação Industrial - Laboratório de Hardware) que se refere ao estudo do controle Difuso aplicado à máquinas-ferramenta [Barczak 93] e processos.

A placa RESELEC FUZZY CONTROLLER- AT (RFC1-AT) foi adquirida pelo laboratório em 1995 e até então suas capacidades e limitações não tinham sido analisadas. Inicialmente, o projeto seria controlar a posição de uma mesa posicionadora de dois eixos (X-Y), mas por problemas na montagem de um sistema de medição optou-se por fazer ensaios de controle de temperatura usando um túnel de aquecimento que é parte de um trabalho de mestrado no LHW. A temperatura deste forno era inicialmente controlada por um PID, e achamos que seria interessante fazer algumas comparações entre as duas técnicas. O hardware usado em um caso e em outro não é o mesmo (a rotina de controle PID foi implementada internamente em um CLP e não utiliza a placa RFC1-AT, para aquisição de dados).

1.3 Revisão Bibliográfica

Falar em teoria de controle clássico, necessariamente significa realizar cálculos relativos a zeros e pólos de um dado sistema, dados estes obtidos através das funções de transferência que relacionam as saídas com as entradas do mesmo. As funções de transferência, por sua vez, nada mais são que o resultado da aplicação da transformada de Laplace (ou da transformada Z, no caso discreto) nas equações diferenciais (à diferenças) que definem este sistema. Estas transformadas, no entanto, são métodos operacionais que só podem ser usados em equações diferenciais (ou à diferenças) lineares [Ogata93][Normey 94] [Franklin 81].

A natureza não-linear do sistema, devido às complexas equações da transferência de calor entre o forno e o meio externo, obrigam o projetista do sistema de controle a linearizá-lo em torno de um ponto de funcionamento [Ogata93]. Isto leva a um controlador que funciona de maneira ótima apenas em uma pequena faixa de valores de referência, não permitindo variações significativas no “setpoint” e sendo pouco robusto a variações de parâmetros internos.

O controle Difuso é particularmente adequado a este tipo de problema: sistemas não-lineares, variantes no tempo ou onde não se dispõe da modelagem matemática ou esta é muito complexa [Zadeh 73] [Self 90]. Além disto, o controle difuso tem a capacidade de fornecer conclusões a partir de informações imprecisas, ambíguas, vagas, qualitativas [Barczak 93] [Simas 95] [Lee 90]. E porque considera a falta de exatidão e a incerteza do raciocínio humano, verifica-se uma poderosa ferramenta para manipular o conhecimento [Gomide 94]. “Precisão é cara...”, já dizia Ronald Yager, diretor do Instituto de Máquinas Inteligentes em Iona College, New Rochelle, N.Y., “... a lógica fuzzy é útil por causa da sua pequena especificidade, pois permite uma resposta mais flexível para uma dada entrada” [Arbex 94].

De modo a facilitar a implementação e acelerar a velocidade de avaliações de regras de um sistema baseado em lógica difusa, foram criados softwares e hardwares especiais. O primeiro microprocessador especialmente projetado para aplicações em controle difuso foi desenvolvido em 1985 nos laboratórios da AT&T por Togai e Watanabe . A partir dele, outros foram sendo desenvolvidos, versões mais rápidas e inovadoras, sendo que a velocidade com que as regras são

avaliadas é medida em FLIPS (inferências de lógica *fuzzy* por segundo). Geralmente são implementados em uma arquitetura RISC (Reduced Instruction Set Computer) e armazenam regras e funções-membro em RAM ou ROM externas ao chip [Self90]. Mais tarde, uma versão preliminar de um computador *fuzzy* foi proposta por Yamakawa (1986), sendo implementada pela Omron [Gomide 94].

Existem no mercado vários fabricantes de hardwares (placas) que utilizam microprocessadores desse tipo como sua principal força para desenvolver aplicações em controle difuso. Apesar de a grande maioria dos grupos de pesquisa em lógica difusa se encontrar no Japão, uma das mais importantes empresas especializadas é a Togai Infralogic Inc. (Irvine, Califórnia). Um dos produtos fabricados por ela é a placa RESELEC FUZZY CONTROLLER-AT (RFC1-AT) e o seu processador "FC100 Digital Fuzzy Processor". O FC110 é capaz de manipular 50.000 inferências *fuzzy* por segundo (50 KFLIPS), possui um conjunto reduzido de instruções (RISC) especiais e a base de conhecimento é armazenada em RAM externa [Self 90] [Togai91] [Togai93] [Silva95].

No caso de softwares, não é pequeno o número de ferramentas para desenvolvimento no mercado. Estes oferecem facilidades para a edição das funções de pertinência, regras, recursos gráficos, possibilidade de simulação, compiladores para geração de código para linguagem de alto nível. Como exemplos, pode-se citar: TILShell, da Togai [Togai 92a], Cubicalc da Hyperlogic, Manifold Editor da Fuzzy Systems Engineering, o ambiente integrado da FuzzyTECH, as ferramentas de desenvolvimento FIDE da Apronix, o SDAF da HI Tecnologia [Gomide 94].

Pela transparência do algoritmo de controle, a possibilidade de trabalhar em tempo real devido a simplicidade na realização de inferências e a rapidez conseguida por meio de hardwares especializados, o controle difuso é muito apropriado para aplicações industriais. As aplicações encontradas são as mais diversas possíveis, por exemplo: operação automática de trens (subway de Sendai), máquinas de lavar roupa (Sanyo), ar condicionado (Mitsubishi), elevadores (Toshiba), geladeiras, máquinas fotográficas (Sony), câmeras de vídeo [Wellstead][Self 90][Gomide94], controle de servomotores [Mandani 74] [Rosa 90a] [Li 89], controle de temperatura de reator [King, Mandani 76], controle de

temperatura de ar condicionado [Imaiida 90], aplicações em máquina ferramenta [Rosa 90b][Rosa 91][Barczak 93], sistemas de freios ABS.

1.4 Descrição do Conteúdo

No capítulo 1 são apresentados os objetivos do trabalho, a justificativa do mesmo e também uma breve introdução ao tema Controle *Fuzzy* e suas aplicações.

O capítulo 2 tem o objetivo de apresentar ao leitor as características da lógica nebulosa e mostrar como funciona um controlador Difuso.

O capítulo 3 pode ser considerado o capítulo central, mostra as características do sistema a ser controlado e que ferramentas estavam disponíveis para o trabalho de controle. A modelagem difusa do sistema é apresentada também neste capítulo.

No capítulo 4, serão apresentados os resultados obtidos com o sistema e a comparação com o sistema anterior, controlado por um PID. Os resultados serão discutidos e analisados.

Enfim, no capítulo 5, são mostradas as conclusões tiradas deste trabalho e recomendações para trabalhos posteriores.

O trabalho traz em anexo a listagem dos programas usados no projeto. O anexo A mostra o código do programa em Assembly usado para aquisição de dados, o anexo B mostra a listagem do programa FPL e o anexo C traz a listagem de um programa escrito em linguagem C.

2. Controle Fuzzy

2.1 Introdução à Lógica Fuzzy

A lógica difusa foi introduzida em 1965 por Lofti Zadeh da Universidade da Califórnia, Berkeley, e é baseada na teoria dos conjuntos difusos (“Fuzzy Sets”) [Zadeh65], e portanto fortemente baseada em teoria matemática. Em 1974, o prof. Mandani, Queen Mary College, após tentativas frustradas em controlar uma máquina a vapor com diversos tipos de controladores, conseguiu fazê-lo usando a lógica difusa. A partir daí a lógica difusa vem ganhando cada vez mais atenção e conquistando seu espaço na área do controle automático. Porque consegue manipular a imprecisão e a incerteza do raciocínio humano, a teoria dos conjuntos difusos tenta uma aproximação entre este raciocínio e a lógica executada pela máquina [Arbex 94].

2.1.1 Grau de Pertinência

Na teoria de conjuntos tradicionais, a pertinência de um elemento a um conjunto só pode ser “1” (pertence, \in) ou “0” (não-pertence, \notin), ou seja, a transição de um conjunto para o outro é brusca. Por exemplo, em relação ao conjunto dos números reais próximos de 3, o número 4,2 teria grau de pertinência “0” (\notin) e o número 3,8 teria grau de pertinência igual a “1” (\in) (figura 2.1a).

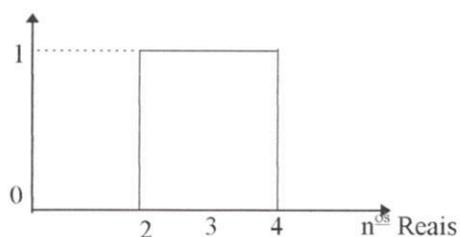


Fig 2.1a - Função de pertinência dos números reais ao conjunto não-difuso dos números próximos a 3.

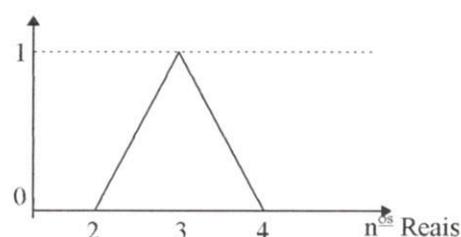


Fig. 2.1b - Função de pertinência dos números reais ao conjunto difuso dos números próximos a 3.

Por outro lado, na lógica difusa, o grau de pertinência de um elemento em relação a um conjunto pode variar entre “0” e “1”. Um elemento pode então pertencer “um pouco” a um determinado conjunto e pertencer “bastante” a outro conjunto, por exemplo. Assim, para expressar a idéia de um elemento pertencer ao conjunto dos números próximos de 3, por exemplo, pode-se usar uma função de pertinência triangular com o pico em 3. Deste modo, quanto mais perto de 3 estiver o tal número, maior é o grau de pertinência dele ao conjunto (figura 2.1b).

Importante: *Grau de pertinência não é probabilidade! “...é uma medida de compatibilidade do objeto com o conceito representado pelo conjunto fuzzy” [Arbex94].*

As variáveis, em lógica difusa, assumem expressões qualitativas, ao invés de valores numéricos. Para cada variável é atribuído um conjunto de termos lingüísticos (palavras, expressões ou frases) e a cada um deles é associado um subconjunto difuso (função-membro) num universo de discurso comum. Estas funções-membro podem ter várias formas, no entanto, nas aplicações práticas, geralmente são usadas formas triangulares ou trapezoidais, e normalmente se sobrepõem.

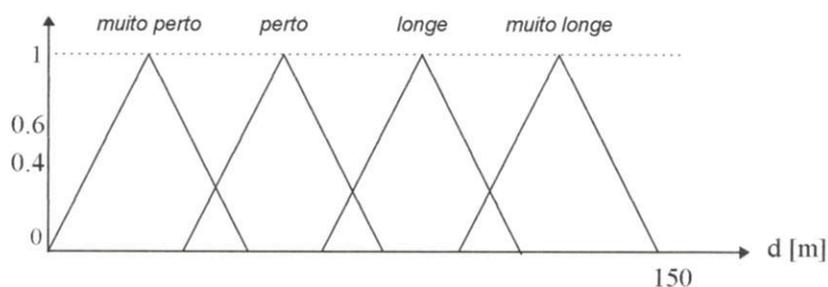


Fig. 2.2 - Funções de pertinência da variável distância. Graus de pertinência.

Na figura acima, a variável distância tem um universo de discurso $U:[0,150]$, um conjunto de quatro termos lingüísticos associados cada um a seus respectivos subconjuntos difusos {muito perto, perto, longe, muito longe}.

2.1.2 Função de Pertinência

“Um elemento X pode não pertencer ao subconjunto A ($\mu_A(x)$ exatamente 0), pode pertencer “um pouco” ao subconjunto A ($\mu_A(x)$ próximo de 0), pode ser “praticamente” elemento de A ($\mu_A(x)$ próximo de 1) ou pertencer inteiramente ao subconjunto A ($\mu_A(x)=1$)” [Simas 95].

