

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM MECANISMO SEGUIDOR
BASEADO EM UM CONTROLADOR DIFUSO DEDICADO**

DISSERTAÇÃO SUBMETIDA À
UNIVERSIDADE FEDERAL DE SANTA CATARINA PARA A OBTENÇÃO DO
GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO

DAVID DANIEL E SILVA

FLORIANÓPOLIS, AGOSTO DE 1995

**UM MECANISMO SEGUIDOR
BASEADO EM UM CONTROLADOR DIFUSO DEDICADO**

DAVID DANIEL E SILVA

ESTA DISSERTAÇÃO FOI JULGADA PARA OBTENÇÃO DO TÍTULO DE

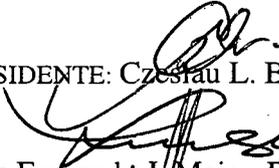
MESTRE EM CIÊNCIA DA COMPUTAÇÃO

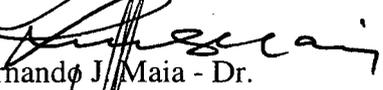
ESPECIALIDADE SISTEMAS DE COMPUTAÇÃO E APROVADA EM SUA FORMA
FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO


ORIENTADOR: Czesław L. Barczak

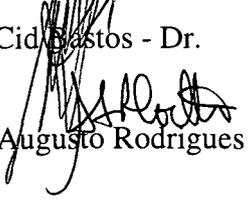

COORDENADOR DO CURSO: Rogério C. Bastos

BANCA EXAMINADORA


PRESIDENTE: Czesław L. Barczak - Dr.


Luiz Fernando J. Maia - Dr.


Rogério Cid Bastos - Dr.


Antônio Augusto Rodrigues Coelho - Dr.

Aos meus familiares:

Sempre há a hora do despertar. Independente da hora vivida, sempre é possível conseguir mais um pouquinho de luz. Eu adoro todos vocês e tenho certeza que o amor e a fraternidade está em cada coração. Muito obrigado.

AGRADECIMENTOS

Obrigado a todos aqueles que me ajudaram de forma direta ou indireta na construção deste trabalho.

Ao **Professor Hermann Lücke** desejo que a luz divina esteja sempre lhe guiando, seja lá a onde estiver.

Um agradecimento especial ao Carlos Montez, ao Eduardo Kern e ao Rodrigo Cabral pelos momentos dispensados a este trabalho. E, obrigado a todos aqueles que pelo menos nos fazem abrir um sorriso e sair do sério.

Um grande abraço na alegre Verinha, que com o seu jeito meigo e competente sempre consiga unir a alegria e os alunos.

Para a **Professora Clara**, muito obrigado pela confiança no empréstimo de alguns livros.

Ao **Professor Czeslau**, meu orientador, obrigado pelas horas dedicadas a mim, para que o trabalho pudesse ser feito adequadamente.

Obrigado a toda banca examinadora (**Prof. Rogério C. Bastos, Prof. Antônio A. Coelho, Prof. Luiz F. J. Maia**) pelas sugestões importantes que me foram passadas.

Para todos...

...“Era de manhã e o novo sol cintilava nas rugas de um mar calmo.

A dois quilômetros da costa, um barco acariciava a água. Subtamente, os gritos do Bando da Alimentação relampejaram no ar e despertaram um bando de mil gaivotas, que se lançou precipitadamente na luta pelos pedacinhos de comida. Amanhecia um novo dia de trabalho.

Mas lá ao fundo, sozinho, longe do barco e da costa, Fernão Capelo Gaivota treinava” tentando, através da velocidade, chegar a perfeição. [Trecho de **Richard Bach** do livro “A História de Fernão Capelo Gaivota”.]

Aos amigos, Obrigado! E que a busca da perfeição continue onde quer que vocês estejam.

Ao CPGCC, ao INE, a UFSC e a CAPES, muito obrigado .

SUMÁRIO

Pág.

Lista de Figuras.....	viii
Lista de Tabelasix
Resumo.....	...x
Abstractxi

Capítulo 1

INTRODUÇÃO

1.1 - Abordagem Geral	01
1.2 - A Sistemática do Trabalho	02
1.3 - Análise Bibliográfica e Sobre o Trabalho Proposto	03

Capítulo 2

A LÓGICA DIFUSA

2.1 - Introdução.....	05
2.2 - Conceitos da Lógica Difusa	05
2.3 - Variáveis e Graus de Pertinência	06
2.4 - Função de Pertinência	08
2.5 - Conjuntos Difusos.....	08
2.6 - Universo de Discurso	09

Capítulo 3

CONTROLE DIFUSO

3.1 - Introdução.....	10
3.2 - Fuzificação	11
3.3 - Processo de Inferência.....	12
3.3.1 - As Regras Difusas e Seus Operadores.....	12
3.3.2 - Agregação.....	13
3.3.3 - Composição.....	14
3.4 - Procedimentos Para Determinação da Saída.....	15
3.4.1 - Defuzificação	16
3.4.1.1 - Método do Centróide	17
3.4.1.2 - Método das Barras Verticais.....	18
3.4.1.3 - Método da Altura	19

Capítulo 4

O MECANISMO SEGUIDOR DE SUPERFÍCIES

4.1 - Introdução.....	22
4.2 - A Arquitetura do Seguidor Difuso.....	23
4.2.1 - O Acionamento da Mesa X-Y-Z.....	24
4.2.2 - A Placa mc-FUZZYLAB.....	25
4.3 - Constituição Física do Sistema do Seguidor Difuso.....	26
4.4 - O Controle por Lógica Difusa.....	30
4.4.1 - Determinação dos Limites das Funções de Pertinência.....	30
4.4.2 - Definição de Variáveis e Parâmetros.....	31
4.4.3 - O Controle com Três Regras Difusas.....	32
4.4.4 - O Controle com Três Regras com Limites Médio dos Conjuntos não Coincidentes.....	36
4.4.5 - O Controle com Cinco Regras Difusas.....	40
4.5 - Determinação da Região de Instabilidade.....	47
4.6 - Avaliação dos Experimentos.....	48
4.7 - Os Objetos Utilizados e as Curvas Reais.....	49
4.8 - Conclusão.....	51

Capítulo 5

CONCLUSÕES E RECOMENDAÇÕES

5.1 - Conclusões do Trabalho.....	52
5.2 - Recomendações.....	53

REFERÊNCIAS BIBLIOGRÁFICAS.....	55
--	-----------

APÊNDICE A

O Processador Difuso FC110.....	64
A.1 - Introdução.....	64
A.2 - O Processador FC110.....	64
A.2.1 - A Interface de memória de Base de Conhecimento.....	68
A.2.2 - A Interface Para Sistemas Baseados no FC110.....	69
A.2.3 - A RAM de Dados Partilhados Interna.....	69
A.2.4 - A Interface Periférica.....	70
A.2.5 - Resumo do Conjunto de Instruções do FC110.....	72
A.2.6 - Modos de Endereçamento do FC110.....	73
A.2.7 - Metodologia de Transferência de Dados do Sistema.....	74

APÊNDICE B

A Placa mc-Fuzzylab	77
B1 - Introdução	77
B2 - O “Hardware”	77
B3 - O “Software”	78

APÊNDICE C

Procedimento Para o Controle do Mecanismo Seguidor.....	80
C1 - Introdução	80
C2 - O SEGUIDOR_FC110.C.....	80
C3 - O Arquivo SEGUE110.C	81

Lista de Figuras

2.1 - Conjuntos difusos para a variável - TEMPERATURA	07
3.1 - Fluxo de dados de um sistema difuso	10
3.2 - Conjuntos difusos para a variável de saída potência_de_motor (PM)	15
3.3 - Resultado da União dos Conjuntos Difusos de Saída.....	16
3.4 - Resultado da Defuzificação.....	18
3.5 - Defuzificação pelas barras verticais.....	19
4.1 - Arquitetura do Sistema de Controle.....	23
4.2 - As conexões Físicas.....	24
4.3 - Representação da Montagem.....	27
4.4 - Fotografia do Dispositivo.....	28
4.5 - Relação da Voltagem no Potenciômetro	29
4.6 - Diagrama de Blocos	29
4.7 - Faixa de Valores Aceitáveis.....	30
4.8 - Característica do Contorno no plano.....	35
4.9 - Característica do Contorno do Objeto Curvo.....	35
4.10 - Característica do Plano com Três Regras com Limites variados.....	38
4.11 - Características do Contorno Curvo Efetuado com Três Regras	39
4.12 - Característica do Contorno no Plano	43
4.13 - Característica do Objeto Curvo	44
4.14a - Gráfico da Relação Pos_x e Nível.....	47
4.14b - Gráfico da Região da Instabilidade	48
4.15 - O Plotter.....	49
4.16 - Superfícies Utilizadas.....	49
4.17 - Função real do Contorno	50
4.18 - Função Característica do Contorno do Objeto Curvo.....	50
A.1 - Arquitetura básica com o FC110.....	66
A.2 - Microcontrolador baseado no FC110.....	67
A.3 - Arquitetura de microcomputador baseado no FC110	67
A.4 - A arquitetura de multiprocessamento com o FC110.....	68
A.5 - Diagrama de blocos do FC110	69
A.6 - Mapeamento de endereços visto do sistema e do FC110	71
B1 - A Placa mc- Fuzzylab	78

Lista de Tabelas

3.1 - Aplicação da operação E na lógica booleana.....	13
3.2 - Operações elementares da lógica difusa	14
4.1 - Tempo e erro no Contorno	36
4.2 - Tempo e erro no Contorno	39
4.3 - Quadro de Controle para o Caso de Três Regras.....	39
4.4 - Tempo e o Erro para o Uso de Cinco Regras.....	44
4.5 - As cinco regras de Controle	45
4.6 - Dados de Processamento	46
A.1 - Resumo das Instruções do FC110.....	72

RESUMO

Este trabalho descreve um mecanismo seguidor de superfícies com a utilização de um sensor eletro-mecânico e uma placa, “*mc-FUZZYLAB*” [Klein-91], que faz uso do processador FC110, específico para processar lógica difusa [Togai-91].

A placa está acoplada a um equipamento anfitrião, PC-AT, e o processador “fuzzy” (FC110) controlará uma máquina que manterá o dispositivo seguidor em uma distância constante ao contornar objetos.

Os dados de entrada para o controle do contorno do traçado de superfícies são fornecidos pelo dispositivo sensor eletro-mecânico. O controle é feito em duas dimensões, ou seja, no plano.

ABSTRACT

This work describes a surface follower with the use of a electromechanic sensor and a board, "*mc-FUZZYLAB*" [Klein-91], that makes use of the FC110 processor, specific for fuzzy logic processing [Togai-91].

The board is mounted to a host equipment, PC-AT, and the fuzzy processor (FC110) will control a machine that will keep the following device at a constant distance while contouring objects.

The input data for the surface trace contour control are provided by the electromechanic sensor device. The control is done in two dimensions, that is to day, in the plane.

CAPÍTULO 1

INTRODUÇÃO

1.1 - Abordagem Geral

A área de controle automático de processos vem se desenvolvendo, aceleradamente, ano após ano. Uma grande contribuição para o desenvolvimento da área de controle de processos é proveniente da informática [Klein91], [Jamshidi93], [Filho93]. Os computadores, desenvolvidos hoje, ganham cada vez mais capacidade de processamento e, dispositivos baseados em processadores dedicados são fabricados para melhorar ainda mais a eficiência da ação de processos [Tog91].

É importante ressaltar, que inúmeras técnicas e ferramentas originadas do desenvolvimento da inteligência artificial estão sendo utilizadas em controle inteligente nas mais variadas aplicações [Tog90], [Viot93], [Runkler94], [López94], [Jamshidi93], [Mamdani74], [Chang94]. Em particular, as técnicas baseadas na lógica difusa (“fuzzy logic”) estão contribuindo para o desenvolvimento de muitos projetos [Jamshidi93].

Mais recentemente, surgiram no mercado processadores como o FC110 da Empresa Togai Infralogic [Tog91] que são dedicados no processamento da lógica difusa. (Características do processador FC110 estão resumidamente descritas no apêndice A).

Neste trabalho, realiza-se a análise de questões que envolvem técnicas de controle, baseadas na lógica difusa e com o uso do processador dedicado FC110 da Togai Infralogic .

O processador FC110 e alguns circuitos periféricos auxiliares e de interface para o computador PC/AT , são utilizados na elaboração e montagem de uma placa, denominada **mc-Fuzzylab** [Klein91] (apêndice B). Esta placa é parte do conjunto de elementos deste trabalho, possibilitando fazer testes práticos em laboratório.

1.2 - A Sistemática do Trabalho

Um primeiro aspecto a ser abordado neste trabalho, são alguns conceitos resumidos da lógica difusa (Capítulo 2) e, com maiores detalhes, do controle difuso (Capítulo 3).

Pretende-se, com esta pesquisa, utilizar as técnicas de controle difuso, fazendo-se testes, observando-se simulações, comparações e análises de aplicações com os recursos disponíveis, para que se tenha um produto final, em controle, que faça uso da lógica difusa. Dentro desta análise desenvolve-se um *Sistema de Controle Difuso de um Mecanismo Seguidor de Superfícies* e implementa-se fisicamente (Capítulo 4).

O Seguidor Difuso é um mecanismo constituído de uma barra metálica que acompanha a superfície de objetos, como por exemplo, placas planas, cilindros, etc. O percurso feito pelo seguidor é controlado por motores elétricos de passo, acionados com sinais provenientes de controle difuso, de modo a deixar a barra seguidora o mais próximo possível da peça (objeto a ser seguido).

O objetivo é a implementação do mecanismo seguidor de superfície utilizando controle difuso e ferramentas que vem sendo cada vez mais aplicadas na prática: a lógica difusa, “softwares” aplicativos, “hardware” disponível, etc. Para esse estudo, baseia-se em referências como [Klein91], [Li88], [Gomide92], [Kaufmann75], [Palm93], [Tog91], [Zimmermann93], [Zimmermann85], [Kandel86], [Jamshidi93], [Zadeh92]. Dessa forma, pode-se observar como aplicar a lógica difusa ao controle.

A instalação física para o dispositivo seguidor apresenta um computador, PC/AT tipo 386/DX 40MHz, no qual é instalada a placa de circuito, mc-Fuzzylab, baseada no controlador difuso - FC110 - e “softwares” necessários para a execução do projeto e que incluem, também, demonstrações.

Uma plataforma, X-Y, para confecção de placas de circuitos impressos, acionada por motores elétricos de passo, constitui o elemento eletromecânico para o deslocamento do seguidor. Os dados para controle, são fornecidos por sensor instalado na plataforma X-Y, de modo a informar a posição atual do seguidor.

Em resumo o controle baseado na lógica difusa, com o FC110, é aplicado a um dispositivo para regular, de forma precisa, a distância no contorno do traçado em superfícies.

O sistema desenvolvido é mostrado no capítulo 4, onde apresenta-se a placa mc-Fuzzylab, que é constituída basicamente pelo processador FC110 e sua base de conhecimento. Também é parte integrante do sistema, o processador anfitrião, os sensores e atuadores que auxiliarão na execução do controle do processo. Ao final, faz-se sugestões de aplicações baseadas no mecanismo seguidor e apresenta-se os resultados obtidos.

1.3 - Análise Bibliográfica e Sobre o Trabalho Proposto

Entre as referências da literatura que auxiliam os trabalhos da área de controle, adquiriu-se informações no manual da Togai [**Tog91**] que descreve, o funcionamento do processador FC110, como compilar programas com o uso desse processador, a linguagem de baixo nível do processador FC110 e o funcionamento da lógica difusa com o uso desse processador.

No manual da placa mc-FUZZYLAB [**Klein91**], é descrito o funcionamento da placa em relação ao “hardware” e “software” para projetos e também são incluídos alguns exemplos de programas que podem usar tanto a placa com o FC110 solitário, como em conjunto com um processador anfitrião.

Em [**Kaufmann75**], [**Dubois80**], [**Zimmermann85**], [**Kosko92**], [**Kickert78**], [**Kandel86**], [**Graham88**], há uma série de informações a respeito do tratamento matemático da lógica difusa, conjuntos difusos, suas relações, entradas e saídas de sistemas difusos, incerteza e tomada de decisão utilizando a lógica difusa.

A análise referente a sistemas de controle convencional e difuso é discutido em [**Barczak93**], que além de estudar o controle digital com modelamentos do controle convencional, dedica um capítulo a respeito do controle difuso. Em [**Li88**] é investigado a possibilidade de aplicação de algoritmos difusos baseados em microprocessador no controle de servomotor.

[Gomide92] analisa a utilização da lógica difusa no controle de processos e em [Gomide94], na modelagem de sistemas de controle difusos. Em [Han93], é proposto um novo tipo de controle para múltiplas variáveis com lógica difusa.

Em [Runkler94] aplica-se a lógica difusa no controle preciso da ré de um caminhão reboque, em [Tog90] a lógica difusa é empregada para controlar um pêndulo invertido, que se tornou uma aplicação clássica desta lógica, em [Altrock93] aplica-se o controle difuso em controle de temperatura, no controle de um guindaste e em fotocâmeras.

A aplicação da lógica difusa em fotocâmeras [Altrock93], despertou a vontade de se controlar o foco e o “zoom” de filmadoras. Contudo, após os estudos iniciais sobre o funcionamento de uma câmera de filmar, percebeu-se que não havia recursos para colocar em prática a referida idéia.

Um sistema de controle difuso de um posicionador foi desenvolvido na engenharia mecânica da UFSC [Barczak94], neste sistema o controlador difuso foi programado em assembler 8086/8088 para poder alcançar alta velocidade de computação.

Seguindo na busca de uma aplicação que tivesse boa abrangência e motivação, verificou-se que podia-se desenvolver um sistema mínimo, simples, que fosse a base para sistemas mais elaborados, ou seja, aplicar a lógica difusa para controlar um mecanismo seguidor de superfícies [Silva95]. Neste trabalho, utiliza-se um processador de lógica difusa da Togai Infralogic, FC110, e um sistema anfitrião. A linguagem C é utilizada como ferramenta de interface.

Apesar de haver inúmeras aplicações com controle difuso, não há até agora conhecimento de dispositivo prático semelhante ao que se pretende desenvolver aqui.

CAPÍTULO 2

A LÓGICA DIFUSA

2.1 - Introdução

Neste capítulo são apresentados conceitos básicos sobre a lógica difusa que são relevantes para o entendimento deste trabalho. Na sequência, procura-se voltar os assuntos abordados para a área de controle.

“Em muitos casos um operador humano tem maior sucesso em controlar um complexo processo industrial que um controlador projetado por técnicas de controle moderno” [Jamshidi93]. “O método de expressar a estratégia de um operador humano usando a teoria dos conjuntos difusos tem sido frequentemente proposta” [Kickert76].

2.2 - Conceitos da Lógica Difusa

A lógica difusa é a lógica que baseia-se na teoria dos conjuntos difusos que, por sua vez, é uma teoria baseada nos termos: vago, incerto, ambíguo, ou seja, conceitos que envolvem problemas de imprecisão ou indefinição. Nesta lógica, a resposta exata é um caso limite da resposta aproximada, sendo o processo um conjunto de relações nebulosas [Gomide94].

Na lógica difusa, o valor de uma proposição pode ser um subconjunto difuso de qualquer conjunto parcialmente ordenado, ao contrário dos sistemas lógicos binários, onde o valor de uma proposição só pode assumir dois valores: verdadeiro (1) ou falso (0) [Klein91], [Tog91], [Altrock93].

Na lógica nebulosa (difusa), os valores verdade são expressos linguisticamente, ou seja, ela tem a vantagem de se poder utilizar conceitos humanos, tais como “muito”, “perto”, “longe”, “alto”,

“quente” entre outros, com relativa facilidade. Esses conceitos são interpretados como um subconjunto difuso do intervalo unitário. [Gomide94]

A lógica difusa é uma metodologia que facilita a resolução de problemas que não tenham um modelo matemático exatamente definido da instalação a ser controlada pois ela não precisa da definição destes modelos matemáticos [Klein91]. Qual seria a dificuldade de se tentar modelar matematicamente um sistema de um processo que fosse vago, ambíguo como quase quente, frio ou frio e quente ao mesmo tempo, etc. ?

Em [Ursino88], procura-se modelar matematicamente a regulação química do fluxo de sangue no cérebro e utiliza-se o método de Runge-Kutta de resolução numérica de equações diferenciais de quarta ordem para estudo da dinâmica do sistema.

Ainda, muitos sistemas físicos são resolvidos por meio de equacionamento analítico aproximado, como a equação da distância estudada na física $d = d_0 + v_0.t + 1/2 a.t^2$. Esta equação é uma aproximação para se determinar a distância, pois não leva em consideração a resistência do ar.

A lógica difusa, além de tratar de problemas de incerteza, ao invés de trabalhar com a função de transferência de um sistema, utiliza apenas um conjunto de instruções de linguagem natural [Viot93], [Zadeh65]. Isto representa uma vantagem para a lógica difusa ser utilizada, em controle, no lugar de controladores comuns para resolver problemas de difícil modelagem [Jamshidi93].

2.3 - Variáveis e Graus de Pertinência

Uma variável é um ente matemático que pode representar uma quantidade, que pode assumir um dentre um conjunto de valores, ou ainda pode ser tida como entidade com nome usado para fazer referência aos dados e à qual podem ser atribuídos valores. Seus atributos permanecem constantes, mas pode fazer referência a valores diferentes em momentos distintos [SUCES85].

Tanto na lógica difusa como na lógica clássica, há um “ente” denominado grau de pertinência (“*degree of membership*”). Na lógica clássica o grau de pertinência só pode ser 0 (zero) ou 1 (um),

ou seja, esta lógica só nos diz se um elemento de um determinado conjunto está 100% dentro do conjunto ou se absolutamente não está.

Na lógica difusa, os graus de pertinência expressam se o elemento pertence ou não a um conjunto em uma faixa de valores que varia de 0 (zero) a 100%. Um valor nesta faixa é uma grandeza numérica **real** de zero a um, que representa a quantidade em termos de possibilidade de uma variável pertencer a um certo conjunto difuso e denominada grau de pertinência [Kaufmann75]

A figura 2.1 representa um gráfico cartesiano dos graus de pertinência em uma escala *numérica real* de zero até um (0 ; 1). Para o eixo das ordenadas tem-se os **graus de pertinência**, no eixo das abcissas representa-se, em graus Celcius, a **temperatura**. $\mu(T)$ é a função de representação dos graus de pertinência..

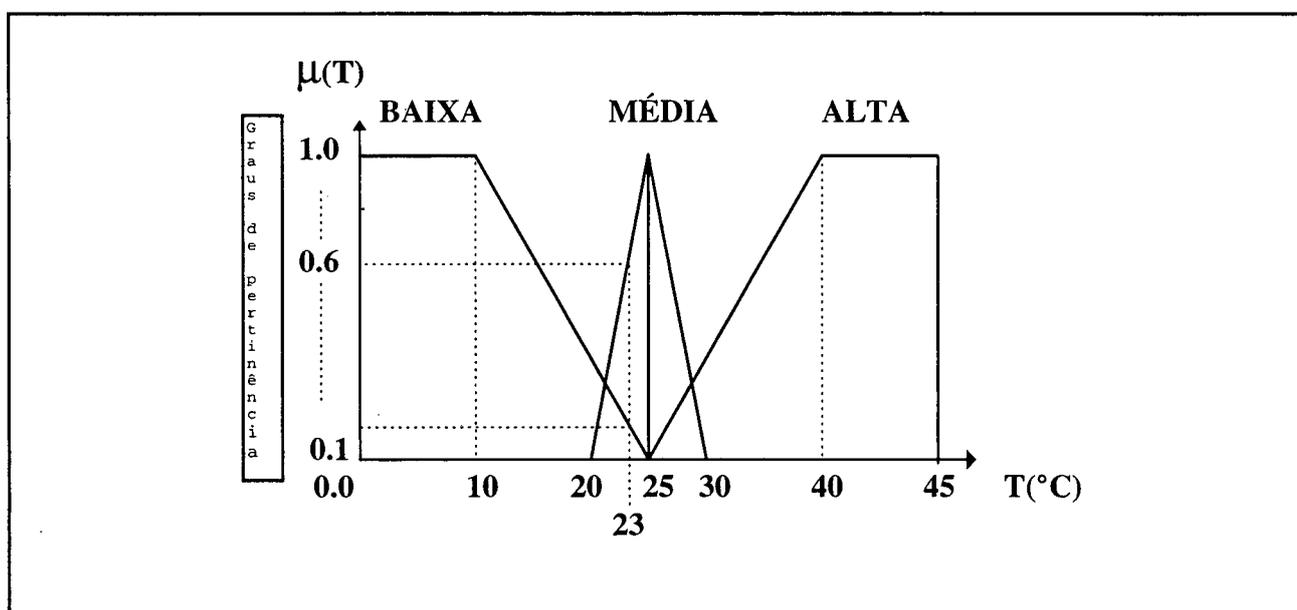


Figura 2.1 - Conjuntos difusos para a variável - TEMPERATURA - em graus Celcius.

2.4 - Função de Pertinência

A função de pertinência é, essencialmente, uma curva distribuída com graus de pertinência que represente melhor o comportamento de uma variável e que denote a pertinência numérica dessa variável. As funções de pertinência podem ser constantes, lineares ou não lineares [Tog91].

Nos programas em linguagem FPL (“Fuzzy Programming Languages”) para serem executados no FC110, pode-se utilizar as funções de pertinência tanto na forma geométrica, através de números que definam a forma da curva, como na forma analítica [Tog91].

2.5 - Conjuntos Difusos

“Fuzzy set” ou conjunto difuso, é a união de uma ou mais funções de pertinência formando um ou mais conjuntos denominados membros que são identificados por variáveis linguísticas. Por exemplo, na figura 2.1, o conjunto difuso (membro) “**BAIXA**” (variável linguística) é formado por uma função de pertinência constante com grau de pertinência unitário até a temperatura de 10°C decrescendo até a temperatura de 25 °C, e acima dessa temperatura, um grau de pertinência zero [Klein91], [Viot93], [Zimmermann93].

O conjunto difuso deve ser determinado baseando-se no sistema em questão [Viot93]. Por exemplo, na figura 2.1 procura-se definir as funções de pertinência difusas em faixas de temperaturas que represente, para uma pessoa, as temperaturas baixa, media e alta, ou seja, com *três variáveis linguísticas - BAIXA, MÉDIA e ALTA*.

Estas variáveis linguísticas são definidas como **MEMBROS** [Tog91], sendo determinados não somente de maneira que os graus e as funções de pertinência representem da melhor maneira o comportamento da variável, mas também com base em experiências.

Na figura 2.1, um outro membro, “**QUENTE**”, pode ser inserido entre as temperaturas de 30°C e 35°C. Ainda, outro membro, “**FRESCO**”, também pode estar situado entre 15°C e 20°C.

Na prática, verifica-se a utilização de conjuntos difusos com um número de membros de 3, 5, 7..., ou seja, números ímpares de elementos difusos, com garantia da intermediariedade e simetria [Lee90], [Klein91], [Palm93], [Barczak94], [Silva95].

Entretanto, *O número de funções de pertinência para o conjunto difuso e a forma da função, escolhe-se de modo dependente da exatidão requerida, responsável pela estabilidade do sistema, facilidade de implementação, manipulação, manutenibilidade.* [Jamshidi93].

A sobreposição entre limites de conjuntos é desejável para a operação suave do sistema [Viot93]. Na figura 2.1, 23°C pode ser uma temperatura baixa e média, mas é média com um grau de pertinência maior ($\mu(T) = 0.6$).

Mudanças nas funções de pertinência (MEMBROS) são aceitáveis para sintonização do sistema de modo a alcançar as respostas desejadas. Geralmente, uma vez que o sistema esteja em operação, a função de pertinência não muda [Viot93].

2.6 - Universo de Discurso

O universo de discurso determina a faixa de atuação do conjunto difuso total que representa determinada variável e representa alguma característica do sistema em questão. [Klein91], [Sugeno85].

Os valores mais comuns, computacionalmente, para o universo de discurso são de -128 a 127 e de 0 a 255 [Tog90], [Tog91], [Chang94]. Para maior comodidade a figura 2.1 está ilustrada com a temperatura cotada em °C.

CAPÍTULO 3

CONTROLE DIFUSO

3.1 - Introdução

Sistemas de controle de processos são sistemas que, em geral, analisam dados de entrada provenientes de sensores, comparam com referências, fazem um processamento de controle, e fornecem uma saída para atuadores que modificam a saída do processo para condições previamente projetadas [Barczak93], [Ogata93],[Viot93],[Jamshidi93].