Podemos, então, definir formalmente a função de pertinência. Considerando A como sendo um subconjunto difuso e U como sendo o universo de discurso (domínio) de uma variável, a função de pertinência μ_A pode ser definida como:

$$\mu_A: U \rightarrow [0,1]$$

A teoria dos conjuntos difusos envolve muitas definições e teoremas, algumas extremamente complexas, mas as definições descritas abaixo podem ser consideradas a base necessária ao desenvolvimento de estratégias de controle baseados em lógica difusa [Li 89] :

Seja A e B dois conjuntos difusos, e μ_A e μ_B suas funções de pertinência, respectivamente, então:

a) União de dois conjuntos $A \cup B$ (corresponde à função OU):

$$(A \cup B) = (A \text{ OU } B) = \mu_A(x) \wp \mu_B(x)$$

b) Intersecção entre dois conjuntos $A \cap B$ (corresponde à função E):

$$(A \cap B) = (A \text{ E } B) = \mu_A(x) \wp \mu_B(x)$$

c) Complemento do conjunto A (corresponde à função NÃO):

$$(\text{NÃO } A) = 1 - \mu_A(x)$$

Onde \wp é uma norma triangular (norma-t) e \wp é uma co-norma triangular (norma-s). Uma norma-t é uma função $\wp: [0,1] \times [0,1] \rightarrow [0,1]$ tal que $\forall x, y, z, w \in [0,1]$:

- I. $x \mathfrak{T} w \leq y \mathfrak{T} z$, se $x \leq y$, $w \leq z$
- II. $x \mathfrak{T} y = y \mathfrak{T} x$
- III. $(x \mathfrak{T} y) \mathfrak{T} z = x \mathfrak{T} (y \mathfrak{T} z)$
- IV. $x \mathfrak{T} 0 = 0$; $x \mathfrak{T} 1 = x$

Uma co-norma triangular (norma-s) é uma função $\wp: [0,1] \times [0,1] \rightarrow [0,1]$ que satisfaz as propriedades (I) a (III) acima e ainda :

$$\text{IV. } x \wp 0 = x ; x \wp 1 = 1$$

2.1.3 Operação de Intersecção

Como exemplos de norma-t temos:

- Mínimo de Zadeh(\wedge)

$$\mu_A \mathfrak{T} \mu_B = \text{Min} (\mu_A , \mu_B)$$

- Produto algébrico (\cdot)

$$\mu_A \mathfrak{T} \mu_B = \mu_A * \mu_B$$

- Média

$$\mu_A \mathfrak{T} \mu_B = (\mu_A + \mu_B) / 2$$

- Média ao Quadrado

$$\mu_A \mathfrak{T} \mu_B = ((\mu_A + \mu_B) / 2)^2$$

- Raiz da Média

$$\mu_A \mathfrak{T} \mu_B = ((\mu_A + \mu_B) / 2)^{1/2}$$

- Soma Limitada

$$\mu_A \mathfrak{T} \mu_B = \text{Máx} (0, \mu_A + \mu_B - 1)$$

- AND de Yanger

$$\mu_A \mathfrak{T} \mu_B = 1 - \text{Min}(1, ((1 - \mu_A)^k) + (1 - \mu_B)^k)^{1/k}) , \text{ onde } k \text{ é um parâmetro a}$$

ser determinado.

2.1.4 Operação de União

Como exemplos de norma-s podemos citar:

•Máximo de Zadeh (\vee)

$$\mu_A \wp \mu_B = \text{Máx} (\mu_A, \mu_B)$$

•Soma Limitada (\oplus)

$$\mu_A \wp \mu_B = \text{Min} (1, \mu_A + \mu_B)$$

•Produto

$$\mu_A \wp \mu_B = \mu_A + \mu_B - (\mu_A * \mu_B)$$

•OR de Yanger

$\mu_A \wp \mu_B = \text{Min} (1, \mu_A^K + \mu_B^K)^{1/K}$, onde k é um parâmetro a ser determinado.

2.2 Controlador Fuzzy

É importante ressaltar que as técnicas tradicionais de controle e as baseadas em lógica difusa não podem ser consideradas áreas “rivais”, e sim devem ser reconhecidas como áreas complementares, na medida que a lógica difusa vem a ser muito útil nos casos onde o controle clássico não encontrou solução. E ainda, soluções baseadas em lógica difusa não tem como principal objetivo a precisão das técnicas tradicionais, mas visa achar uma solução satisfatória em menor tempo [Costa 95].

Considerando que as entradas do controlador são valores numéricos vindos dos sensores e de cálculos algébricos feitos a partir deles, estas então devem ter seus valores adequados à lógica difusa (fuzzyficação) de modo que o sistema possa ser submetido a algumas condições (regras) programadas para que o sistema de controle possa agir do modo desejado. A ação qualitativa que resulta destas

condições deve então ser transformada em valores novamente numéricos (defuzzyficação) de modo a gerar uma ação de controle.

A figura 2.3 mostra as transformações nas variáveis durante o processo de controle [Barczak 93]:



Fig. 2.3 - Fluxograma - Controle Fuzzy

2.2.1 Fuzzyficação

É o processo de transformação do valor numérico das entradas (erro, velocidade, variação de temperatura, etc.) em graus de pertinência em relação às diferentes funções-membro de cada variável. Isto é feito escolhendo um conjunto de valores qualitativos de cada variável de entrada dentro do domínio desta variável (universo de discurso) e definindo as suas funções de pertinência.

Estes valores qualitativos de cada variável, assim como suas funções de pertinência associadas são definidos pelo projetista do sistema de controle tendo como informações os dados sobre as variáveis a serem controladas e as variáveis que serão as ações de controle. O número de conjuntos difusos para cada variável

leva a uma precisão melhor da variável, em geral, são usados conjuntos de três (positivo, zero e negativo) ou cinco conjuntos difusos (positivo grande, positivo pequeno, zero, negativo pequeno e negativo grande).

Por exemplo, em uma aplicação de controle de temperatura, a variável de entrada temperatura T pode ser dividida em três conjuntos de valores: negativa, zero e positiva (figura 2.4).

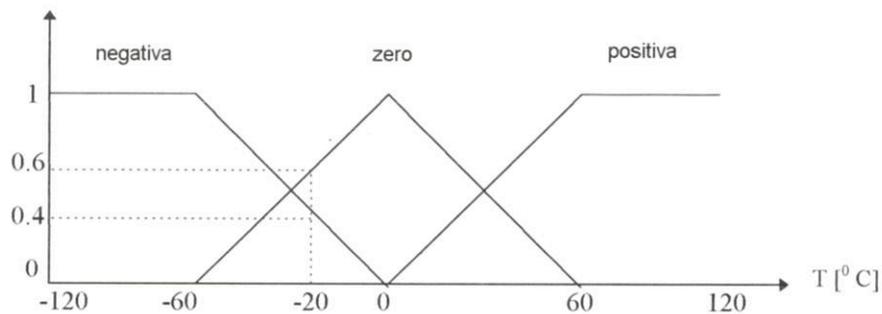


Fig. 2.4 - Funções de Pertinência para a Variável

De acordo com o gráfico, o valor -20° C está contido tanto no conjunto “negativa” como no conjunto “zero”, com graus de pertinência 0.6 e 0.4 respectivamente. As funções de pertinência são predefinidas e usadas em forma de tabelas, geralmente normalizados (graus de pertinência variam de 0 a 1).

2.2.2 Avaliação de Regras

O conhecimento utilizado para o controle de processos é expresso por meio de regras do tipo condição-ação. Estas condições descrevem o estado das variáveis observáveis do sistema e as ações são a saída de controle do sistema Fuzzy. O resultado desta operação será defuzzyficado de modo a transformar esta saída de controle em lógica difusa em saída de controle para o sistema real.

As regras tem a forma geral do tipo “Se ... então ...” . Do lado esquerdo ficam os antecedentes, ou seja, as condições que correspondem às entradas e do lado direito estão os conseqüentes, as ações resultantes da avaliação do conjunto de

condições e que correspondem às saídas do sistema. Por exemplo, algumas regras para um controlador de temperatura poderiam ter a forma:

Regra 1: **Se** (Temperatura é **Baixa**) **E** (Variação da Temperatura é **Grande Positiva**) **Então** Potência **Alta**

Regra 2 : **Se** (Temperatura é **Baixa**) **E** (Variação da Temperatura é **Pequena Negativa**) **Então** Potência **Muito Alta**

Regra 3 : **Se** (Temperatura é **Média**) **E** (Variação da Temperatura é **Grande Negativa**) **Então** Potência **Alta**

Regra 4 : **Se** (Temperatura é **Alta**) **E** (Variação da Temperatura é **Grande Negativa**) **Então** Potência **Zero**

A partir do conhecimento de como o sistema reage às diversas ações que possam ser aplicadas a ele, forma-se uma base de regras onde dita-se a ação necessária de acordo com cada estado possível do sistema. Para facilitar a visualização das ações de controle a partir dos estados do sistema e possível fusão de algumas regras pode-se usar uma tabela para representar a base de regras (figura 2.5).

		Temperatura		
		<i>BAIXA</i>	<i>MÉDIA</i>	<i>ALTA</i>
Variação da Temperatura	<i>GR_N</i>	M_ALTA	ALTA	MÉDIA
	<i>NEG</i>	M_ALTA	ALTA	BAIXA
	<i>ZERO</i>	M_ALTA	MÉDIA	M_BAIXA
	<i>POS</i>	ALTA	BAIXA	M_BAIXA
	<i>GR_P</i>	MÉDIA	BAIXA	M_BAIXA

Fig. 2.5 - Base de Regras

O processo de avaliar as regras difusas tendo como entrada os resultados da fuzzyficação dos dados numéricos, tendo como objetivo inferir as ações do controle difuso é chamado de procedimento de inferência.

Este processo se dá da seguinte forma: Após a avaliação do resultado da fuzzyficação usando a base de regras, obtém-se algumas regras cujas condições (antecedentes) foram verificados. Para cada uma dessas regras calcula-se um grau de pertinência associado a ela. Este grau de pertinência de cada regra é obtido aplicando a operação de intersecção (*AND Fuzzy*) entre os antecedentes. Assim, o resultado da operação são ações (algumas coincidentes) com seus respectivos graus de pertinência. Para cada conjunto de ações idênticas é feita uma operação de união (*OR Fuzzy*), para assim obter então, um conjunto de ações diferentes com graus de pertinência associados a elas.

O processo de inferência mais usado é o chamado "MÍN-MÁX", isto porque a intersecção é calculada pelo mínimo grau de pertinência (Mínimo de Zadeh) e a união é calculada pelo máximo grau de pertinência (Máximo de Zadeh) de cada variável.

2.2.3 Defuzzyficação

As saídas de controle difuso devem sofrer um processo de defuzzyficação para transformar os valores difusos em valores determinísticos que possam ser transformados em ações no processo. Pode-se dizer que é um “mapeamento do espaço de ações de controle difuso definido sobre o universo de discurso para o espaço de ações não-fuzzy ou escalares” [Arbex 94].

De modo geral, a saída de controle real é obtida usando as sub-funções da ação de controle difuso e os respectivos graus de pertinência associados a elas. Existem vários métodos para se realizar esta transformação, sendo que alguns deles são descritos abaixo:

2.2.3.1 Centro de Gravidade

A saída de controle, neste caso é um valor calculado a partir do centro de gravidade das figuras formadas pelas sub-funções da ação de controle limitadas pelos seus respectivos graus de pertinência associados.

Primeiramente, é calculado o ponto centróide X_i de cada figura; em seguida, um ponto é calculado pela média dos pontos centróides destas figuras, ponderadas pelo valor da área A_i de cada uma delas.