Um sistema de controle difuso é uma estrutura baseada na lógica difusa tratando as entradas de forma nebulosa e respondendo um valor determinístico na saída. [Viot93], [López94], [Altrock93], [Jamshidi93].

Na figura 3.1, pode-se observar as fases de um sistema de controle difuso.

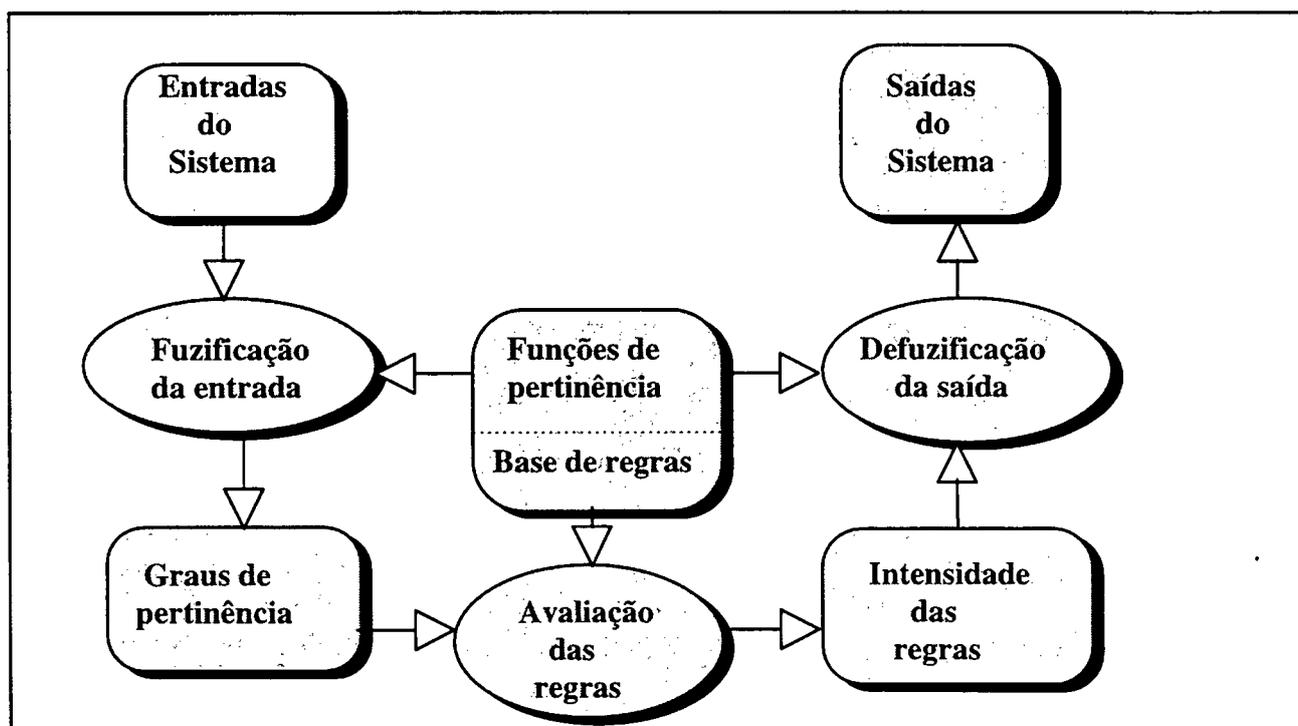


Figura 3.1 - Fluxo de dados de um sistema difuso [Viot93].

Depois de definidas as entradas, há um processo denominado de fuzificação, quando essas entradas são graduadas com graus de pertinência e transformadas em funções de pertinências. Então, é formado um conjunto de regras, que em seguida irá avaliar as condições das entradas fuzificadas e determinar as saídas com suas respectivas intensidades, então a união dessas saídas sofre um processo denominado defuzificação, resultando em um valor determinístico na saída do sistema. [Gomide92], [Viot93].

Como exemplos de aplicação prática de controle difuso, pode-se citar o controle de temperatura [López94], controle de foco de imagens [Altrock93], controle de marcha a ré em caminhões reboque [Runkler94], controle de um posicionador [Barczak94].

3.2 - Fuzificação

A fuzificação, *primeira etapa em um sistema difuso*, é o procedimento de determinar os valores de graus de pertinência das entradas de um sistema em um ou mais grupos, denominados conjuntos difusos [Viot93]. *Portanto, é necessário identificar-se todas as variáveis do sistema para então, fuzificá-las.*

No controle difuso é usual adotar-se duas entradas: uma que represente o erro, outra que represente a derivada do erro (variação do erro) [Barczak93].

A figura 2.1 mostra alguns conjuntos difusos. A entrada, temperatura, é fuzificada com os conjuntos difusos **baixa**, **média** e **alta**. Cada temperatura tem um grau de pertinência em cada um desses conjuntos. O grau de pertinência é determinado por uma função de pertinência que, como observado anteriormente, é feito dependente do sistema e/ou baseando-se em experiências ou intuição. [Palm93]

Ao procedimento de fuzificação estão associados: a escolha das variáveis de entrada e saída, o número e o tipo dos conjuntos difusos, o universo de discurso de cada conjunto difuso, as funções de pertinência e os graus de pertinência [Klein91], [Tog91].

3.3 - Processo de Inferência

Depois de se ter interpretado linguisticamente, através da fuzificação, todas as entradas do sistema “fuzzy”, como próximo passo temos que descrever as situações do processo nas quais há reações, e as reações nessas situações tem que serem indicadas. Isso resulta na formulação de regras do tipo “IF - THEN” que são computadas com o auxílio da inferência. A parte “IF” descreve a situação na qual a regra é válida, a parte “THEN”, a reação referente a situação [Palm93]

A inferência se constitui de dois componentes: agregação e composição. A agregação é a computação da parte de regras “IF”, a composição é a computação da parte de regras “THEN”. [Altrock93]

3.3.1 - As Regras Difusas e Seus Operadores

Em um sistema de controle difuso, existem as regras que contribuem diretamente para avaliar os dados de entrada e determinar uma saída. Essas regras são escritas em uma memória base (*base de conhecimento*) e permanece inalterada, a não ser que o projetista queira modificá-la. [Viot93]

Desta forma, para governar o comportamento de um processo, o projetista desenvolve um conjunto de *inferências*, estruturas declaradas na forma “IF-THEN”. O lado “IF” de uma regra contém uma ou mais *condições* denominadas **antecedentes**, o lado “THEN” contém uma ou mais *ações* denominadas **conseqüentes**. Os antecedentes das regras correspondem diretamente aos graus de pertinência calculados durante a fase de fuzificação. [Zimmermann93]

Cada regra, como saída, pode especificar uma simples conclusão ou lista de conclusões. Cada conclusão é especificada por um nome para uma variável de saída e o nome da função de pertinência (“MEMBER object”) definida nessa variável de saída, separado por um operador ‘=’, ‘==’, ou ‘IS’ [Tog91].

A complexidade das regras de um sistema difuso depende das entradas e das saídas desejadas para esse sistema, ou seja, do próprio sistema.

Para sistemas com apenas uma entrada e uma saída, as regras ficam bastante simples. Contudo, se o número de variáveis de entrada ou o número de membros é maior, e estas influírem de algum modo no processo, as regras são em número cada vez maior e o processo mais complexo. [Jamshidi93]

Nas regras, pode-se empregar o uso de operadores com sintaxe diferente mas que produzem o mesmo efeito (possuem a mesma semântica). Isto é possível, já que a saída não é afetada. Os operadores e as palavras reservadas estarão, doravante, em negrito. [Tog91]

3.3.2 - Agregação

A condição prévia **SE** (parte “**IF**”) de cada regra, relaciona as duas condições, por exemplo: Entrada1 **É não baixa** **E** Entrada2 **É baixa**, (vale precisamente quando valem tanto uma como a outra condição). *O processo de computação da parte SE das regras em um sistema difuso, denomina-se - agregação.* [Altrock93]

No caso da lógica clássica, em que as informações são somente falsas ($\mu=0$) ou verdadeiras ($\mu=1$), pode-se empregar o **E** booleano para estas condições, cujas características estão representadas na tabela 3.1. Define-se “ \wedge ” como o operador mínimo.

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 3.1 - Aplicação da operação E (\wedge) na lógica booleana.

Na lógica de Boole esta operação só pode ser aplicada a termos linguísticos precisos: “verdadeiro” ou “falso”, mas não é possível ser empregada a termos linguísticos vagos ou difusos, pois, o **E** booleano não está definido para agregar informações imprecisas. Por isso, definiu-se

operadores de combinações **E**, **OU** e **NÃO** para as operações lógicas na lógica difusa, que podem também processar os valores intermediários entre o zero e o um (ou entre o verdadeiro e o falso). [Altrock93]

É preciso ressaltar que, em muitas publicações da lógica difusa, são sugeridos operadores (Tabela 3.2) que se empregam hoje nos dispositivos que usam essa lógica. [Altrock93], [López94], [Jamshidi93], etc. Observe que na tabela 3.2 os símbolos \wedge , \vee e \neg são os operadores mínimo, máximo e de negação, respectivamente.

$\begin{aligned} \text{E} & : \mu_{A \wedge B} = \text{mín.} \{ \mu_A, \mu_B \} \\ \text{OU} & : \mu_{A \vee B} = \text{máx.} \{ \mu_A, \mu_B \} \\ \text{NÃO} & : \mu_{\neg A} = 1 - \mu_A \end{aligned}$
--

Tabela 3.2 - Operações Elementares da Lógica Difusa [Altrock93]

3.3.3 - Composição

Os resultados da parte referente à agregação serão utilizados na composição, que fará o processamento da **ação, ou consequência, das regras**, ou seja, a parte “**THEN**”. [Altrock93]

A validade da condição prévia de cada regra descreve o grau ao qual ela está adaptada para a atual situação de processo. Nas técnicas difusas se determina a saída ou resultado de cada regra para a qual valem as condições prévias. Se ocorrer o fato de regras apresentarem a mesma conclusão, tem-se que reunir os resultados de todas as regras válidas para, então, a regra que pesar mais, ser acionada. [Viot93]

Devido as regras estarem formuladas em igual grau no conjunto de regras difusas, elas valem alternativamente, isto quer dizer: “ou vale a regra 1 ou vale a regra 2 ou vale a regra 3 ou O essencial aqui é o **ou** que na tecnologia difusa se pode reproduzir com o auxílio do operador máximo (\vee)” [López94].

Mais adiante será discutido o processo de defuzificação, que determinará um valor único a partir da união das saídas devido a composição.

Uma análise mais eficiente para o entendimento do processo de inferência é feito através do estudo do comportamento geométrico das combinações dos conjuntos difusos das variáveis envolvidas nesse processo [Gomide92]. No capítulo 4 o processo de inferência é mostrado, utilizando-se o sistema difuso do seguidor de superfície, com a aplicação do método de Mín/Máx.

3.4 - Procedimentos Para Determinação da Saída

Terminado o processo de inferência, a ação final é calculada a partir da união das contribuições proporcionadas por cada regra ativada.

Por exemplo, considerando três conjuntos de saída, conforme mostra a figura 3.2, para uma variável de saída denominada “potencia_de_motor” - PM.

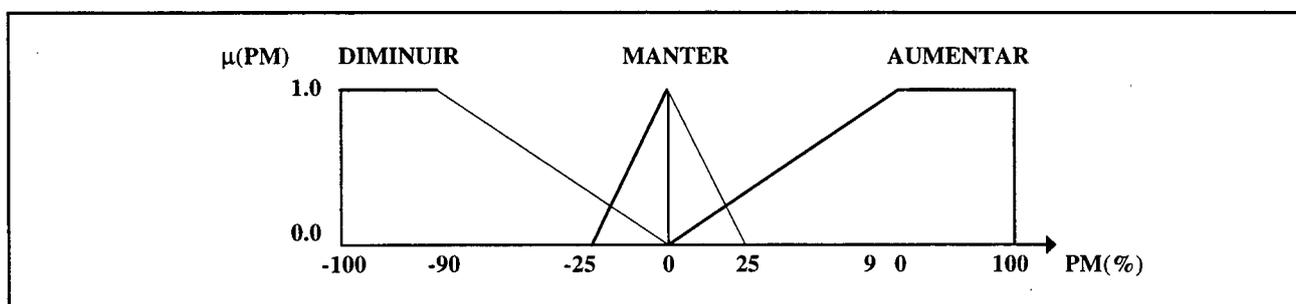


Figura 3.2 - Conjuntos difusos para a variável de saída potência_de_motor (PM) em %.

Considerando que das regras ativadas determinou-se as saídas, diminuir com 0.1 de intensidade, manter com 0.2 e aumentar a potência do motor com 0.4, e se a união é definida como sendo o operador máximo (\vee), o resultado pode ser visto, na figura 3.3 com todos os conjuntos de saída.

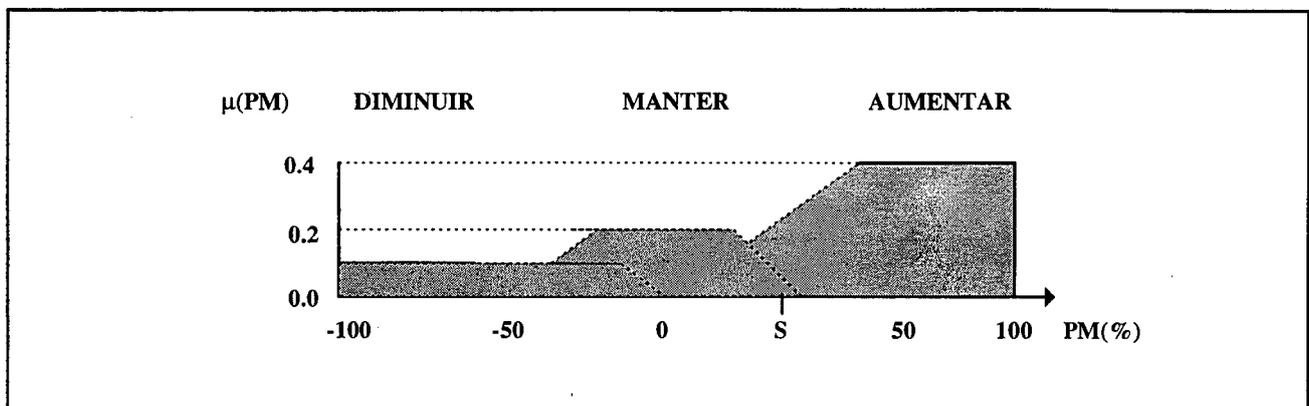


Figura 3.3 - Resultado da união dos conjuntos difusos de saída, referente a contribuição de cada regra ativada, para a variável de saída potência_de_motor (PM) em %.