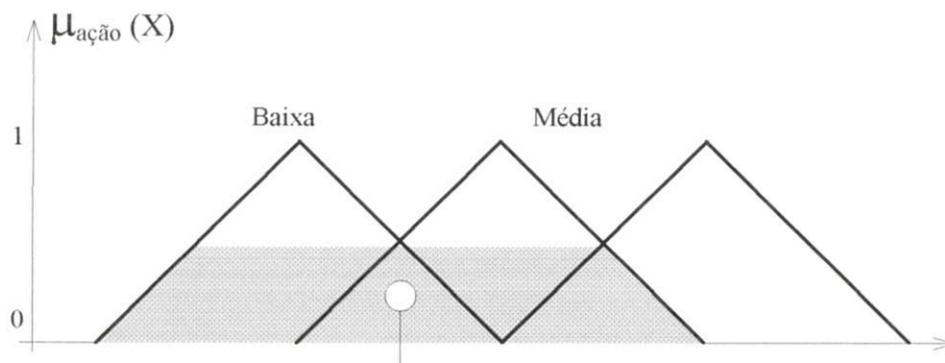


Fig. 2.6 - Método do Centróide

2.2.3.2 Média do Máximo

A saída é o valor associado ao ponto médio da base superior do trapézio formado pela limitação da sub-função com maior grau de pertinência.

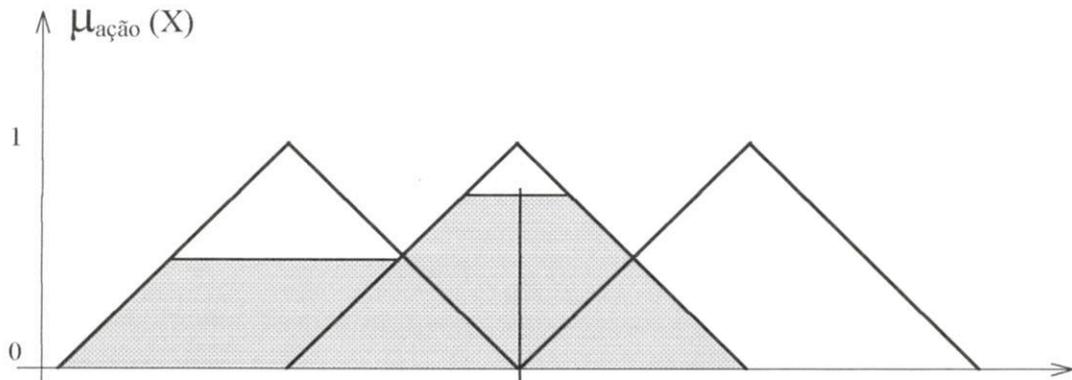


Fig. 2.7 - Método da Média do Máximo

2.2.3.3 Máximo à Direita (à Esquerda)

Na função de maior grau de pertinência, obtém-se o valor referente ao ponto superior mais à direita (à esquerda).

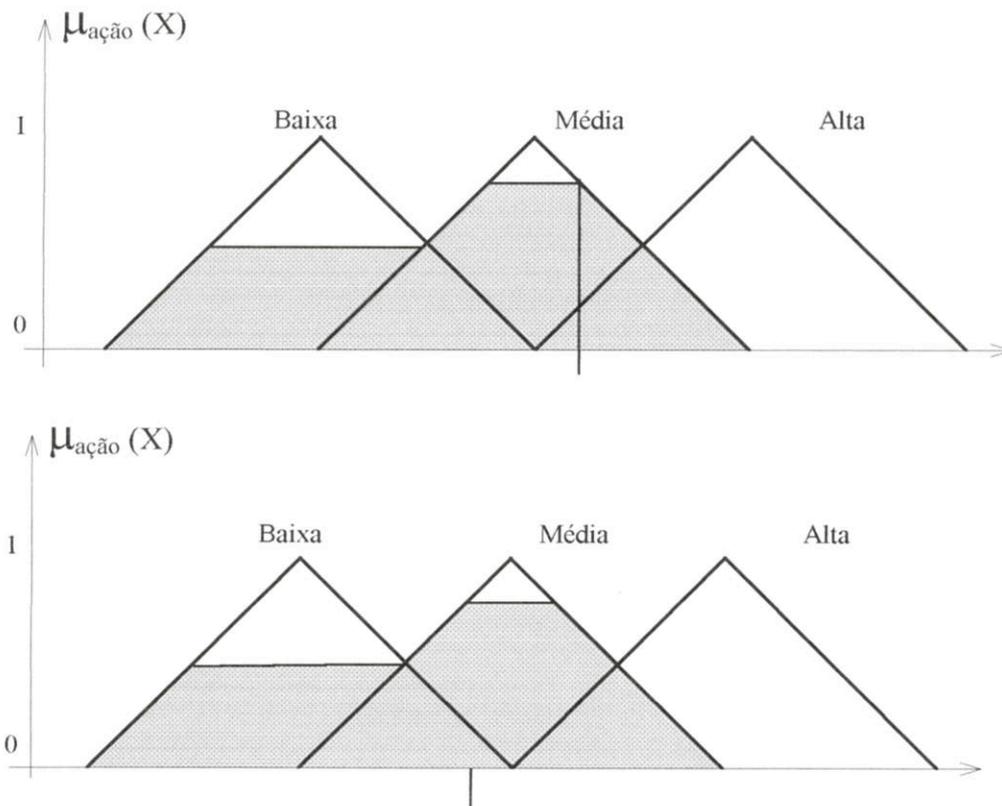


Fig. 2.8 - Métodos do Máximo à Direita (2.8a) e à Esquerda (2.8b)

2.2.3.4 Método da Altura

O valor é calculado pela seguinte fórmula:

$$D_h = \frac{\sum_{i=1}^n \alpha_i H_i C_i}{\sum_{i=1}^n \alpha_i H_i} \quad [2.1]$$

Onde, D_h é o valor de saída defuzzyficado

n é o número de conjuntos difusos da saída

α_i é o grau de pertinência de cada sub-função de saída

H_i é a altura máxima, para todo o universo de discurso

C_i é a abscissa do ponto centróide de cada sub-função fuzzy

2.2.3.5 Centróide dos Máximos

O método é similar ao do centróide, com a diferença de que só são usadas as duas sub-funções difusas com os maiores graus de pertinência associados.

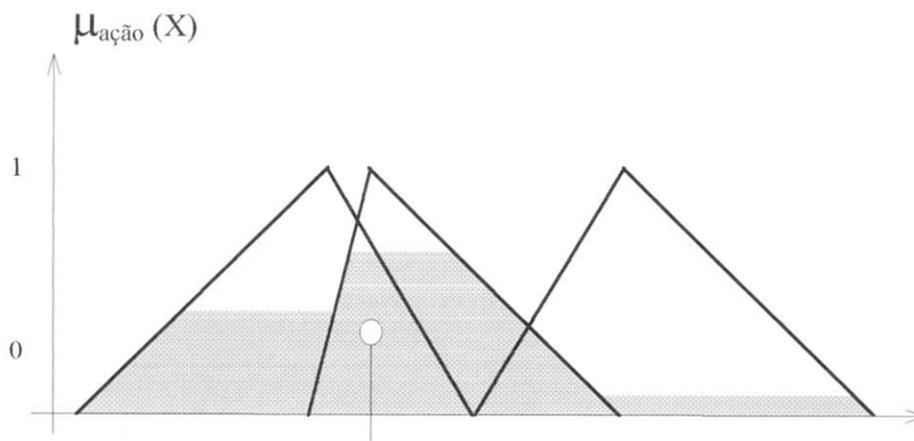


Fig. 2.10 - Método do Centróide dos Máximos

2.2.4 Métodos de Geração da Base de Regras

Um problema muito importante em controle difuso é como vai ser gerada a base de regras do sistema. Existem basicamente quatro maneiras diferentes de se obter estes dados [Lee 90]:

- *Baseado nas ações de um operador deste sistema* - Este método parte do princípio de que a partir do conhecimento prático do operador, seja possível gerar regras condicionais para atuar no sistema. O conhecimento subjetivo de operadores humanos capacitados pode controlar um sistema sem que seja preciso modelos matemáticos do processo.
- *Baseado em conhecimento de especialistas no processo e em engenharia de controle* - Considerando que decisões tomadas por humanos têm base em conhecimento de natureza lingüística, pode-se então, a partir do conhecimento destas pessoas obter regras para controle de um sistema. Isto pode ser feito através de questionários cuidadosamente elaborados.
- *A partir de uma modelagem difusa do sistema* - A lógica difusa pode ser usada também para fazer modelagem de sistemas, isto porque é possível descrever lingüisticamente o modo como o sistema se comporta diante de entradas e perturbações. E baseando-se nesta modelagem, é possível gerar regras para melhorar a dinâmica do sistema.
- *Automaticamente, através de aprendizado* - Algumas vezes, não se dispõe de um especialista no controle que possa auxiliar na escolha da base de regras, para evitar perda de tempo no ajuste do controlador, pode-se optar por um método de aprendizado. Existem vários métodos de ajuste de regras e muitos estudos são feitos neste sentido. Geralmente são usados redes neurais, algoritmos genéticos [Homaifar 95] [Ishibuchi 95], controladores Fuzzy adaptativos [Babaci 96], mas existem também métodos mais simples [Wang 92] e outros mais complexos frutos de estudos mais recentes (algoritmos evolucionários).

2.2.5 Complexidade Computacional de Inferências Fuzzy

Como visto anteriormente, o processo de inferência Fuzzy consiste em transformar as entradas, devidamente fuzzyficadas, em saídas de controle difuso por meio da aplicação de regras lingüísticas.

Este procedimento pode ser mais ou menos complexo em termos computacionais dependendo de alguns parâmetros fundamentais [Costa 95]:

- a) Número de entradas e saídas
- b) Número e forma das funções-membro para cada entrada e saída
- c) Número de regras
- d) Número de antecedentes e conseqüentes por regra
- e) Precisão requerida em termos de resolução
- f) Método de computação das funções de pertinência, método de inferência utilizado e método de defuzzyficação.

De acordo com o nível de complexidade do problema, deve-se escolher entre usar ou não softwares e hardware especializados, processadores RISC ou até ASIC's dedicados. As soluções são muitas, importante é precisar qual o nível de rapidez e flexibilidade desejado para o sistema em questão. Por exemplo, usar um software especializado pode facilitar a construção da estrutura do controlador, mas no entanto limita a flexibilidade em relação a programação do sistema, e aos métodos de inferência e de defuzzyficação que podem ser usados.

Pode-se dividir as soluções de implementação possíveis em quatro grupos:

- Software e hardware de propósito geral (não-especial)
- Processadores não-especiais com suporte fuzzy
- Coprocessadores Fuzzy dedicados
- ASIC's Fuzzy dedicados

3. Detalhamento do Trabalho

3.1 Constituição Física do Sistema

O sistema é formado por um forno, um sistema de aquecimento (resistências de tungstênio), um sensor de temperatura (termopar) e uma placa controladora com processador dedicado à aplicação, instalada em um computador hospedeiro (conectada ao barramento do PC, em um slot).

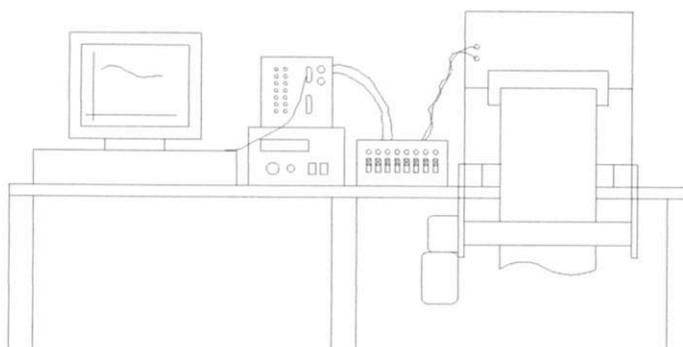


Fig. 3.1 - Figura do Esquema da Montagem

A temperatura no interior do túnel de aquecimento é medida usando-se um transmissor com termopar tipo J como elemento primário, este fornece um sinal em corrente e é convertido em tensão e, posteriormente, lido pela placa controladora através de uma entrada analógica. A saída de controle calculada pelo software no micro-computador é enviada para o sistema (saída analógica) e é convertida em sinal PWM (Pulse Width Modulator - Modulador de Largura de Pulso) que fornece a potência necessária ao forno (Fig. 3.2).