O sinal de saída S , será determinado no processo de defuzificação, que segue no item abaixo.

3.4.1 - Defuzificação

A defuzificação consiste na determinação do sinal de controle (S) a ser enviado à saída do processo, que no caso do exemplo acima será um sinal que ordenará a diminuição ou a manutenção ou o aumento na potência do motor em um certo nível. [Klein91], [Gomide92]

A defuzificação de saídas é o último passo em um sistema difuso [Viot93]. Apesar de o processo de inferência associar intensidades para cada ação específica, este passo ou defuzificação é necessário por duas razões:

A primeira é decifrar o significado de ações vagas (difusas) como, a ordem é “DIMINUIR” a potência do motor, utilizando funções de pertinência.

A segunda é resolver os problemas de ações conflitantes como, “ordem é “**DIMINUIR**” e “**MANTER**”, que podem ter sido acionadas durante certas condições na avaliação das regras. Alguns métodos para a defuzificação são discutidos em diversos registros como em [Viot93], [Runkler94], [López94], [Jamshidi93].

3.4.1.1 - Método do Centróide

Uma técnica de defuzificação comum, o *método do centro de gravidade ou do centróide*, consiste numa série de passos. [Viot93]

1 - Primeiramente, um ponto centróide no eixo **X** é determinado para cada função de pertinência de saída.

2 - As funções de pertinência são limitadas na altura pela intensidade da regra aplicada, e as áreas das funções de pertinência são computadas.

3 - Calcula-se a área formada pela limitação das alturas com os respectivos conjuntos difusos (área sombreada da figura 3.6).

4 - No final, a saída defuzificada é determinada pela média por pesos dos pontos centróides do eixo **X**, e das áreas computadas, com as áreas servindo como os pesos.

Os cálculos a seguir referem-se a defuzificação das saídas do exemplo de refrigeração. Os cálculos são feitos com todos os conjuntos de saída.

1 - Ponto centróide 1 para o conjunto **DIMINUIR** eixo **X** = $-100/2 = -50\%$

Ponto centróide 2 para o conjunto **MANTER** eixo **X** = 0%

Ponto centróide 3 para o conjunto **AUMENTAR** eixo **X** = $100/2 = 50\%$

2 - Intensidade (altura) aplicada para saída 1 = 0.1

Intensidade (altura) aplicada para saída 2 = 0.2

Intensidade (altura) aplicada para saída 3 = 0.4

3 - Área sombreada para saída 1 = $0.1(100+91)/2 = 9.55$

Área sombreada para saída 2 = $[0.2 (25+20)/2] 2 = 9.00$

Área sombreada para saída 3 = $0.4(100+64)/2 = 32.80$

4 - Média por pesos = $\frac{9.55 (-50) + 9(0) + 32.80 (50)}{9.55 + 9.00 + 32.80} = 22.64$

Portanto o valor do ponto centróide (figura 3.4) C será 22.64 %.

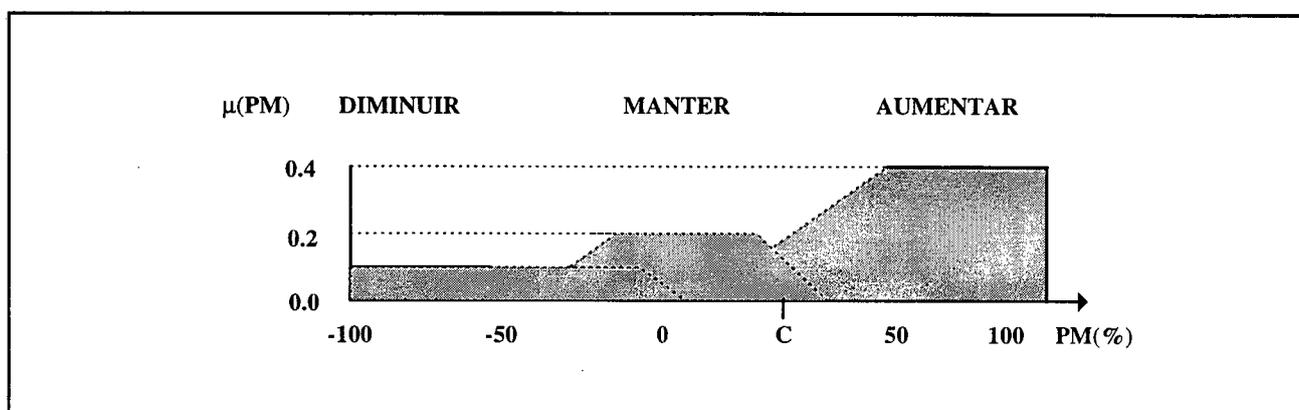


Figura 3.4 - Resultado da defuzzificação. C é o centróide (sinal a ser enviado à saída).

3.4.1.2 - Método das barras verticais (“Singletons”)

Outro processo de defuzzificação decorre da simplificação do método do centróide - o método das barras verticais [Viot93]. Este método é uma função de pertinência de saída representada por uma única linha perpendicular aos pontos centróides dos conjuntos difusos de saída. Veja os cálculos a seguir, referentes as saídas do exemplo anterior.

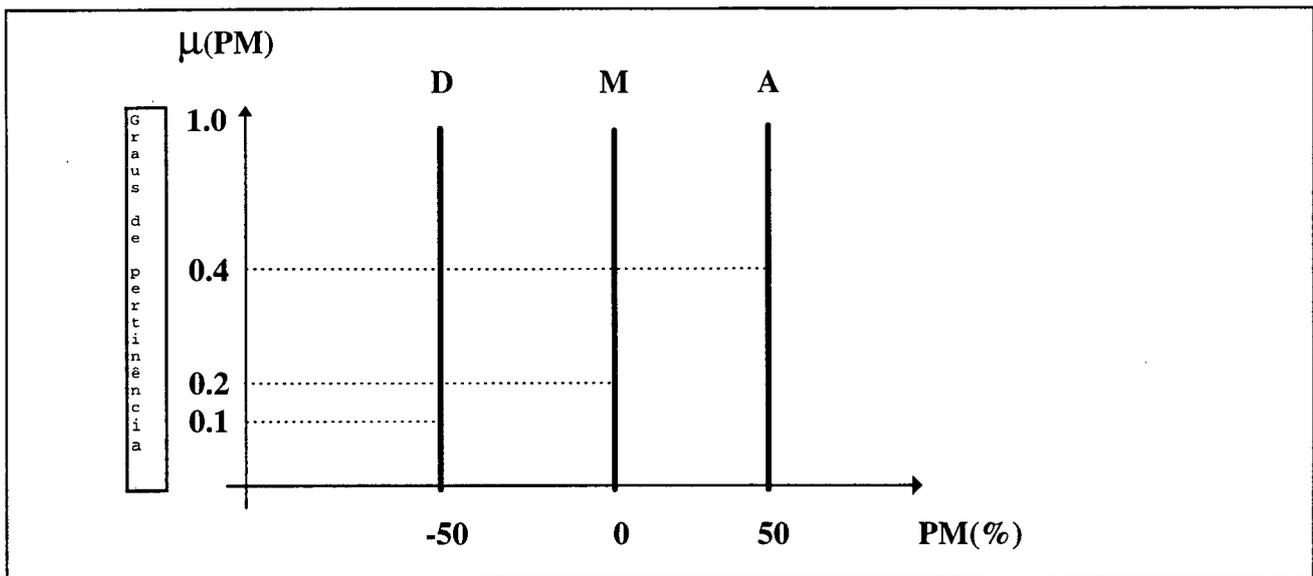


Figura 3.5 - Funções de pertinência de saídas em barras verticais [Viot93].

$$\text{Média por pesos} = \frac{0.1(-50) + 0.2(0) + 0.4(50)}{0.1 + 0.2 + 0.4} = 21.43 \%$$

Uma vez que uma barra vertical intersecta o eixo **X** em um único ponto, então o cálculo do centro de gravidade se reduz apenas ao cálculo da média por pesos dos pontos do eixo **X** (tocados pelas barras) e as intensidades das regras utilizadas como pesos.

3.4.1.3 - Método da Altura

Um outro método de defuzificação, o método da altura, consiste em calcular um valor, determinístico, de saída, com cálculos baseados nos pontos centróides dos conjuntos difusos e nas alturas máximas e de saídas.

O método da defuzificação pela altura utiliza um modelo matemático baseado na equação abaixo [Klein91].

$$D_h = \frac{\sum_{i=1}^n \alpha_i H_i C_i}{\sum_{i=1}^n \alpha_i H_i}$$

onde:

D_h - Defuzificação pelo método da altura;

n - Número de conjuntos difusos de saída;

α_i - Grau de pertinência de saída;

H_i - Altura máxima, observando cada conjunto difuso de saída para todo o universo de discurso;

C_i - Ponto centróide de cada conjunto difuso de saída;

A seguir, é calculado o valor determinístico de saída do exemplo anterior com o método da defuzificação pela altura.

$$SOMA1 = \sum_{i=1}^n \alpha_i H_i C_i = 0.1 \times 0.4 \times (-50) + 0.2 \times 0.4 \times 0 + 0.4 \times 0.4 \times 50 = 6$$

$$SOMA2 = \sum_{i=1}^n \alpha_i H_i = 0.1 \times 0.4 + 0.2 \times 0.4 + 0.4 \times 0.4 = 0.28$$

$$D_h = SOMA1 / SOMA2 = 21.4 \%$$

O modelo matemático utilizado no método da altura, tem a desvantagem de os conjuntos difusos terem que ser simétricos [Viot93], [López94], [Jamshidi93].

Apesar de existirem outros métodos para processos de defuzificação, o método mais utilizado é o do centróide. [Tog91]

Em geral, os métodos de defuzificação retornam um valor resultante, ou seja, *analisam funções de distribuição construídas com graus de pertinências* que através da influência das regras produzem as saídas. Um único valor (determinístico) resulta desse processo [Klein91], [Viot93], [López94], [Altrock93], [Jamshidi93].

CAPÍTULO 4

O MECANISMO SEGUIDOR DE SUPERFÍCIES

4.1 - Introdução

Neste capítulo, faz-se a apresentação da implementação, em laboratório, do dispositivo eletromecânico seguidor de superfícies, com controle difuso, utilizando-se o processador dedicado FC110, específico para processar lógica difusa. Mostra-se as características dessa implementação, da construção do dispositivo seguidor e os resultados obtidos.

A placa com o processador “fuzzy” FC110, cujos detalhes e especificações técnicas estão descritos no apêndice B, está acoplada a um equipamento anfitrião, um PC-AT-386 DX/40 Mhz, com 4 Mb de memória RAM. A arquitetura do sistema é descrita no item 4.1.

No item 4.2 descreve-se a constituição física do sistema, que envolve, um sensor constituído por um potenciômetro linear e uma mesa que é constituída por três eixos (X-Y-Z), acionados por motores elétricos de passo.

No item 4.3 são mostradas as regras propostas para o controle do seguidor e a apresentação de resultados e nos itens seguintes, 4.4, a 4.6, faz-se a descrição dos objetos utilizados nos ensaios, as curvas reais e o plotter. Uma análise do dispositivo seguidor e as conclusões são apresentadas nos itens 4.7 e 4.8, onde se faz referências a possíveis aplicações, fazendo-se algumas sugestões para melhorar a performance do sistema em aplicações que possam aproveitar a idéia do mecanismo aqui apresentado.

4.2 - A Arquitetura do Seguidor Difuso

A figura 4.1 mostra a arquitetura do sistema construído no Laboratório de Microprocessadores e Automação (LMA) do Departamento de Informática e de Estatística (INE) no Curso de Pós-Graduação em Ciências da Computação da Universidade Federal de Santa Catarina.

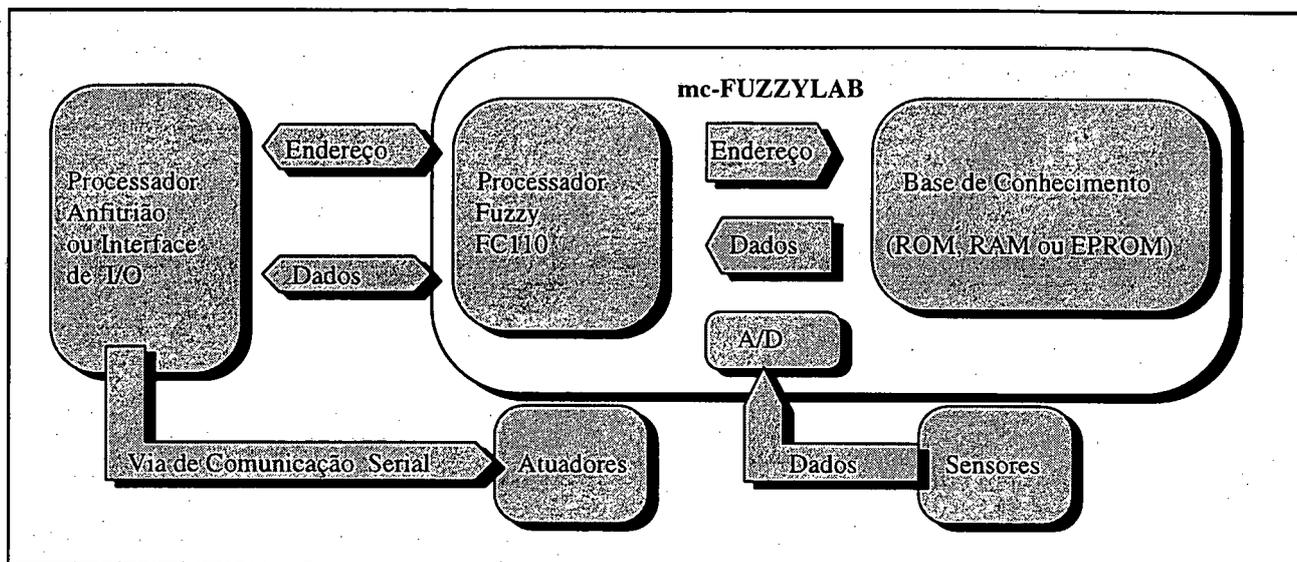


Figura 4.1 - Arquitetura do sistema de controle do seguidor difuso utilizando processador FC110 [Klein91].

A arquitetura do sistema do seguidor de superfícies consta de um processador anfitrião (PC-AT), um atuador (plataforma x-y-z), um sensor (potenciômetro - resistor linear variável), um processador de lógica difusa (FC110, que está na placa mc-FUZZYLAB), um conversor analógico digital e um conjunto de memória RAM que forma a base de conhecimento, também localizados na placa mc-FUZZYLAB.

A transferência de dados entre o sensor e a placa mc-Fuzzylab, é realizada pela entrada analógica/digital da placa mc-Fuzzylab que é conectada ao barramento do PC-AT, em um slot.

Após passar pelo conversor A/D, os dados provenientes do sensor são processados difusamente e a saída deste processamento é levada a endereços de memória do equipamento anfitrião.

A memória do PC-AT é constantemente lida, e os dados resultantes do processamento difuso são utilizados para enviar o sinal de controle para o atuador, que no caso são os motores elétricos de passo que controlam os eixos x e y da mesa x-y-z. A via de comunicação serial do equipamento anfitrião é utilizada para enviar os dados necessários para acionar esses atuadores.

4.2.1- O Acionamento da Mesa X-Y-Z pelo Equipamento Anfitrião

A via de comunicação serial do equipamento anfitrião foi programada para trabalhar a uma taxa de transmissão de 19.200 Baud. A via de comunicação serial é conectada a uma interface de comando que faz parte integrante dos circuitos eletrônicos de potência que acionam os motores de passo da mesa x-y-z. Esta interface contém ainda os circuitos para envio dos passos dos motores que controlam os eixos x-y, como ilustra a figura 4.2.

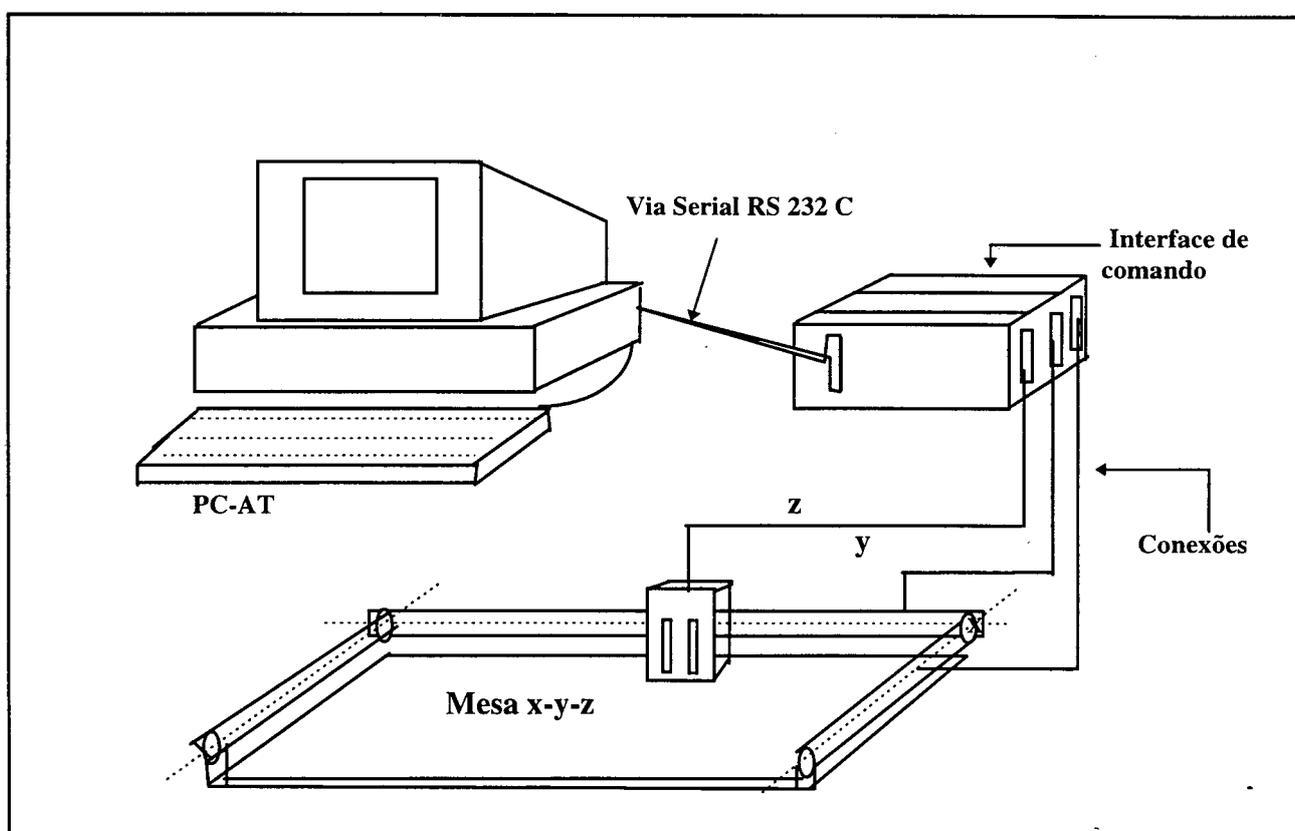


Figura 4.2 - O PC-AT e as conexões com a máquina x-y-z.

A máquina tem três motores elétricos de passo, um para cada eixo. O controle difuso somente se dá no eixo x, pois é através deste eixo que controla-se a aproximação ou o afastamento do seguidor ao objeto a ser seguido. O eixo y somente é utilizado para deslocar o seguidor no sentido do objeto a ser seguido, desde que o eixo x esteja dentro da faixa permitida pelo controlador difuso. O eixo z não é utilizado nos experimentos deste trabalho.

As características da mesa x-y disponível no laboratório, são:

Deslocamento mínimo : 0,01 mm por passo;

Velocidade da mesa x-y-z : 70 mm/s;

Distância máxima de percurso do eixo x : 700 mm ;

Distância máxima de percurso do eixo y : 1000 mm ;

Observa-se que a velocidade de transmissão dos eixos é baixa se comparada com mesas x-y acionadas por motores de corrente contínua e mesmo outros tipos de mesas x-y com motores de passo.

Essa baixa velocidade, entretanto, não prejudica a validade dos experimentos pois o controle é de posicionamento em um eixo. Permite também o acompanhamento visual dos experimentos.

4.2.2 - A Placa mc-FUZZYLAB

O mc-FUZZYLAB é uma placa de programação baseada no circuito integrado, processador FC110 da Empresa Togai, que pode ser instalada em um slot do computador tipo PC-AT. Conforme dados do fabricante, o mc-FUZZYLAB pode, ainda, trabalhar de modo totalmente independente, sem o PC, quando a base de conhecimento for instalada em memória EPROM instalada na própria placa e não instalada com RAM [Klein91].

A base de conhecimento é gravada em memória do tipo RAM localizada na placa, todavia, caso a placa venha a ser utilizada independente do PC, é preciso fazer apenas algumas modificações em alguns “jumper” da placa (JP1, JP2 e JP3).

Como as regras difusas podem ser avaliadas muito rapidamente pelo FC110 (ver apêndice A), o mc-Fuzzylab pode ser utilizado para trabalhar em tempo real para controle de sistemas mecânicos ou eletromecânicos.

A comunicação de dados na placa, se realiza através de circuitos periféricos, que neste trabalho não cabe discutir, e uma área de memória RAM que pode ser lida ou escrita.

Um conversor A/D da ANALOG DEVICES com 8 canais e 8 bits por canal é utilizado na aquisição de dados pela placa. A taxa de conversão é de $2,5\text{v}/\mu\text{s}$ por canal. Um conversor D/A, também integrado na placa, com 4 canais opera igualmente com 8 bits e com a mesma taxa de conversão, contudo, este conversor não é utilizado neste trabalho, pois não há no sistema do seguidor de superfície deste trabalho, necessidade de conversão digital analógica.

Outros detalhes e dados dessa placa encontram-se no apêndice B.

4.3 - Constituição Física do Sistema do Seguidor Difuso

O sistema foi construído como um posicionador bidirecional, conforme descrito no item 4.1.2. São utilizados dois eixos para o movimento do seguidor no plano x-y, contudo o controle difuso de posição é feito somente com o eixo x.

Ao mesmo tempo que o objeto é contornado (apalpado), o braço sofrerá variação do ângulo de sua inclinação em relação à superfície da peça e esta variação fará com que o potenciômetro gire em torno do seu próprio eixo, provocando assim uma variação na tensão dos seus terminais que pode estar entre 0 (zero) e 5 volts, o que corresponde a uma amplitude de 100 milímetros ao longo do eixo x. Conforme a posição do eixo x, mais próximo da superfície ou mais afastado dela, é feito o recuo ou a aproximação do braço que segue o contorno da peça (objeto), conforme indicado na figura 4.3.

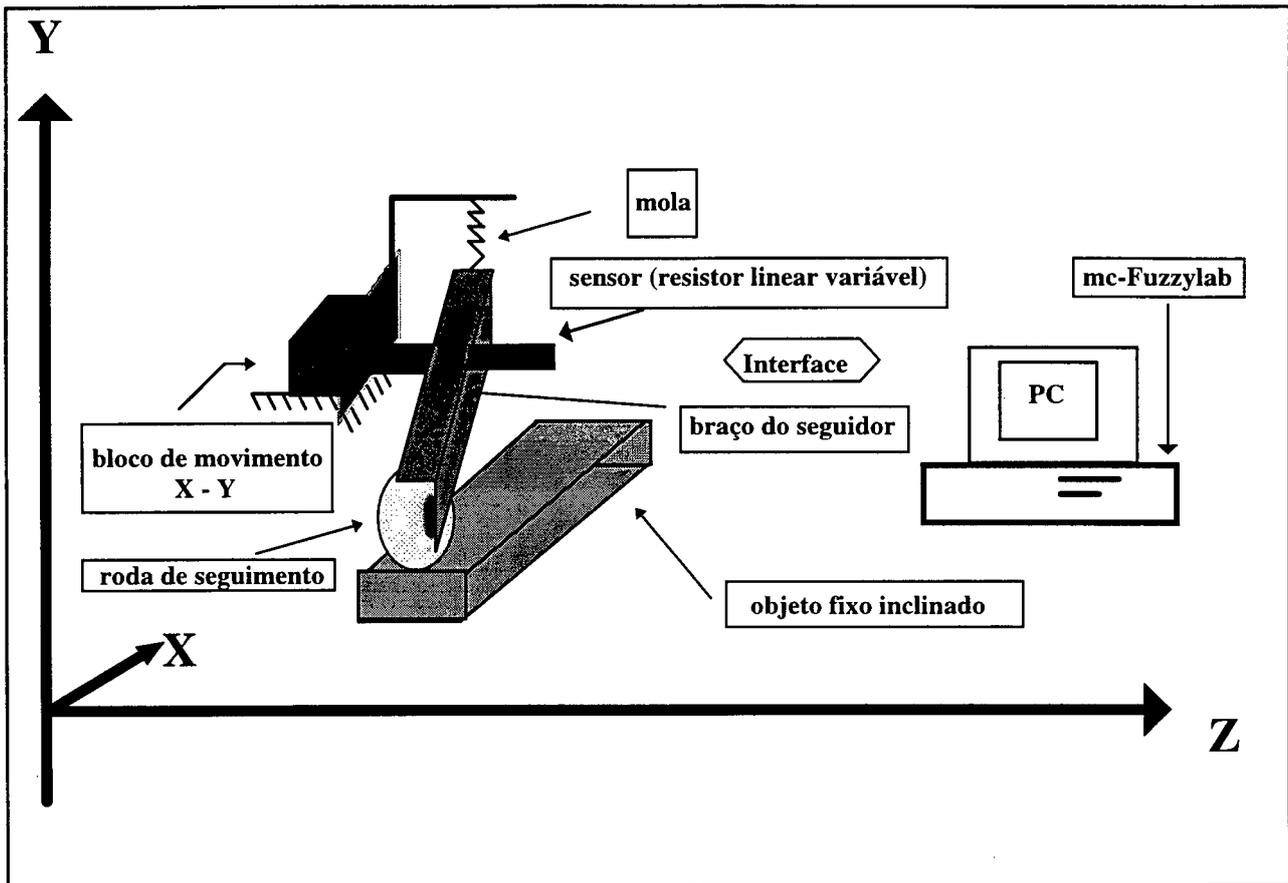


Figura 4.3 - Representação da montagem do seguidor difuso - modelagem do processo.

O braço é de alumínio, leve, com 15 cm de comprimento, largura de 2,5 cm e 1,5 mm de espessura. Um sensor (potenciômetro) sensível às variações de posição do braço é conectado de modo que seu eixo coincide com o eixo de giro do braço. Uma mola mantém o braço junto (tocando) à peça (objeto) de modo a manter a roda de seguimento, instalada na ponta do braço, seguindo o contorno.

O movimento na direção do eixo y é independente, isto é, a sua posição ao longo do percurso não é controlada, pois para manter o braço próximo da superfície do objeto, apenas o controle do eixo x é necessário. O eixo y serve apenas para o braço se deslocar do início ao fim do objeto. A figura 4.4 mostra a fotografia do sistema físico montado.