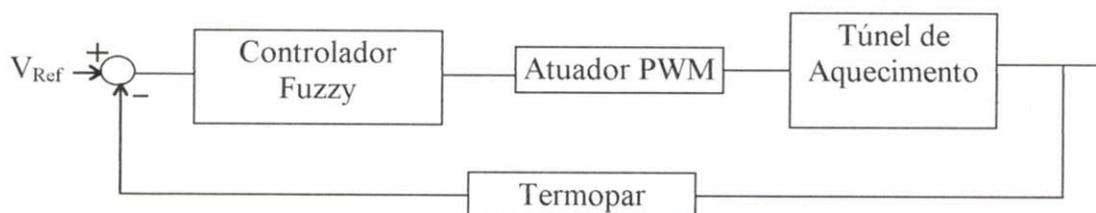


Fig. 3.2 - Diagrama de blocos da configuração do sistema

3.1.1 Túnel de Aquecimento

O túnel de aquecimento usado no projeto é um protótipo de um equipamento que tem a finalidade de aquecer uma manta de polietileno expandido para então esta ser conformada por uma prensa (termoformagem) [Luz Jr. 96].

A estrutura foi construída com chapas de aço de 1 mm de espessura e é revestida internamente por uma camada de manta isolante de fibra cerâmica de 25mm de espessura.

As dimensões internas, adotadas para o protótipo do túnel de aquecimento, foram as seguintes:

- Comprimento do túnel: 400 mm;
- Altura do túnel: 200 mm;
- Largura do túnel: 250 mm.

3.1.1.1 Transferência de calor

Existem três formas básicas de transferência de calor em fornos: radiação, convecção e condução que, geralmente, atuam concorrentemente. No sistema em questão, a transferência de calor interna (entre o forno e a manta de polietileno), é predominantemente por radiação infravermelha (filamentos de tungstênio). A figura 3.3 mostra um corte transversal do túnel de aquecimento usado como planta.

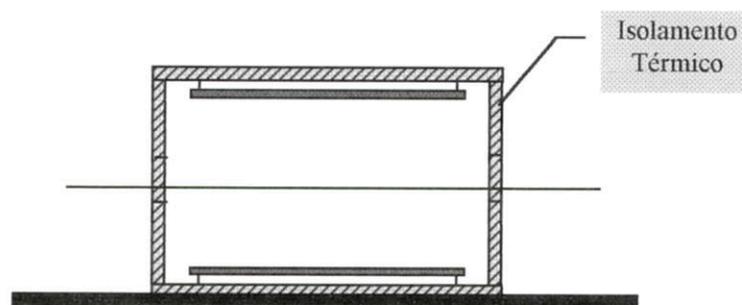


Fig. 3.3 - Corte Transversal do Túnel de Aquecimento

Em relação ao meio externo pode se considerar a transferência de calor por condução através das paredes do túnel de aquecimento e a perda de calor para o meio externo por convecção (circulação de ar) e irradiação. A modelagem matemática do sistema é complexa e as equações bastante não-lineares.

3.1.2 Sensor de Temperatura

O sensor utilizado para ler a temperatura interna do forno é um termopar do tipo "J". A faixa possível de utilização deste termopar é de -150°C a 1200°C com "span" mínimo de 150°C e máximo de 750°C . Inicialmente, o termopar estava calibrado para uma faixa de 0 a 400°C . No entanto, decidiu-se que para a aplicação de ensaios de controle, a faixa de utilização poderia ser diminuída e então o termopar foi recalibrado para uma faixa de 0 - 200°C .

A compensação da temperatura ambiente é feita eletronicamente em um módulo comprado juntamente com o termopar. A saída deste módulo é em corrente (4 - 20 mA) e precisou ser condicionada a faixa de tensão na entrada da placa (0 - 2,5 V) e isso foi feito por meio de um circuito a resistores.

3.1.3 O Atuador PWM

O fornecimento de energia ao forno é feito usando resistências de tungstênio e modulação por largura de pulso (PWM) (Fig. 3.4).

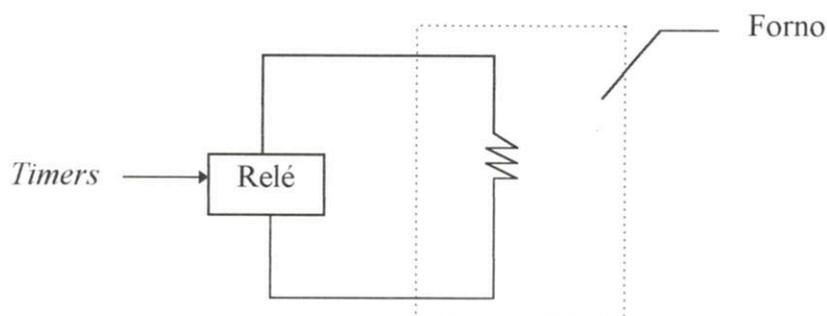
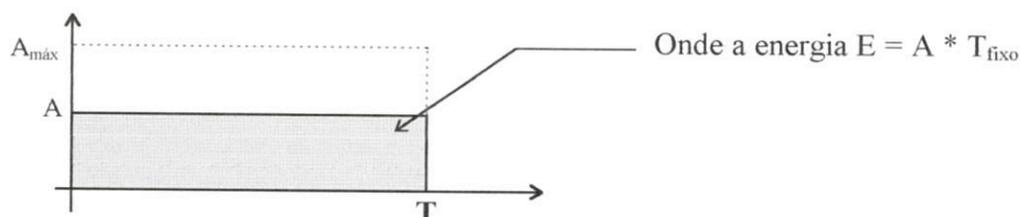


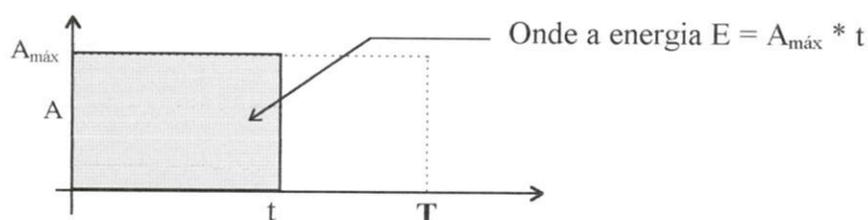
Fig. 3.4 - Esquema da montagem do modulador PWM

3.1.3.1 Modulação PWM

Ao contrário da modulação em amplitude a que estamos acostumados, a modulação PWM varia não a amplitude do pulso e sim a largura deste em uma amplitude fixa. Por exemplo, para uma mesma quantidade de energia a ser fornecida a um sistema, em um espaço de tempo definido T , um sinal modulado em amplitude teria esta forma:



Se a modulação do sinal fosse PWM:



3.1.4 A Placa RFC1-AT

Para aquisição de dados do processo e processamento do controle difuso foi utilizada a placa Reselec Fuzzy Controller (RFC1-AT) fabricada pela Togai Infralogic Inc.. Esta placa foi especialmente projetada para aplicações de controle difuso e pode ser usada como um módulo em um computador hospedeiro ou de modo independente ("stand alone").

O coração da placa é o "FC110 Digital Fuzzy Processor", um processador RISC dedicado que tem um conjunto especial de instruções. Este conjunto de instruções permitem ao FC110 processar mais rapidamente os problemas em controle difuso.

Além do processador, a placa dispõe de entradas analógicas e digitais, timers programáveis, possibilidade de comunicação serial e paralela, memória para base de conhecimento e suporte em linguagem de alto nível. O sistema foi projetado para desenvolver, testar e implementar uma aplicação baseada em controlador difuso.

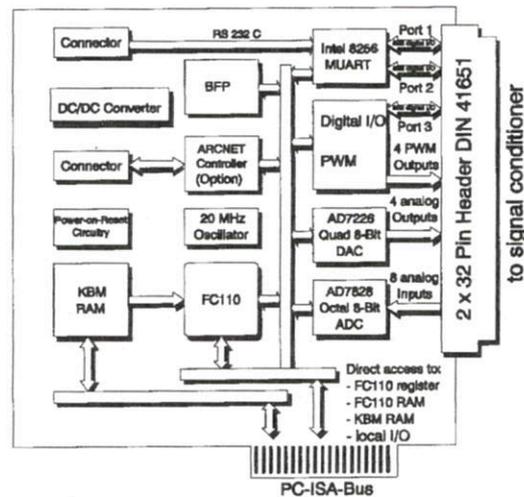


Fig. 3.5 - Diagrama de Blocos da Placa RFC1-AT

A figura 3.5 mostra um diagrama de blocos simplificado da placa RFC1-AT. Os principais blocos são descritos abaixo:

- O processador *fuzzy* FC110 é um microprocessador RISC com instruções especializadas.
- O bloco DAC representa 4 conversores digital/analógico de 8 bits, cada um com seu amplificador e seu "latch" para segurar o dado na saída.
- O bloco ADC consiste de um conversor analógico/digital com uma taxa de conversão de $2,5 \mu\text{s}$ por canal. Cada uma das 8 entradas podem ser endereçadas separadamente e usadas para ler sinais analógicos de um sistema externo.
- O Intel 8256 MUART (Multifunction Universal Asynchronous Receiver-Transmitter) faz o interfaceamento digital externo. Permite uma entrada serial externa e possui também duas portas (entrada/saída) de 8 bits que

podem se comunicar com o barramento interno. Possui também cinco timers programáveis de 8 bits e um controlador de interrupções com 8 níveis de prioridade e ainda um gerador de “baud rate”.

- A comunicação entre o processador FC110 e os outros módulos é feita por meio de um barramento multiplexado.

3.2 Programação da Placa - Softwares

A estrutura de software pode ser dividida em duas partes, um programa para leitura dos sensores, cálculos aritméticos e lógicos (não-difusos) e escrita dos dados de controle nas saídas, esta feita totalmente em linguagem FC110 assembly (o trabalho de programação é todo feito nesta parte), e uma outra parte onde é feita a rotina *Fuzzy* (usando o software TILShell+ e seu editor gráfico). A figura 3.6 dá uma idéia da estrutura de software do sistema [Togai93].

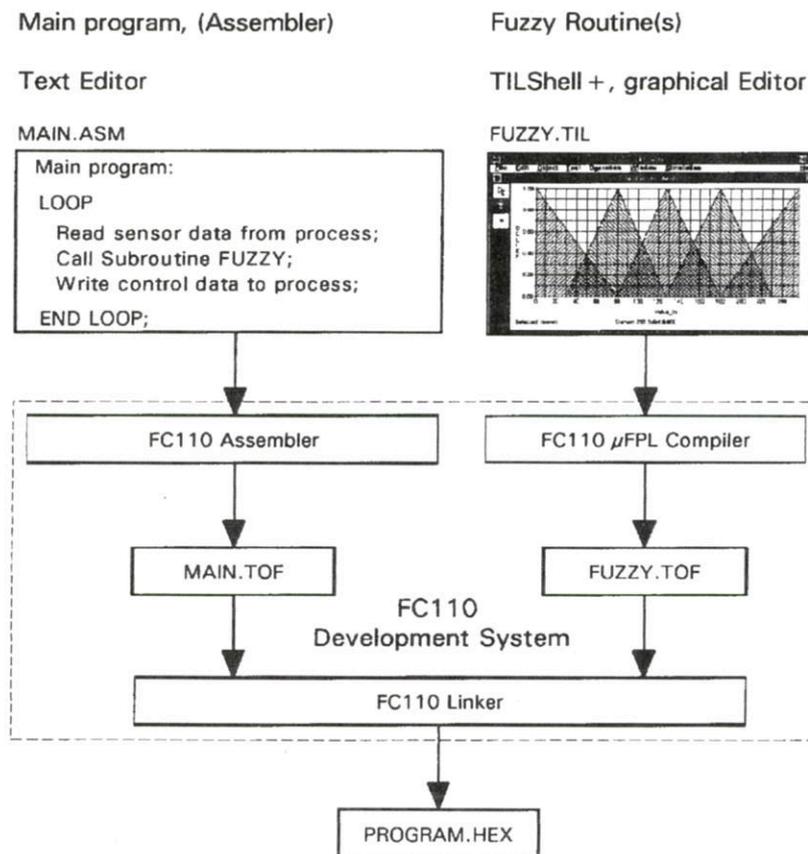


Fig. 3.6 - Estrutura de Software

3.2.1 Programação em FC110 Assembly

O FC110 é um processador RISC, o que significa dizer que tem um conjunto reduzido de instruções básicas, por isso facilita o trabalho de programação. A rotina Fuzzy é totalmente implementada fora do programa (editor gráfico) e só precisa ser chamada neste programa principal. Para facilitar o trabalho do programador alguns programas úteis são entregues junto com os discos para instalação (inicialização, utilitários e interrupções). Com estes programas, o programador pode se concentrar na sua aplicação tendo somente que acrescentar algumas linhas no programa para incluir tais facilidades. Abaixo pode ser visto o conjunto de instruções do FC110 Assembly (Tabela 3.1). O programa principal para aquisição de dados usado no projeto é mostrado no Anexo A .