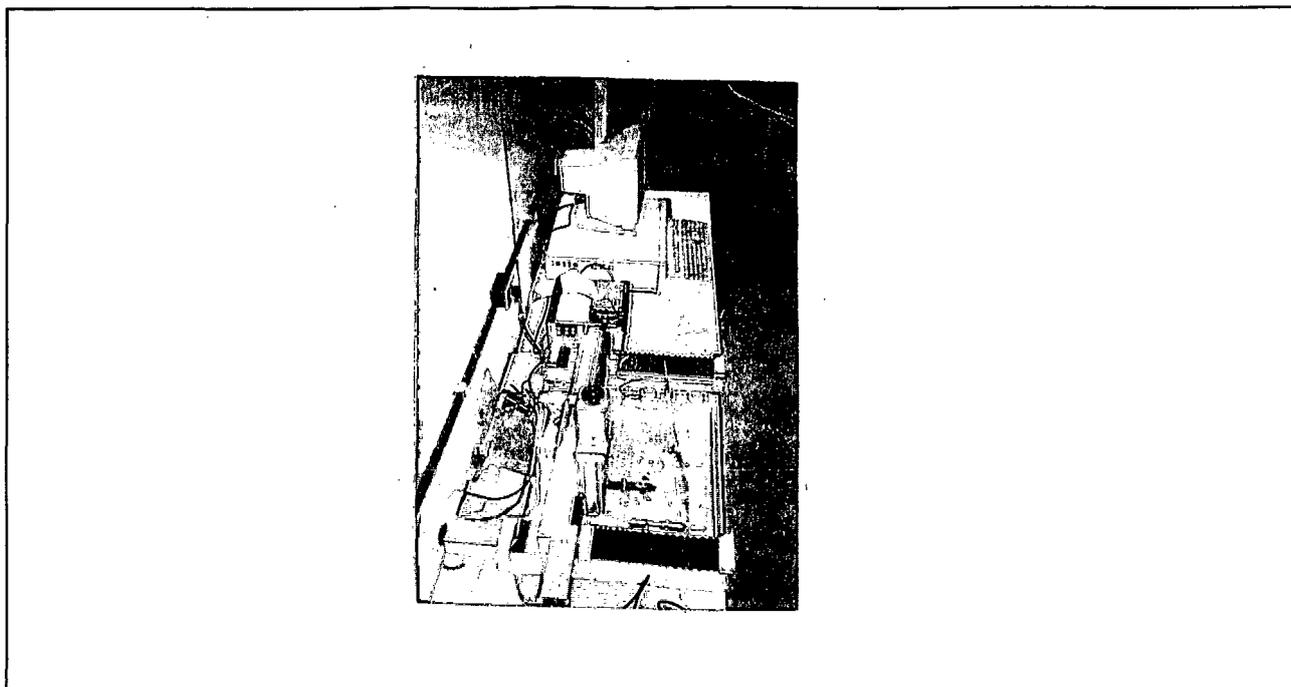


Figura 4.4 - Fotografia do sistema físico do seguidor difuso.

A tensão do sensor é convertida em uma variável digital, que será comparada a um valor pré-definido, usado como referência, que corresponde a um valor médio situado entre 2,0 a 2,6 volts, conforme indica a figura 4.5 (a relação da figura 4.5 foi feita experimentalmente), sem considerar o controle difuso, correspondendo à uma posição média do seguidor situada entre 40 e 56 mm, contados a partir da posição inicial do eixo x. O deslocamento pode se dar nos dois sentidos (ida e volta).

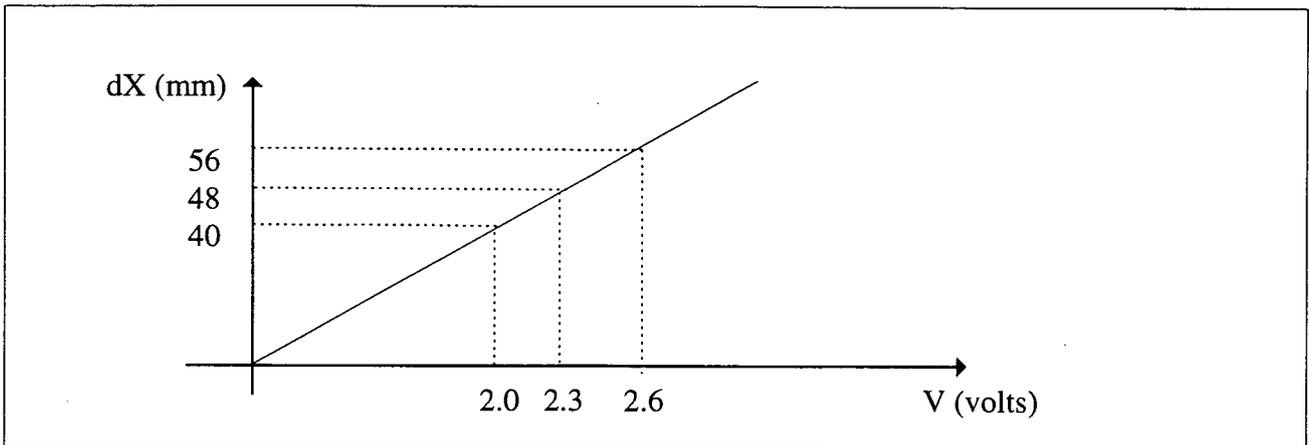


Figura 4.5 - Relação da variação da voltagem no potenciômetro com o deslocamento em milímetro sem considerar o controle difuso.

O diagrama de blocos do conjunto está mostrado na figura 4.6 a seguir.

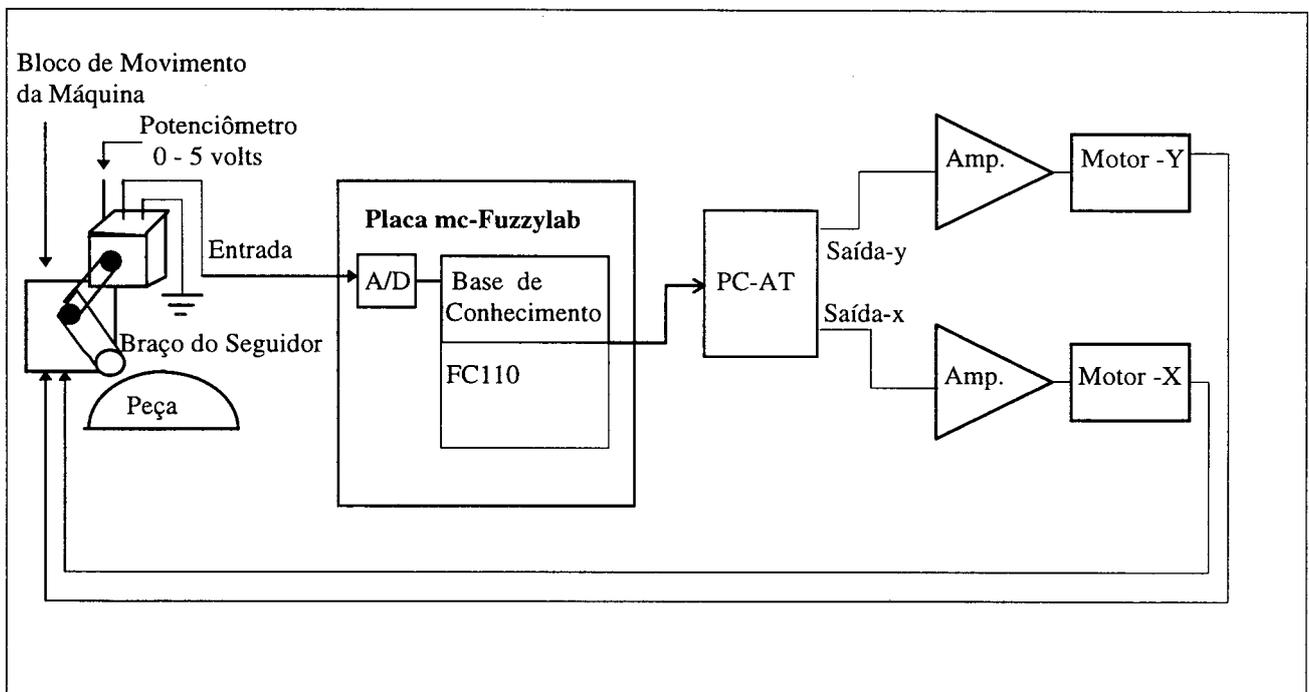


Figura 4.6 - Diagrama de blocos do sistema do seguidor de superfícies.

4.4 - O Controle por Lógica Difusa

Neste item é mostrado o processo de desenvolvimento do controlador difuso. Descreve-se a obtenção dos limites das funções de pertinência e as regras de controle.

4.4.1 - Determinação dos Limites das Funções de Pertinência

As funções de pertinência escolhidas para os membros dos conjuntos difusos, por facilidade, foram triangulares e trapezoidais. Os limites de cada trecho dessas funções podem ser diferentes dependendo do número de trechos e da forma como se considera que cada entrada ou saída influi no comportamento do sistema e dependem de uma adequada avaliação durante o projeto.

Experimentalmente foi considerado o valor 2.3 volts como valor de referência e, após uma série de experimentos, uma faixa de voltagem obtida a partir desse valor. Ver figura 4.5.

A faixa de valores foi determinada com um acréscimo de ± 0.3 volts para o valor central que no caso é 2.3 volts. Este valor central, dentro da base de conhecimento, corresponde a um valor de 128, e fazendo a relação entre a voltagem, chega-se a figura 4.7.

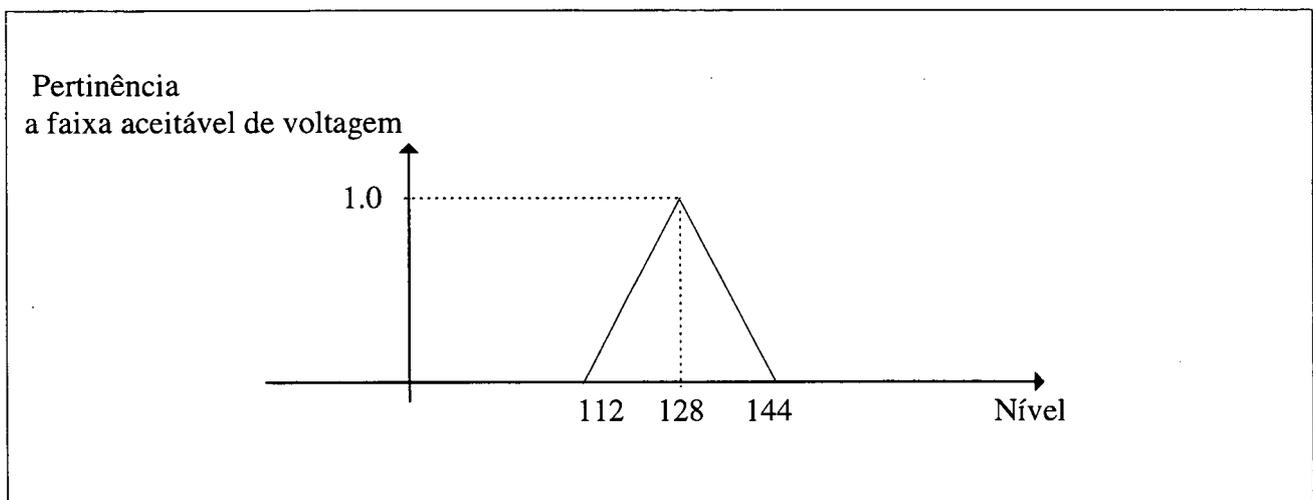


Figura 4.7 - Faixa de valores aceitáveis do nível de voltagem

Note-se que a cada variação de 1 mm na posição x , corresponde uma variação de 0.5375 volts na voltagem dos terminais do potenciômetro e que corresponde a uma variação de aproximadamente 3 da faixa de 0 (zero) a 255 que são os limites admissíveis de valores de entrada e saída dos dados manipulados pela placa mc-Fuzzylab, cuja extensão é limitada a 8 bits.

Para a precisão da máquina, 0.01mm, ocorrerá uma variação de 5.375 mV (milivolts) na voltagem dos terminais do mesmo potenciômetro.

4.4.2 - Definição de Variáveis e Parâmetros

Um arquivo construído na linguagem FPL (“Fuzzy Programming Languages”), denominado **segue110.til** (arquivo .til é o formato de arquivo FPL, próprio para programar a base de conhecimento), cujos detalhes serão descritos neste item, comporta as regras e forma a base de conhecimento para o sistema difuso do seguidor. Uma interface feita em linguagem C e que é formada juntamente com a compilação da base de conhecimento (ver apêndice C, **PROGRAMA SEGUE110.C**), faz a comunicação entre o programa de controle (ver apêndice C, **PROGRAMA DE CONTROLE**) e o programa em FPL.

A saída do processamento da base de conhecimento pelo FC110, fica armazenada na variável de saída, que no caso é a variável **pos_x**. A variável **nível** é a variável de entrada que armazenará a voltagem do potenciômetro.

Estas variáveis podem ser trabalhadas dentro do programa de controle, desde que o programa de interface (Segue110.c), seja incluído no código desse programa de controle.

No projeto do controlador difuso do seguidor de superfícies propõe-se a aplicação de um número variado de regras de controle, para verificar-se na prática, qual a base de conhecimento que é mais adequada ao controle deste tipo de dispositivo eletromecânico, e que estão descritas nos itens 4.3.3 (controle com três regras), 4.3.4 (três regras com limites não coincidentes) e 4.3.5 (cinco regras).

As regras foram escolhidas de acordo com as justificativas expostas mais abaixo e que estão incluídas no final de cada programa. As listagens são mostradas na sequência, em negrito.

A entrada (**var nível**) é fuzzyficada usando-se as funções de pertinência correspondentes aos valores dos membros da(s) entrada(s), aplicando-se as regras escolhidas para obter a saída (**var posição_x**). O método de defuzzyficação utilizado é o do centróide.

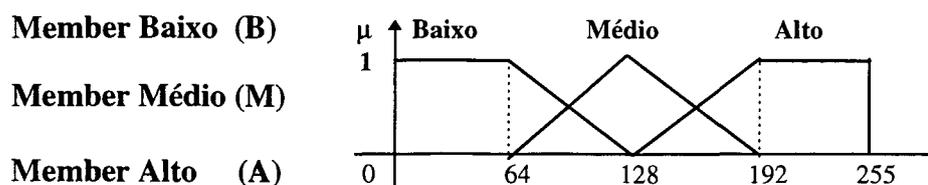
As seguintes variáveis e parâmetros foram definidas:

- . **Project seguidor difuso** - Parâmetro de entrada que indica o nome do projeto da base de conhecimento construída;
- . **var nível** - Define um nome a uma variável de entrada, que no caso denomina-se nível, e que contém a variação de tensão no potenciômetro;
- . **mín 0** - Parâmetro que define o mínimo valor para o universo de discurso;
- . **máx 255** - Parâmetro que define o máximo valor para o universo de discurso;
- . **var posição_x** - Define um nome a uma variável de saída, que no caso denomina-se posição_x, e que determinará o deslocamento do eixo X.

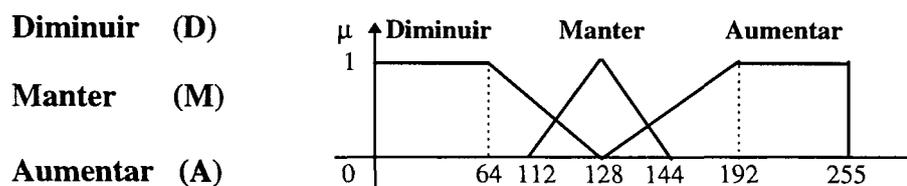
4.4.3 - O Controle com Três Regras Difusas

As seguintes funções de pertinência triangulares e trapezoidais foram definidas:

a) Para a variável - **var nível** - (entrada):



b) Para a variável - **var posição_x** - (saída):



O arquivo abaixo contém o programa em linguagem FPL, (Fuzzy Programming Languages), para o controle difuso do seguidor, com o uso de 3 regras.

```
Project seguidor_difuso
var nível /* variável de entrada - nível de tensão no resistor - afastamento do seguidor*/
mín 0
máx 255
type unsigned byte
member baixo
points 0,1 64,1 128,0
end

member médio
points 64,0 128,1 192,0
end

member alto
points 128,0 192,1 255,1
end
end
```

```

var posicao_x /* variável de saída - aciona o eixo x*/
mín 0
máx 255
type unsigned byte
member diminuir
  points 0,1 64,1 128,0
end

member manter
  points 112,0 128,1 144,0
end

member aumentar
  points 128,0 192,1 255,1
end

fuzzy regras
  rule r1
    if nível = baixo then
      posicao_x is aumentar
    end

    rule r2
      if nível = médio then
        posicao_x is manter
      end

    rule r3
      if nível = alto then
        posicao_x is diminuir
      end
    end
  end
connect from nível to regras end
connect from regras to posicao_x end
end

```

As figuras 4.8 e 4.9 mostram as formas das curvas das superfícies feito pelo seguidor para uma superfície plana e para uma superfície curva. $F(D)$ é a função representativa das irregularidades das superfícies e D é o deslocamento.

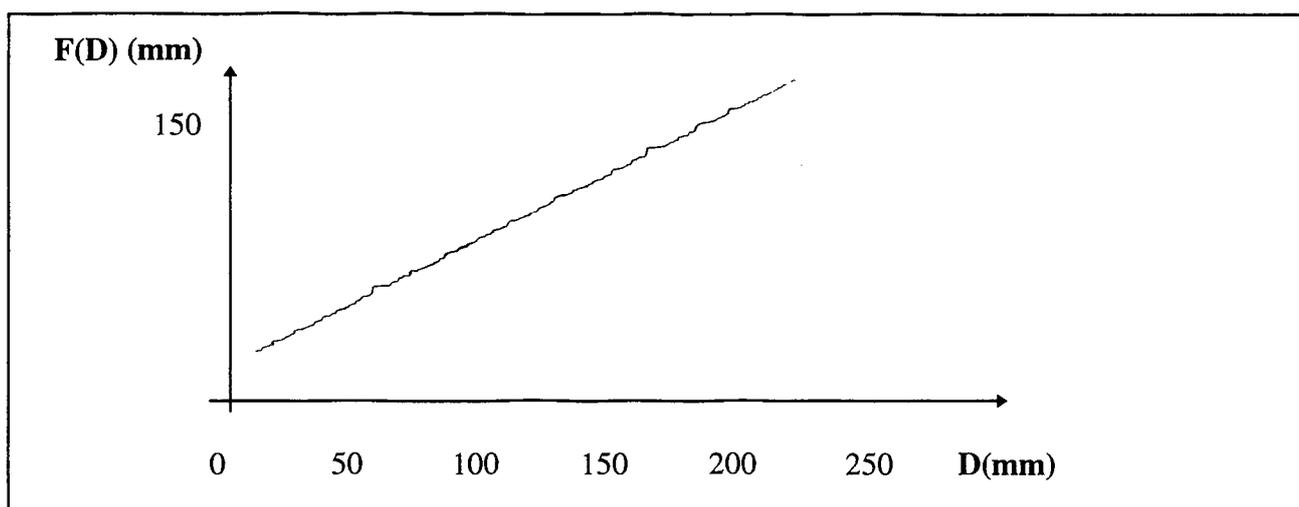


Figura 4.8 - Característica do plano utilizando o seguidor difuso com o uso de três regras.

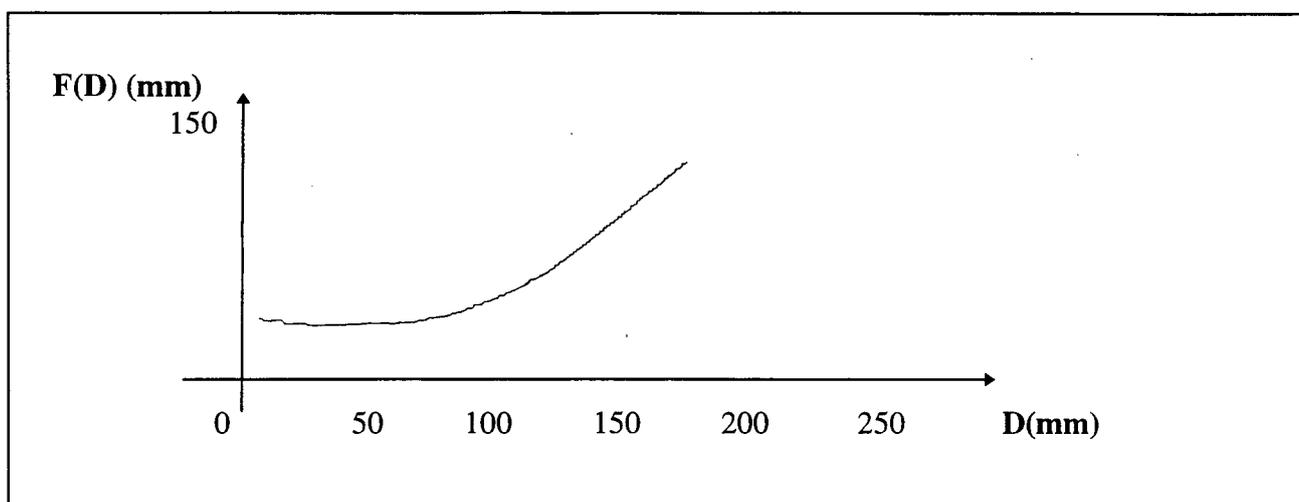


Figura 4.9 - Característica do objeto curvo efetuado pelo seguidor difuso com o uso de três regras.

Os resultados estão resumidos na tabela 4.1 a seguir, onde são mostrados para cada tipo de superfície que foi seguida, o tempo gasto para seguir a superfície do objeto e o erro máximo em milímetros relativo a curva real .

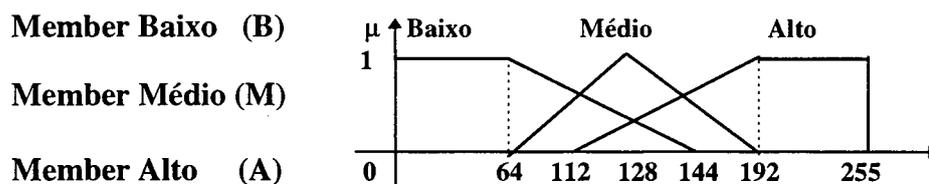
Superfície	Tempo (Segundos)	Erro máx (mm) ± 0.05 (mm)
Linear	258	2,25
Curva	197	1,85

Tabela 4.1 - Tempo e o erro com o uso de três regras

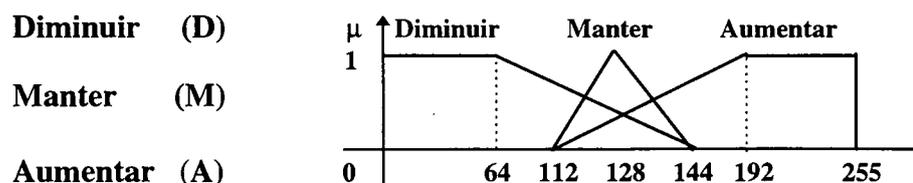
4.4.4 - O Controle com Três Regras com Limites Médio dos Conjuntos não Coincidentes

Neste caso, as funções de pertinência triangulares e trapezoidais foram modificadas fazendo-se com que os limites superior da pertinência baixa e a inferior da pertinência alta não coincidam.

a) Para a variável - **var nível** - (entrada):



b) Para a variável - **var posição_x** - (saída):



O arquivo abaixo contém o programa em linguagem FPL, (Fuzzy Programming Languages), para o controle difuso do seguidor, com o uso de 3 regras com os limites médios não coincidentes.

Project seguidor_difuso**var nível /* variável de entrada - nível de tensão no resistor - afastamento do seguidor*/****mín 0****máx 255****type unsigned byte****member baixo****points 0,1 64,1 144,0****end****member médio****points 64,0 128,1 192,0****end****member alto****points 112,0 192,1 255,1****end****end****var posicao_x /* variável de saída - aciona o eixo x*/****mín 0****máx 255****type unsigned byte****member diminuir****points 0,1 64,1 144,0****end****member manter****points 112,0 128,1 144,0****end****member aumentar****points 128,0 192,1 255,1****end****end****fuzzy regras****rule r1****if nível = baixo then****posicao_x is aumentar****end****rule r2****if nível = médio then**

```

    posicao_x is manter
end

rule r3
  if nível = alto then
    posicao_x is diminuir
  end
end
connect from nível to regras end
connect from regras to posicao_x end
end

```

As figuras 4.10 e 4.11 mostram as formas do traçado feito pelo seguidor para a superfície linear e para a superfície curva, respectivamente, para este caso. Os resultados, conforme feito anteriormente, estão resumidos na tabela 4.2.

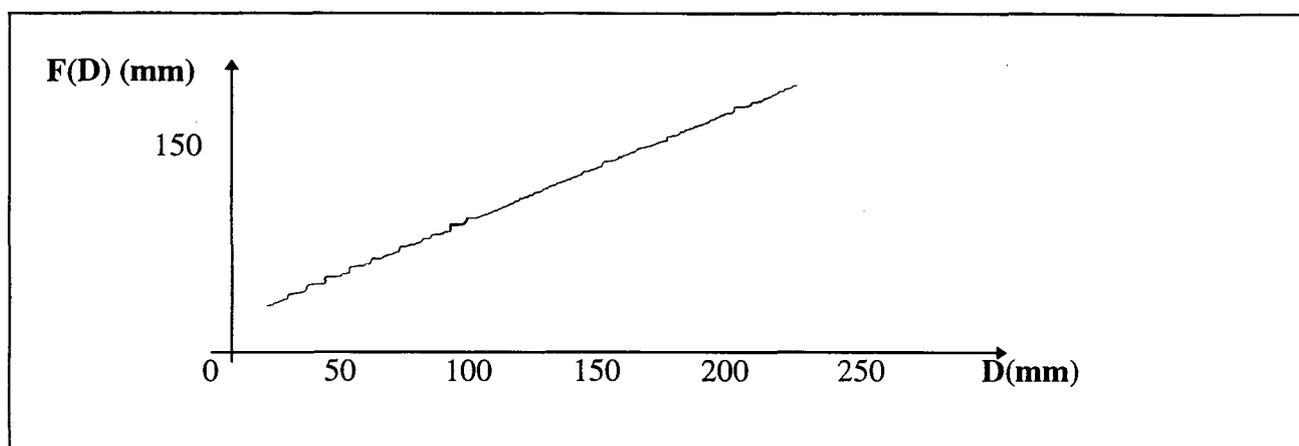


Figura 4.10 - Característica do contorno do plano efetuada pelo seguidor difuso com o uso de três regras com limites dos conjuntos difusos não coincidentes.

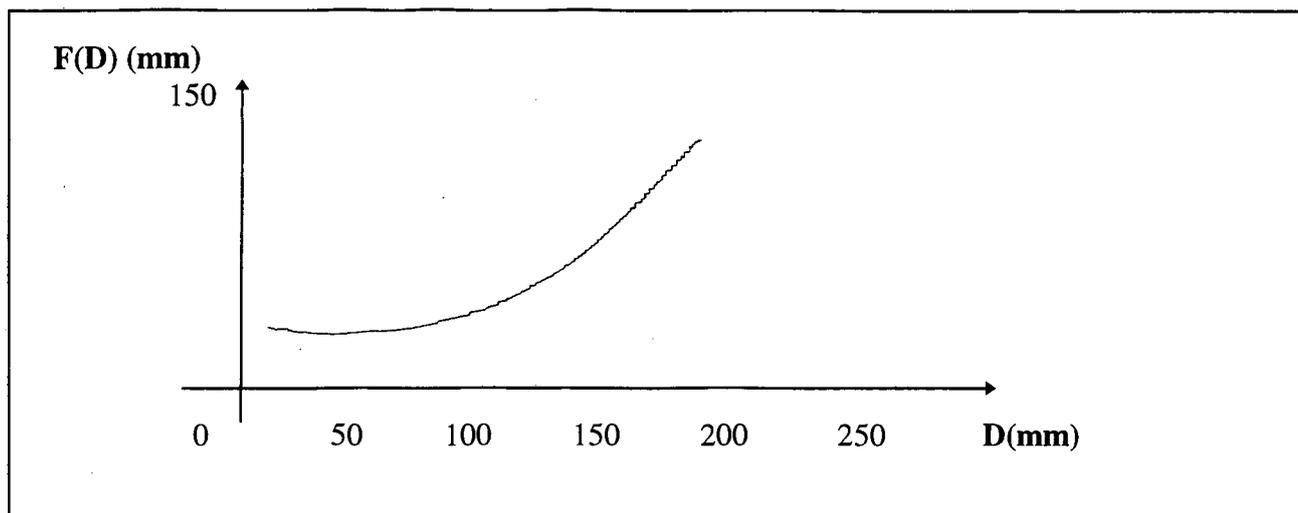


Figura 4.11 - Característica do contorno curvo efetuado pelo seguidor difuso com o uso de três regras com limites dos conjuntos difusos não coincidentes.

Superfície	Tempo (Segundos)	Erro (mm) \pm 0.05 (mm)
Linear	251	2,35
Curva	182	1,90

Tabela 4.2 - Tempo e o erro com limites não coincidentes

Para os dois casos acima (item 4.4.3 e item 4.4.4), onde há três possíveis valores para o nível, (variável de entrada), que corresponde ao valor de tensão do divisor de tensão (sensor), e três possíveis valores para a posição definida no sentido positivo do eixo (variável de saída), resultando em nove regras. No entanto, apenas as três regras indicadas na tabela 4.1 foram usadas para o controle difuso do sistema.