Instrução	Descrição	Instrução	Descrição
add	soma c/carry	load	carrega um registro ou memória
cmp	comparação	mode	determina o modo em que vai operar o FC110
defuz	defuzzyficação	mult	multiplicação aritmética
div	divisão aritmética	nop	nenhuma operação
fzand	fuzzy AND	rhsc	avalia o lado direito de uma regra usando o "centróide"
fzor	fuzzy OR	rhsh	avalia o lado direito de uma regra usando a altura
halt	parada	rts	retorna de uma subrotina
jmp	salta para endereço	store	armazena em um registro ou memória
jsr	salta para subrotina	sub	subtração aritmética
lhs	avalia o lado esquerdo de uma regra		

3.2.2 A Rotina de Controle Fuzzy

O modo mais simples de implementar a parte *Fuzzy* de uma aplicação é usando as ferramentas gráficas do programa TILShell e o sistema de desenvolvimento FC110 da Togai (FC110DS) que consiste de um compilador (FC110 μ FPL Compiler), FC110 Assembler e FC110 Linker.

Para montar a base de conhecimento do sistema é utilizada a linguagem FPL (TIL Fuzzy Programming Language). Baseada em objetos, usa estruturas simples em blocos para descrever o sistema (ferramenta gráfica). Assim pode-se facilmente estruturar a base de regras, variáveis e funções de fuzzyficação. Isto pode ser feito usando a ferramenta gráfica TILShell ou usando um editor de texto qualquer. A listagem do programa FPL utilizada no trabalho é mostrada no Anexo B. O programa Togai TILShell também permite que o usuário faça simulação para testar o projeto. A figura 3.7 mostra a tela principal do programa TILShell.

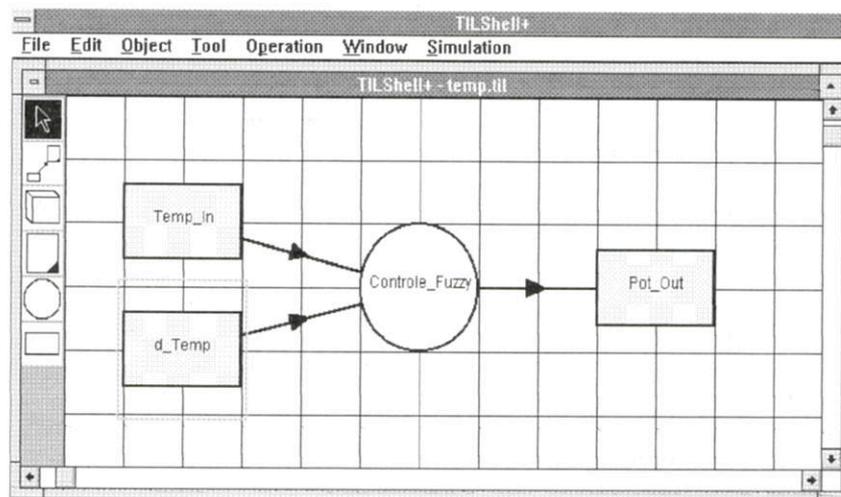


Fig. 3.7- Tela Principal do programa TILShell

Feito isto, deve-se então transformar esta base de conhecimento feita em linguagem de alto nível em uma linguagem que possa ser entendida pelo processador da placa RFC1-AT FC110. Isto pode ser feito usando o sistema de desenvolvimento FC110DS que consiste de três programas para gerar o programa .hex : FC110 FPL Compiler, FC110 Assembler e o FC110 Linker.

Estes programas podem ser chamados usando o Reselec Debug110 que é uma ligação entre o software (ferramenta TILShell) e o hardware (Reselec RFC1-AT). Permite ao usuário as seguintes operações:

- ⇒ Chamar o editor de texto para editar o programa principal em assembly;
- ⇒ Chamar o TIL FC110 Assembler;
- ⇒ Chamar TIL FC110 Linker;

- ⇒ Chamar o DownLoader LOAD110 para copiar a aplicação para a memória da placa;
- ⇒ Rodar as aplicações na placa e ainda inserir “breakpoints” e resetá-la;
- ⇒ Avaliar as entradas e saídas analógicas e digitais da placa.

A tela principal do programa onde podem ser monitoradas entradas, saídas e alguns registros, e de onde podem também ser chamados alguns programas é mostrada na figura 3.8.

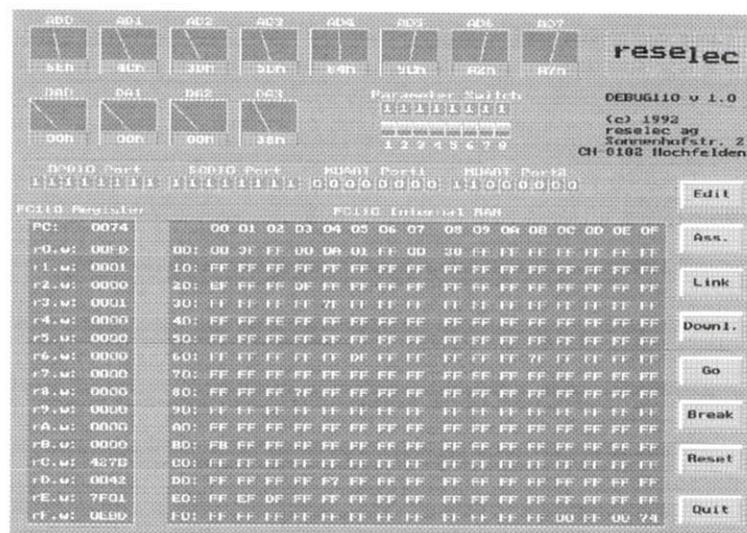


Fig. 3.8 - Tela do Programa DEBUG110

A principal desvantagem deste sistema de desenvolvimento é que o programa DEBUG110 só permite que se monitore o valor das entradas e saídas quando o programa não está rodando (“RESET” ou “BREAK”). Isto significa ter de parar a execução do programa de controle, por causa disto é preciso que se tenha um outro programa para verificar “on line” o estado das variáveis do sistema.

Foi implementado, então, um programa em linguagem “C” para leitura dos dados lidos pelas entradas analógicas da placa. A listagem deste programa pode ser encontrada no Anexo C.

3.3 O Controle usando Lógica Difusa

3.3.1 Definição de Entradas e Saídas

Em controle difuso monovariável é usual considerar que as entradas do controlador sejam o erro em relação ao valor de referência e a variação deste erro. No caso, as entradas do controlador são:

- o erro em relação à temperatura de referência

$$e(k) = T_R - T_F(k) \quad [3.1]$$

- a variação da temperatura durante o tempo de amostragem

$$-\Delta e(k) = -(e(k) - e(k-1)) = -(T_R - T_F(k)) + (T_R - T_F(k-1))$$

$$T_F(k) - T_F(k-1) = \Delta T_F(k) \quad [3.2]$$

Fazendo uma analogia aos esquema de controle clássico, a figura 3.9 mostra o diagrama de blocos no caso de controle difuso aplicado a processos.

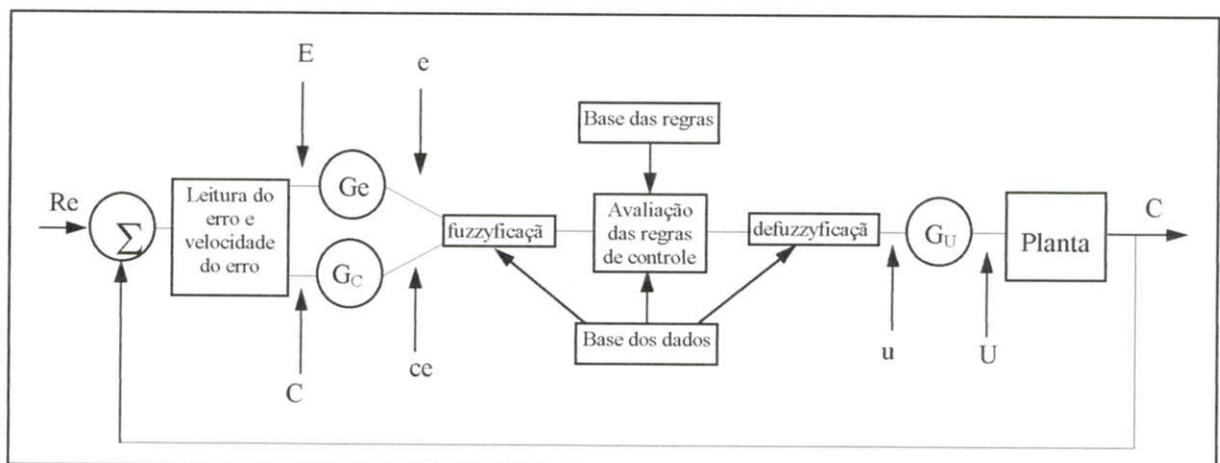


Fig. 3.9 - Esquema de Controle

A saída de controle inicialmente foi considerada como sendo um valor absoluto de potência entregue para o forno, mas logo verificou-se que esta arquitetura não permitia flexibilidade ao sistema na medida que o conjunto de funções de pertinência da saída *fuzzy* (Pot_Out) teria de ser alterado toda vez que fosse mudado o valor de referência. Isto porque o valor médio do conjunto (potência no regime permanente) é diferente para cada temperatura de referência.

Sendo assim, optou-se por um controlador difuso “integrador”, onde a saída é um valor que é somado à potência aplicada ao forno no instante anterior.

$$PotênciaOut = PotênciaAnterior + Soma_de_Potência$$

$$PotênciaAnterior = PotênciaOut$$

3.3.2 A Base de Conhecimento

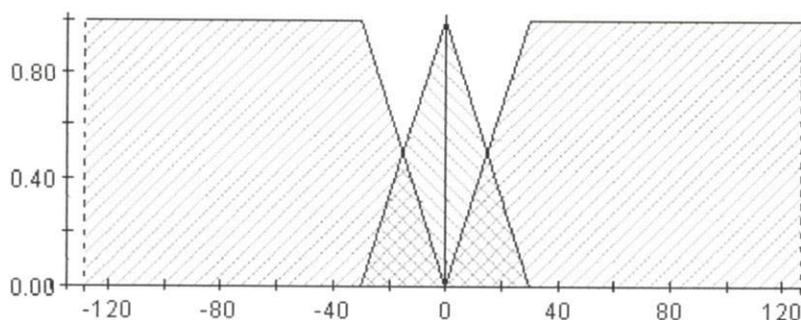
A estratégia de controle utilizada baseou-se primeiramente no comportamento da ação de controle exercida pelo controlador PID, com o objetivo de adaptá-la à lógica difusa e melhorá-la. A partir daí, verificou-se que, por causa do lento tempo de resposta, a ação de controle deve ser máxima até que a temperatura do forno se aproxime da temperatura de referência e então esta deve diminuir rapidamente e estabilizar em um valor. Com base em algumas observações deste tipo foi montada uma base de conhecimento para o sistema.

3.3.2.1 Funções de Pertinência

Foram definidas as seguintes funções de pertinência para as variáveis:

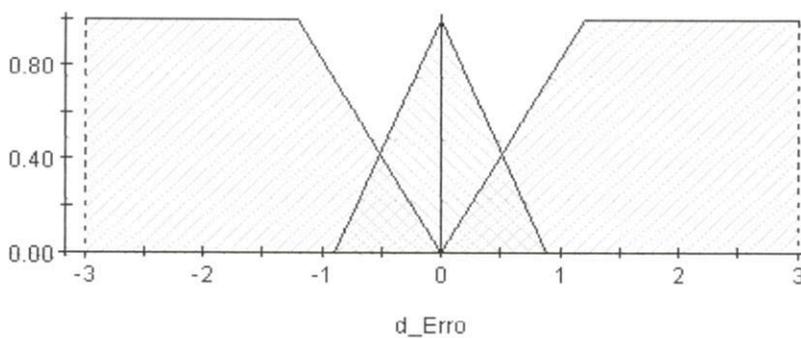
a) para a variável erro,

três conjuntos difusos : {negativo, zero, positivo}



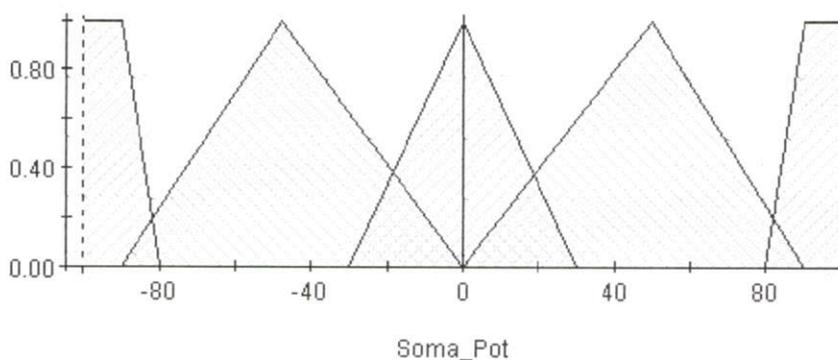
b) para a variável variação do erro,

três conjuntos difusos : {negativo, zero, positivo}



c) para a variável Soma_Potência,

cinco conjuntos difusos: {tira_muita, tira_pouca, mantém, soma_pouca, soma_muita}



3.3.2.2 Base de Regras

A base de regras escolhida para esta configuração de entradas e saídas pode ser vista abaixo, em forma de tabela:

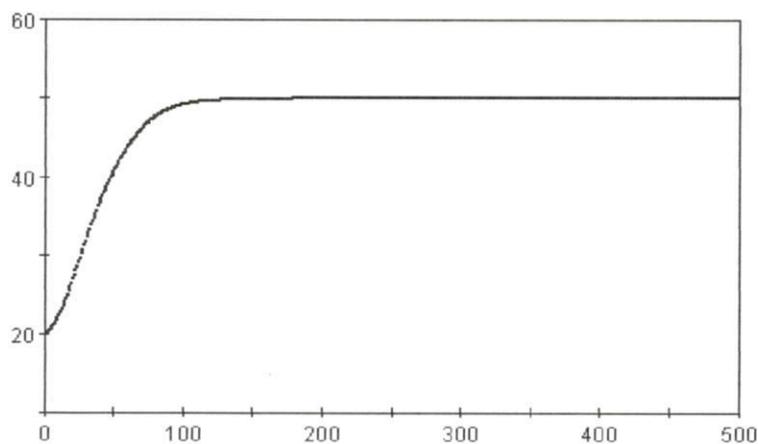
		Erro de Temperatura		
		NEG	ZERO	POS
Variação do Erro	NEG	Zero	Soma_pouca	Soma_muita
	ZERO	Tira_muita	Zero	Soma_muita
	POS	Tira_muita	Tira_pouca	zero

3.3.2.3 A Simulação do Sistema

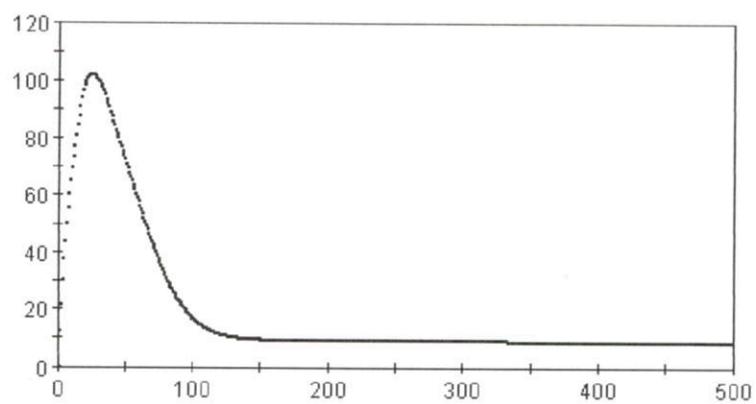
O programa TILShell da Togai permite simulação e teste do sistema. Então, após a determinação da base de conhecimento, foram feitos testes e ajustes do controlador em cima dos resultados.

Para esta configuração do controlador mostrada acima, obteve-se as seguintes respostas:

- Para a temperatura do forno,



- Para a saída de controle,



4. Resultados dos Ensaio

4.1 Materiais e Métodos

A montagem, como já foi dito anteriormente, é formada por um túnel de aquecimento, um sensor de temperatura (termopar tipo J), e uma placa Fuzzy conectada em um slot de um computador hospedeiro.

O software de controle pode ser dividido em duas partes, uma parte escrita em FC110 Assembly (leitura dos dados e cálculos não-difusos), e uma parte feita usando a FPL (Fuzzy Programming Language), onde se escreve a rotina fuzzy de controle.

Para a obtenção dos resultados a partir dos ensaios, foi preciso também um programa em linguagem "C", que faz as leituras referentes ao valor de temperatura que chega na placa e ao valor de potência enviada por esta à planta.

4.2 Ensaio Dinâmico

4.2.1 Simulação X Ensaio:

Os resultados obtidos através de simulação não se repetiram nos ensaios feitos em laboratório, visto que o modelo do sistema usado para esta simulação seja um modelo aproximado de 1ª ordem.

Na prática, verificou-se a necessidade de usar um conjunto maior de funções de pertinência para a variável de entrada "Erro_In", isto porque era importante que o sistema se comportasse de modo um pouco mais complexo quando o erro se aproximasse de zero. Um conjunto de 5 funções se mostrou suficiente no sistema em questão.

A seguir são mostradas as funções de pertinência das variáveis na configuração do controlador usando 5 funções de pertinência para a variável Erro (Fig. 4.1).

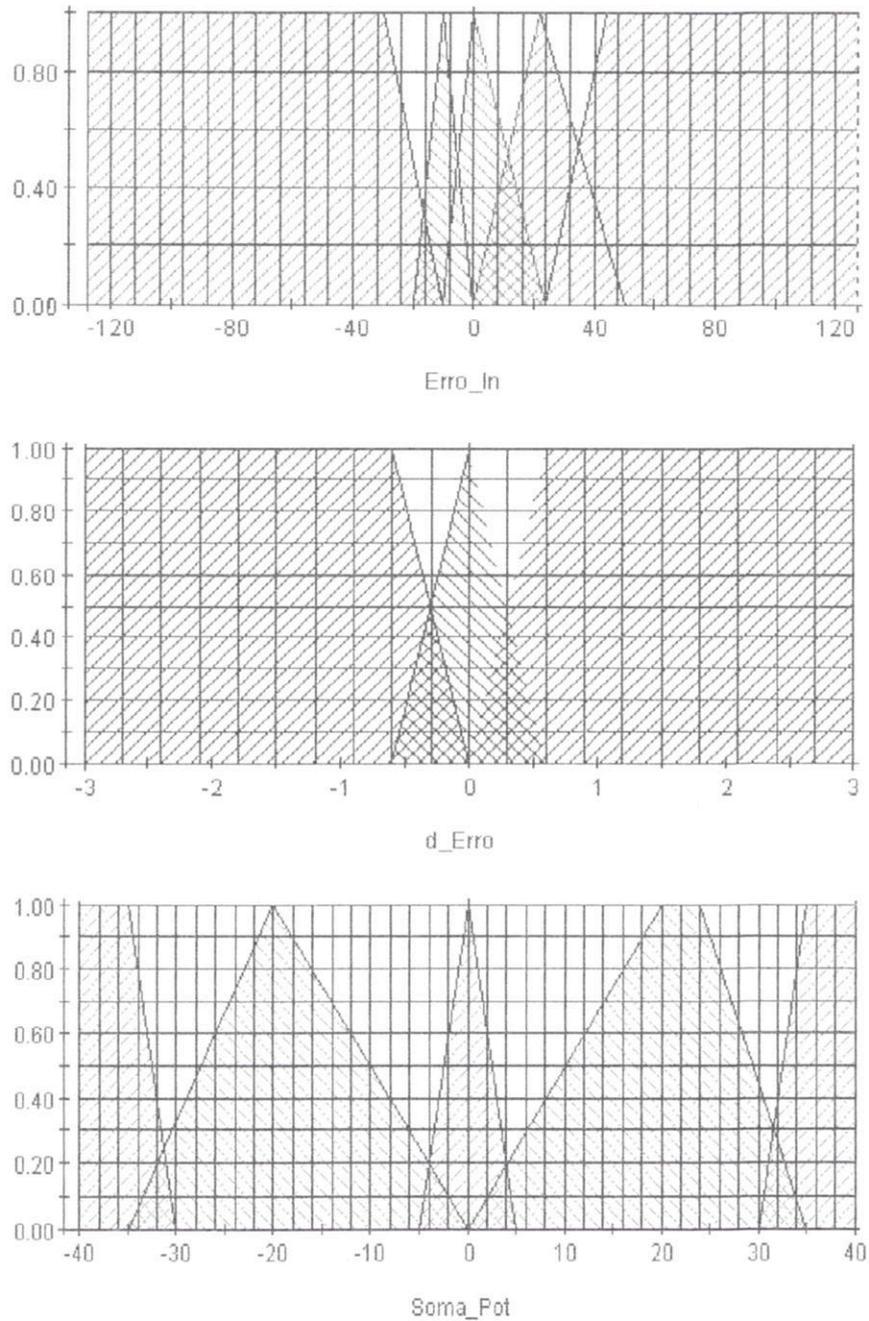


Fig. 4.1 - Funções de pertinência da configuração final do controlador difuso.

O programa FPL do onde é mostrada a configuração final do controlador (variáveis com suas funções de pertinência, base de regras e conexões entre eles) é mostrado no Anexo B.

4.2.2 Respostas Dinâmicas:

As modificações nas funções-membro da variável Erro_In resultaram em uma reformulação do problema e conseqüentemente em mudanças na tabela de controle.

Os ensaios foram realizados no laboratório, e os resultados obtidos a cada tentativa eram usados para otimização do controlador. Após alguns ajustes, encontrou-se as seguintes respostas (Fig. 4.2):

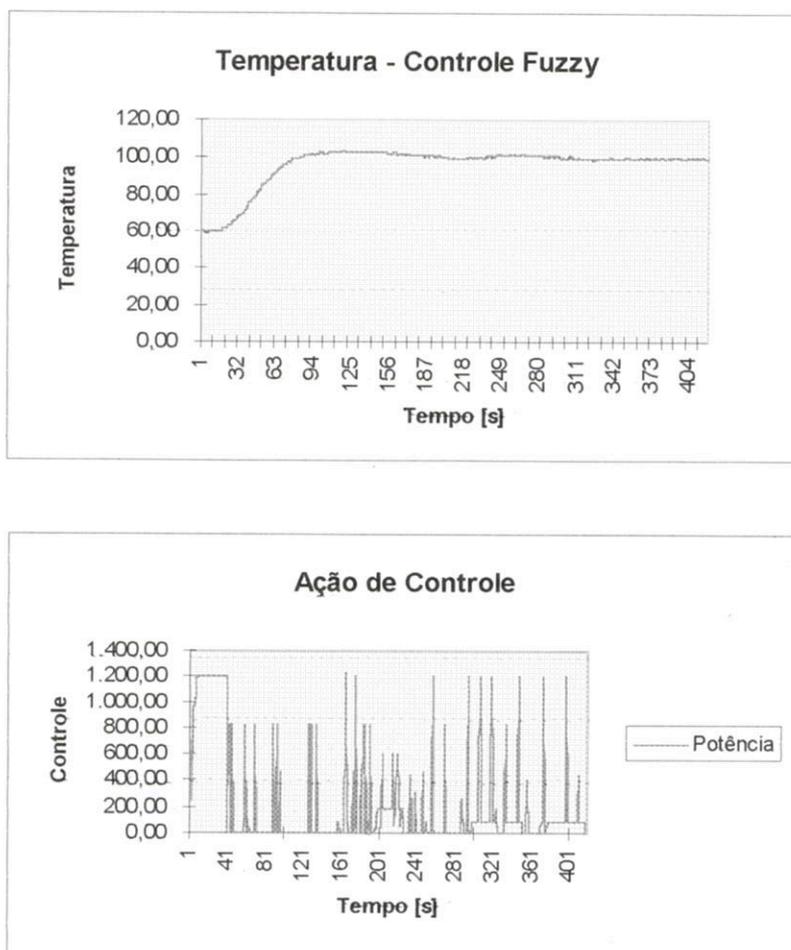


Fig. 4.2 - Respostas dinâmicas obtidas na configuração final do controlador.

Observa-se que o controlador é levado a aumentar a ação de controle (largura de faixa) até um valor de saturação e que esta é diminuída rapidamente até zero quando a temperatura atinge valores próximos da referência. Acontece que a partir deste ponto o forno já possui uma quantidade de calor suficiente para elevar a temperatura até o valor desejado (inércia térmica).

A resposta da temperatura do sistema pode ser considerada razoável, visto que a referência foi seguida, a não ser por variações em regime permanente. Estas oscilações podem ter várias causas:

a) Considerando que a ação de controle é modulada em largura de pulsos (PWM), isto por si só já produz uma frequência que pode estar influenciando no sistema.

b) A instabilidade da ação de controle em regime permanente também leva a oscilações na resposta do sistema, que só não são maiores porque a frequência destas variações é alta e a planta, por responder lentamente a estes estímulos, acaba por não ser muito afetada por eles.

c) Os valores lidos pela placa na entrada analógica (temperatura), passam por um conversor A/D e então ficam armazenados em um registro na memória da base de conhecimento. Através de um programa escrito em "C" para ler estes dados e mostrar na tela do computador, verificou-se que alguns dados fugiam claramente da curva sem que a tensão na entrada da placa tivesse tais variações. Estes valores gerados por ruídos internos da placa são levados em conta pelo controlador e a partir deles são tomadas decisões, o que pode atrapalhar o bom funcionamento do sistema.

4.2.3 Controlador PID X Controlador Fuzzy

A comparação entre os dois controladores pode ser feita por meio da observação das respostas de temperatura no tempo. Isto deve ser feito levando em consideração que o hardware de aquisição de dados e controle usado em cada caso não é o mesmo. Por exemplo, o número de bits de resolução do conversor A/D da placa fuzzy é 8, no caso do CLP (que implementa um controlador PID interno) é

10. Isto significa uma maior sensibilidade do controlador PID em relação ao controlador fuzzy.

A resposta usando controlador PID também possui uma pequena oscilação em regime permanente, isto devido também a modulação PWM da potência do forno (Fig. 4.3).

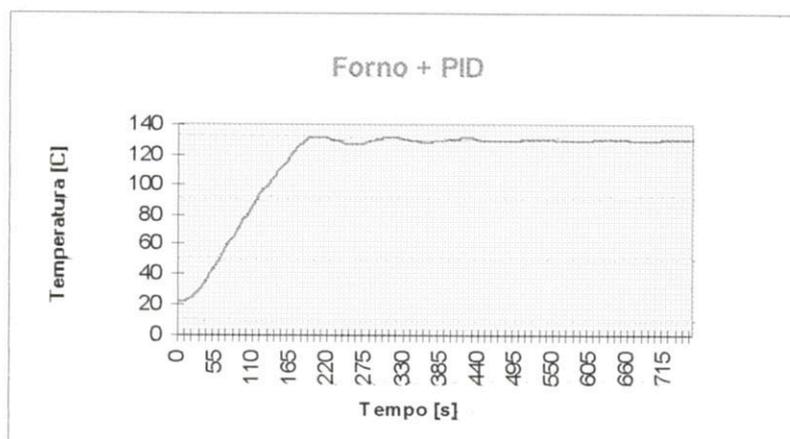


Fig. 4.3 - Resposta Dinâmica da Temperatura no Tempo para a configuração com controlador PID

5. Conclusões e Recomendações

5.1 Conclusões do Trabalho

Este trabalho teve como principais objetivos estudar o controle automático de temperatura e analisar as capacidades da placa RFC1-AT, fabricada pela Togai, utilizando lógica difusa.

Os resultados obtidos mostraram que este tipo de controle se mostrou razoável e se adaptou ao problema, na medida que a modelagem do sistema é complexa e não-linear. É necessário ressaltar que, visto que um controlador tradicional “heurísticamente” ajustado já resolvia o problema, não havia a necessidade de uma troca de filosofia de controle, tendo este trabalho a intenção de avaliar os resultados para fins de estudo e comparação.

A calibração do sensor de temperatura trouxe alguns problemas devido a falta de equipamentos adequados para se fazer este ajuste. Por esta razão, a temperatura lida pela placa não reflete a situação real, mas um valor maior. Assim, pode se dizer que o controlador, na verdade, está controlando o erro do valor lido em relação ao valor de referência e não, como seria correto, o erro entre a temperatura do forno e a temperatura de referência. No entanto, os resultados obtidos mostraram que a dinâmica da temperatura foi satisfatória.

Quanto a placa RFC1-AT, alguns problemas foram encontrados:

- a existência de um possível ruído interno da placa que gera leituras errôneas e prejudicam o perfeito funcionamento do sistema,
- a dificuldade em se fazer um programa em Assembly, sendo que outras placas similares permitem a programação em linguagem de alto nível (“C”),
- um outro problema encontrado residiu no fato de que os aplicativos, entregues juntamente com a placa pela Togai, não permitem a visualização do estado das entradas da placa e ação de controle em tempo real, ou seja, é preciso parar a execução do “loop” de controle para que isto seja possível. Por esta razão,

foi necessária a implementação de um outro programa que pudesse ler os mesmos dados e mostrar na tela do monitor continuamente.

Em relação ao software da rotina de controle, pode-se dizer que é de fácil entendimento, e possui as características desejadas para um software deste tipo como simulação e teste do projeto de controle.

5.2 Recomendações para Trabalhos Futuros

A dinâmica da temperatura do túnel de aquecimento é bastante lenta e não precisa de muitas regras para controle da mesma. Conclui-se que não havia a necessidade de usar uma placa com microprocessador dedicado para tal aplicação. A placa RFC1-AT foi utilizada, no caso, para efeitos de estudo e teste.

Seria interessante o uso desta placa em uma aplicação onde o tempo de processamento das inferências fuzzy determinasse o bom funcionamento do sistema e as características de velocidade de processamento fossem também experimentadas. A complexidade do processamento destes cálculos depende de vários fatores anteriormente citados (2.2.5).

De modo geral, os controladores fuzzy tiveram uma primeira fase de estudos onde o objetivo principal era a justificativa e comparação com outras técnicas de controle. Atualmente, os estudos voltaram-se para métodos de auto-ajuste da base de conhecimento e sobre a estabilidade dos controladores fuzzy. Estes estudos são de grande importância e têm sido muito reconhecidos atualmente.

6. Referências Bibliográficas

[Arbex 94]

R. T. Arbex, "Controle Fuzzy: Conceitos e Aplicações", INSTEC, junho 1994.

[Babaci 96]

S. Babaci, B. Lakehal, "Fuzzy Adaptative Method to Design Fuzzy Controller", IEEE International Conference on Fuzzy Systems, New Orleans, Setembro 1996.

[Barczak 93]

C. L. Barczak et al. , "A Fuzzy Controller for N.C. Machine Tool Positioning System.", IV CIMIC, Antofagasta - Chile, setembro 1993.

[Costa 95]

A. Costa et al., "Hardware Solutions for Fuzzy Control", Proceedings of the IEEE, Vol. 83, N^o 3, pp. 422-433, 1995.

[Franklin 81]

G. F. Franklin, J. D. Powell, "Digital Control of Dinamic Systems", Addison-Wesley Publishing Company, 1981.

[Gomide 94]

F. A. C. Gomide e R. R. Gudwin, "Modelagem, Controle, Sistemas e Lógica Fuzzy", Unicamp - FEE, Campinas - SP. SBA Controle & Automação, Vol. 4, n^o 3, pp. 1-28, Set/Out 1994.

[Homaifar 95]

A. Homaifair, E. McCormick, "Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algoritms", IEEE Transactions on Fuzzy Systems, Vol. 3, N^o 2, May 1995.

[Imaiida 90]

T. Imaiida et al., "Development of Fuzzy Logic Control System for Heat Pump Air Conditioners", Mitsubishi Heavy Ind. Tech. Review, Vol. 27, N^o 3, 1990.

[Incropera 92]

F. P. Incropera, D. P. de Witt, "Fundamentos de Transferência de Calor e de Massa", LTC, 3^a edição, 1992.

[Ishibuchi 95]

H. Ishibuchi, K. Yamamoto, H. Tanaka, "Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algoritms", IEEE Transactions on Fuzzy Systems, Vol. 3, N^o3, August 1995.

[Lee 90]

C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I ", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, N^o 2, March/April 1990.

[Li 89]

Y. F. Li, C. C. Lau, "Development of Fuzzy Algorithms for Servo Systems", IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988.

[Luz jr. 96]

E. B. Luz Jr., "Sistema CLP/Microcomputador para Controle e Monitoração de um Sistema de Termoformagem", VII Congresso Nacional de Ingenieria Mecánica, Valdivia - Chile, 1996.

[Mandani 74]

E. H. Mandani, "Application of Fuzzy Algorithms for Servo Systems", Proc. IEEE, Vol. 121, pp. 1585-1588, 1974.

[Normey 94]

J. E. Normey Rico, "Sinais e Sistemas Lineares", Apostila do curso de Sinais e Sistemas Lineares-ECA, 1994.

[Ogata 93]

K. Ogata, "Engenharia de Controle Moderno", Editora Prentice Hall do Brasil Ltda., 1993.

[Rosa 90a]

L. C. Rosa et al., "Is a Fuzzy Controller Applicable to Machine Tool?", Proceedings of IEEE Symposium on Intelligent Motion Control, istambul, 1990.

[Rosa 90b]

L. C. Rosa et al. , "Controle de Servo Motor Utilizando Algoritmos Difusos", 8^o Congresso Brasileiro de Automática, Belém, 1990.

[Rosa 91]

L. C. Rosa, E. A. Gutierrez, "Aplicabilidade do Controlador Difuso no Posicionamento de Máquina Ferramenta", XI COBEM, 1991.

[Self 90]

K. Self, "Designing with Fuzzy Logic", IEEE Spectrum, 42-44, November 1990.

[Silva 95]

D. D. e Silva, "Um Mecanismo Seguidor Baseado em um Controlador Difuso Dedicado", Dissertação de Mestrado, PPGCC, UFSC, Agosto de 1995.

[Simas 95]

H. Simas, " Implementação de Controladores Difusos Aplicados a Sistemas de Potência ", Monografia de Final de Curso - ECA", 1995.

[Togai 91]

Togai Infralogic, Inc, "FC110 Development System User's Manual", Version 2.3.0, Last Revised: September 10, 1991.

[Togai 92a]

Togai Infralogic, Inc. , "TILShell+ User's Manual", Versão 2.0.0, Last Revision: 1992.

[Togai 92b]

Togai Infralogic, Inc. , "Fuzzy-C Development System User's Manual", Versão 2.4.0, Last Revision: December, 1992.

[Togai 93]

Togai Infralogic, Inc, "Technical Documentation - Reselec Fuzzy Controller AT-Board", Version 1.0, January 1993.

[Wang 92]

L. Wang, "Generating Fuzzy Rules by Learning from Examples", IEEE Transactions on Systems, Man and Cybernetics, Vol. 22, N^o 6, November/December 1992.

[Wellstead]

"Fuzzy Logic Training", Artigo retirado do endereço internet "<http://www.tecquip.co.uk/products/ce124fuz.htm>".

[Zadeh65]

L. A. Zadeh, "Fuzzy Sets". Control, n^o 3, pp. 338-354, 1965.

[Zadeh73]

L. A. Zadeh, " ".

Anexo A - Programa em Assembly para Aquisição de Dados

```

;*****
;
;
;   Programa de controle do forno
;   Priscila P. de P. e Souza
;   Data:31/07/96
;
;*****
;*****
;*****
;
;   .mod    forno
;
;   .ext    Controle_de_Temperatura      ; Nome do Project - TILShell
;   .ext    Erro_In                      ; Variáveis externas
;   .ext    d_Erro
;   .ext    Soma_Pot
;
;=====
; Definição de Constantes
;=====
;
SAMPLE_INT .equ    20                ; Tempo de Amostragem 2 s
TIME_BASE  .equ    100
SET_POINT  .equ    40h               ; Setpoint de Temperatura=59.6 C
;
;-----
;   .incl   "rfcl.inc"                ; Inclui alguns arquivos
;   .incl   "break.inc"
;-----
;
;=====
; Definição de variáveis internas
;=====
;
;   .sect   data sram
;
Erro:      .ds.b    1
ErroAnt:   .ds.b    1
PotOut:    .ds.b    1
Loop_count: .ds.b    1
Sinal:     .ds.b    1
;
;=====
; Code section
;=====
;
;   .sect   code kb
;

```

```

                .pub    Main_start                ;
                ;
                ;-----;
; Bloco Principal
                ;-----;
                ;
                ;-----;
; Inicialização das variáveis
                ;-----;
Main_start:    load    #0, r1                    ;
                store   r1, D(DAC_DATA3)        ; Saída analógica
                store   r1, D(Erro)             ;
                store   r1, D(ErroAnt)          ;
                store   r1, D(PotOut)           ;
                store   r1, D(Sinal)            ;
                load    #0, r3.w                ; contador loop de execução
                ;
                ;-----;
; Inicializa Timer
                ;-----;
                ;
                load    #TIME_BASE, r1          ;
                store   r1, D(MUART_TIMER1)     ;
                load    #SAMPLE_INT, r1        ;
                store   r1, D(Loop_count)      ;
                ;
                ;-----;
; Loop principal começa aqui
                ;
; Espera pelo fim do tempo de amostragem
                ;-----;
mainloop:     load    D(MUART_TIMER1), r1      ;
                cmp     #0, r1                  ;
                jmp     !z, mainloop            ;
                ;
                load    D(Loop_count), r1      ;
                sub     #1, r1                  ; decrementa o contador de loop
                store   r1, D(Loop_count)      ;
                jmp     z, Go_a_loop            ; Salta Go_a_loop
                ;
                load    #TIME_BASE, r1         ; re-inicializa o timer
                store   r1, D(MUART_TIMER1)   ;
                jmp     always, mainloop       ;
                ;
                ;-----;
; Re-inicializa o timer com o tempo de amostragem
                ;-----;
                ;
Go_a_loop:    load    #TIME_BASE, r1          ;
                store   r1, D(MUART_TIMER1)   ;
                load    #SAMPLE_INT, r1       ;
                store   r1, D(Loop_count)     ;
                ;
;***** F U Z Z Y   L O O P *****;

```

```

load    D(ADC_DATA2), r1
load    #SET_POINT, r2
load    #58h, r1
sub     r1, r2
store   r2, D(Erro)
store   r2, D(Erro_In)      ; passa "Temp_In" para TILShell
load    D(ErroAnt), r1
sub     r2, r1
load    #0, r1
store   r1, D(d_Erro)      ; passa "d_Temp" para TILShell

jsr     always, r0.w, Controle_de_Temperatura ;rotina de controle (TILShell)

load    #0, r4.w
load    #0, r1.w
load    #0, r8.w

load    D(Soma_Pot), r4      ; lê o valor da soma de potência (saída de controle)
load    #080h, r5
cmp     r4, r5
jsr     c, r0.w, negativo

load    D(PotOut), r1
load    #0ffh, r1
add     r1.w, r4.w          ; Soma com a saída de controle

load    #8000h, r2.w
cmp     r4.w, r2.w
jmp     c, satura_0

load    #0ffh, r8.w
cmp     r4.w, r8.w
jmp     c, satura_ff

load    #0, r7
jmp     always, continua

negativo: load    #0ff00h, r1.w
          add     r1.w, r4.w
          rts     r0.w

satura_0: load    #0, r4.w
          jmp     always, continua

satura_ff: load    #00ffh, r4.w
          jmp     always, continua

continua: store   r4, D(DAC_DATA3) ; escreve a potência na saída DAC canal 0
          store   r4, D(PotOut)   ; guarda o valor da potência para usar no próximo ciclo
          load    D(Erro), r1      ;
          store   r1, D(ErroAnt)  ;

```


Anexo B - Programa para Controle em Linguagem FPL - Fuzzy Programming Language

PROJECT Controle_de_Temperatura

```
VAR Erro_In
  TYPE signed byte

  MEMBER Grande_negativo
    POINTS -128,1 -30,1 -10,0
  END

  MEMBER Zero
    POINTS -10,0 0,1 24,0
  END

  MEMBER Grande_Positivo
    POINTS 24,0 44,1 127,1
  END

  MEMBER Negativo
    POINTS -20,0 -10,1 0,0
  END

  MEMBER Positivo
    POINTS 0,0 22,1 50,0
  END
END
```

```
VAR d_Erro
  TYPE signed byte
  MIN -3
  MAX 3

  MEMBER negativo
    POINTS -3,1 -0.6,1 0,0
  END

  MEMBER zero
    POINTS -0.6,0 0,1 0.6,0
  END

  MEMBER positivo
    POINTS 0,0 0.6,1 3,1
  END
END
```

```
VAR Soma_Pot
  TYPE signed byte
  MIN -40
  MAX 40

  MEMBER Tira_muita
    POINTS -40,1 -35,1 -30,0
```

```
END

MEMBER Tira_pouca
  POINTS -35,0 -20,1 0,0
END

MEMBER zero
  POINTS -5,0 0,1 5,0
END

MEMBER Soma_pouca
  POINTS 0,0 20,1 24,1 35,0
END

MEMBER Soma_muita
  POINTS 30,0 35,1 40,1
END
END

FUZZY Controle_Fuzzy

RULE Rule_1
  IF (Erro_In IS Grande_Positivo) AND (d_Erro IS positivo) THEN
    Soma_Pot=Soma_pouca
  END

RULE Rule_2
  IF (Erro_In IS Grande_Positivo) AND (d_Erro IS zero) THEN
    Soma_Pot=Soma_muita
  END

RULE Rule0008
  IF (Erro_In IS Grande_Positivo) AND (d_Erro IS negativo) THEN
    Soma_Pot=Soma_muita
  END

RULE Rule0011
  IF (Erro_In IS Positivo) AND (d_Erro IS negativo) THEN
    Soma_Pot=Soma_pouca
  END

RULE Rule0012
  IF (Erro_In IS Positivo) AND (d_Erro IS zero) THEN
    Soma_Pot=Soma_pouca
  END

RULE Rule0013
  IF (Erro_In IS Positivo) AND (d_Erro IS positivo) THEN
    Soma_Pot=Tira_muita
  END

RULE Rule_3
  IF (Erro_In IS Zero) AND (d_Erro IS positivo) THEN
    Soma_Pot=Tira_pouca
  END

RULE Rule_4
```

```
    IF (Erro_In IS Zero) AND (d_Erro IS zero) THEN
      Soma_Pot=zero
    END

    RULE Rule_5
      IF (Erro_In IS Zero) AND (d_Erro IS negativo) THEN
        Soma_Pot=Soma_pouca
      END

    RULE Rule0014
      IF (Erro_In IS Negativo) AND (d_Erro IS negativo) THEN
        Soma_Pot=Soma_pouca
      END

    RULE Rule0015
      IF (Erro_In IS Negativo) AND (d_Erro IS zero) THEN
        Soma_Pot=Tira_pouca
      END

    RULE Rule0016
      IF (Erro_In IS Negativo) AND (d_Erro IS positivo) THEN
        Soma_Pot=Tira_muita
      END

    RULE Rule0007
      IF (Erro_In IS Grande_negativo) AND (d_Erro IS positivo) THEN
        Soma_Pot=Tira_muita
      END

    RULE Rule_6
      IF (Erro_In IS Grande_negativo) AND (d_Erro IS zero) THEN
        Soma_Pot=Tira_muita
      END

    RULE Rule_7
      IF (Erro_In IS Grande_negativo) AND (d_Erro IS negativo) THEN
        Soma_Pot=Tira_muita
      END
    END

    CONNECT
      FROM Erro_In
      TO Controle_Fuzzy
    END

    CONNECT
      FROM d_Erro
      TO Controle_Fuzzy
    END

    CONNECT
      FROM Controle_Fuzzy
      TO Soma_Pot
    END
  END
```

Anexo C - Programa em Linguagem "C" para Visualização de Resultados

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <ctype.h>

void main()
{
    FILE *f1;
    FILE *f2;
    int i;
    int valor1, valor2;
    float potencia;
    float teste;

    valor1 = 0;
    valor2 = 0;
    i = 0;

    /* Limpa a Tela */
    clrscr();

    /* Abre arquivos para escrita */
    f1 = fopen("arq_temp.dat","wt");
    f2 = fopen("arq_pot.dat","wt");

    /* Escreve cabeçalho */
    fprintf(f1,"Arquivo de Temperatura\n\n");
    fprintf(f2,"Arquivo de Potencia\n\n");

    /* Faz Leitura da Memória, escreve nso arquivos, imprime na tela */

    do
    {
        valor1 = peekb(0xd600,0x922);
        valor1 = (valor1 & 255); /* Le temperatura da memoria */
        /* printf("\n valor temp em hexa : %x", valor1); */
        teste = (float)valor1*200/255;
        valor2 = peekb(0xd600,0x943); /* Le potencia da memoria */
        valor2 = (valor2 & 255);
        printf("\n valor potencia em hexa : %x", valor2);
        potencia = 1200 * (float)valor2;
        potencia = potencia/255;
        printf("\nTempo: %d s", i);
        printf("\nTemperatura: %f",teste); /* Escreve os dados na tela */
        printf("\nPotencia: %f", potencia);
        delay(1000);
        clrscr();
        i++;
        fprintf(f1,"%f\n",teste); /* escreve os dados no arquivo de Temp. */
    }
```

```
    fprintf(f2, "%f\n", potencia); /* escreve os dados no arquivo de potencia */  
} while (i < 500);  
  
fclose(f1);  
fclose(f2);  
}
```