	Posição_x		
Baixo	A	-	-
Médio	-	M	-
Alto	-	-	D

Tabela 4.3 - Regras para o controle do seguidor

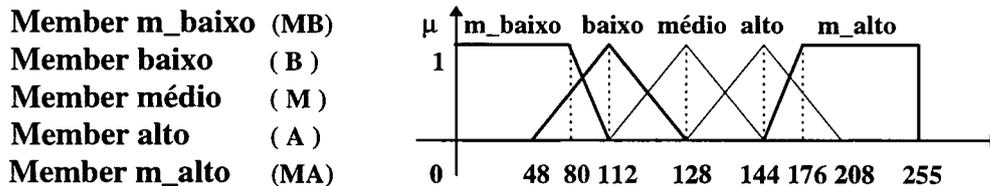
Justifica-se a escolha dessas regras considerando-se que:

- Se o nível é baixo, indicando que a posição deve ser aumentada (braço demasiado próximo do objeto) não há sentido em manter ou diminuir a posição do eixo X;
- Se o nível é médio, não há sentido em aumentar ou diminuir a posição, pois seu valor é o desejado;
- Se o nível é alto, indicando que o braço está demasiado afastado, o comando deve ser de diminuir a posição.

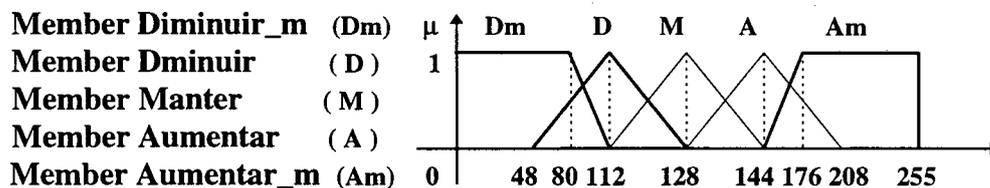
4.4.5 - O Controle com Cinco Regras Difusas

Com o objetivo de verificar-se se um número maior de membros com limites estipulados conforme mostrado neste item, poderá alterar o comportamento do sistema controlado, as seguintes funções de pertinência foram definidas:

- Para a variável - **var nível** - (entrada):



- Para a variável - **var posição_x** - (saída):



O arquivo abaixo contem o programa em linguagem FPL, (Fuzzy Programming Languages), para o controle difuso do seguidor, com o uso de 5 regras.

Project seguidor_difuso

```
var nivel /* variável de entrada*/
min 0 /* nível de tensão no resistor -> afastamento do seguidor*/
max 255
type unsigned byte

    member m_baixo

        points 0,1 80,1 112,0

    end

    member baixo

        points 48,0 112,1 128,0

    end

    member medio

        points 112,0 128,1 144,0

    end

    member alto

        points 128,0 144,1 208,0

    end

    member m_alto

        points 144,0 176,1 255,1

    end

end

var posicao_x /* variável de saída-> aciona o eixo x*/
min 0
max 255
type unsigned byte

    member diminuir_m
```

points 0,1 80,1 112,0

end

member diminuir

points 48,0 112,1 128,0

end

member manter

points 112,0 128,1 144,0

end

member aumentar

points 128,0 144,1 208,0

end

member aumentar_m

points 144,0 176,1 255,1

end

end

fuzzy regras

rule r1

if nivel = m_baixo then

posicao_x is aumentar_m

end

rule r2

if nivel = baixo then

posicao_x is aumentar

end

rule r3

if nivel = medio then

posicao_x is manter

end

rule r4

```
if nivel = alto then  
    posicao_x is diminuir  
end  
  
rule r5  
    if nivel = m_alto then  
        posicao_x is diminuir_m  
    end  
  
end  
connect from nivel to regras end  
connect from regras to posicao_x end  
end
```

As figuras 4.12 e 4.13 mostram as formas do traçado feito pelo seguidor para a superfície linear e para a superfície curva com o controle feito a cinco regras difusas.

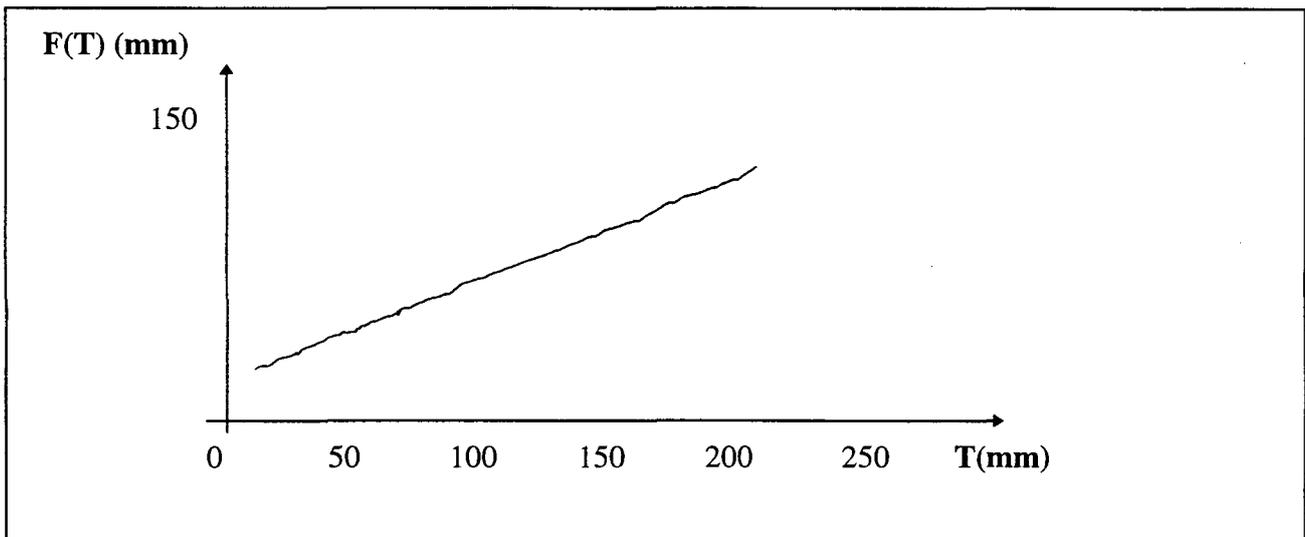


Figura 4.12 - Característica do objeto plano efetuado pelo seguidor difuso com o uso de cinco regras .

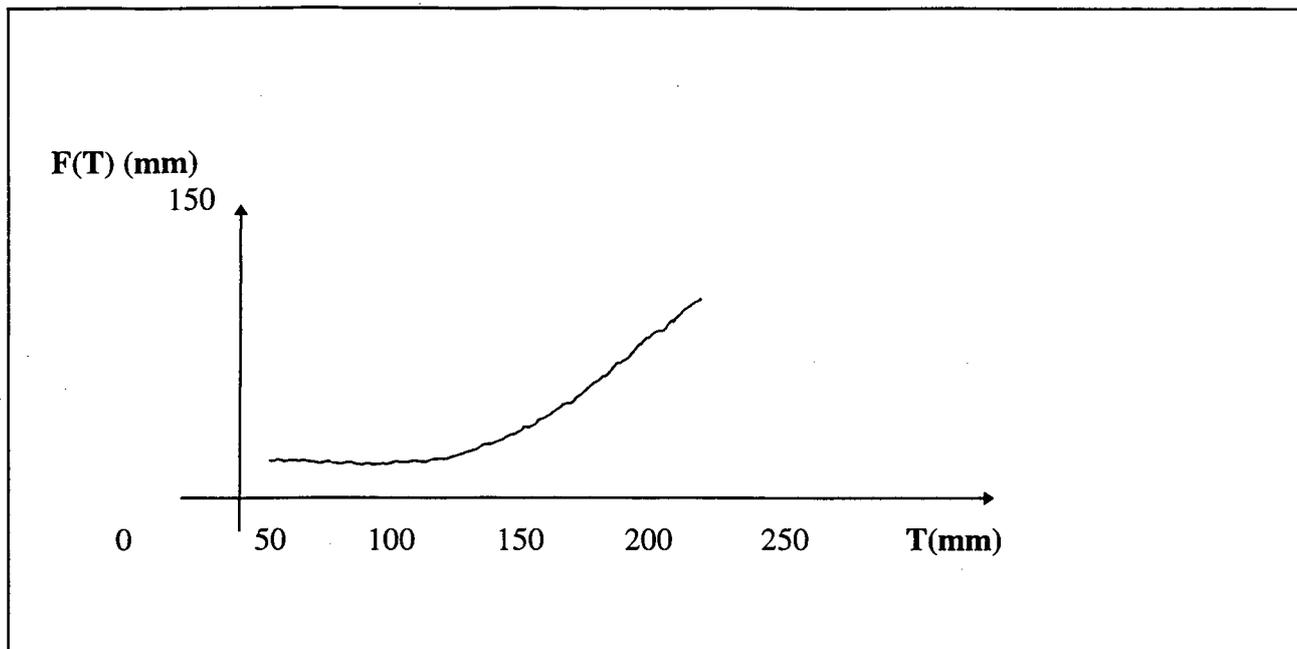


Figura 4.13 - Característica do objeto curvo efetuado pelo seguidor difuso com o uso de cinco regras .

Da mesma forma como anteriormente, pode-se verificar o tempo de percurso e o erro máximo em milímetros na tabela abaixo.

Superfície	Tempo (Segundos)	Erro (mm) \pm 0.05 (mm)
Linear	255	2,35
Curva	181	1,75

Tabela 4.4 - Tempo e o erro com o uso de cinco regras

Há cinco possíveis valores para o nível, (variável de entrada), que corresponde ao valor de tensão do divisor de tensão (sensor), e cinco possíveis valores para a posição definida no sentido positivo do eixo (variável de saída), resultando em vinte e cinco regras possíveis. No entanto, apenas as cinco regras indicadas na tabela 4.5 serão usadas para o controle difuso do sistema.

		Posição_x				
nível	m_baixo	D_m	-	-	-	-
	baixo	-	D	-	-	-
	médio	-	-	M	-	-
	alto	-	-	-	A	-
	m_alto	-	-	-	-	A_m

Tabela 4.5 - Regras para o controle do seguidor

Justifica-se a escolha dessas regras considerando-se que:

- a) Se o nível é muito baixo, indicando que a posição deve ser bastante aumentada (braço demasiado próximo do objeto) não há sentido em manter ou diminuir a posição do eixo X;
- b) Se o nível é baixo, indicando que a posição deve ser aumentada (braço próximo do objeto) não há sentido em manter ou diminuir a posição do eixo X;
- c) Se o nível é médio, não há sentido em aumentar ou diminuir a posição, pois seu valor é o desejado;
- d) Se o nível é alto, indicando que o braço está afastado, o comando deve ser de diminuir a posição.
- e) Se o nível é muito alto, indicando que o braço está demasiado afastado, o comando deve ser de diminuir muito a posição.

A tabela 4.6 mostra o processo de controle e a relação entre o comando dado à variável **posição do eixo x**, conforme os valores da variável **nível**.

VARIÁVEIS		COMANDO - Pos_x		
Pos_x	Nível	Aumentar	Manter	Diminuir
143	125		X	
137	126		X	
148	124	X		
148	124	X		
137	126		X	
143	125		X	
131	127		X	
107	146			X
107	146			X
148	124		X	

Tabela 4.6 - Dados referentes ao processamento da variável nível e da saída pos_x.

A figura 14, mostra através do gráfico **Pos_x X Nível**, o comportamento da variável **pos_x** em relação a variável **nível**. Quando o nível está em uma faixa abaixo do valor de 112 , a posição x é aumentada, e quando está acima de 144 , a posição x é decrementada. Quando a faixa da variável nível está entre 112 a 144, a posição x é mantida.

O gráfico da figura 14a, mostra a relação da entrada nível com a saída pos_x, com o sistema difuso do seguidor de superfícies proposto, atuando.

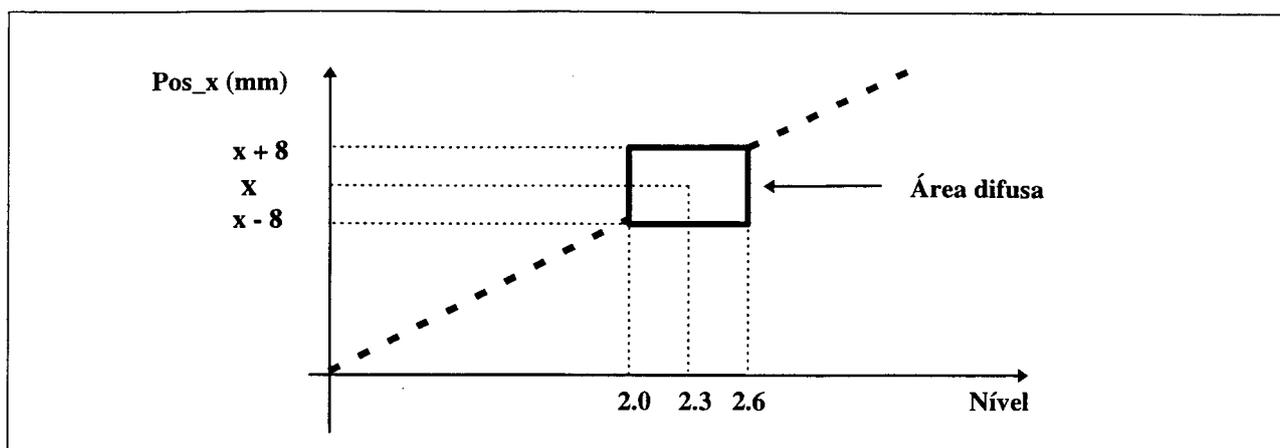


Figura 14a - Gráfico da relação Pos_x X Nível.

Como pode-se observar, no gráfico acima, os valores assumidos pela variável **pos_x** ficam dentro do retângulo. Se a variável nível passar do limite de 2.0 volts para menos ou de 2.6 volts para mais, o controlador difuso atua, e um valor dentro da faixa do retângulo é assumido pela variável **pos_x**.

Note-se que se o controlador difuso não estivesse atuando, o gráfico da figura 14a ficaria idêntico ao da figura 4.5.

4.5 - Determinação da Região de Instabilidade

Durante os ensaios, foram feitos testes com a redução da área difusa (figura 14a, para verificar o comportamento do controlador. Reduziu-se a faixa de 0.1 em 0.1 volts dos extremos de ambos os lados dos conjuntos em que o nível é aceitável. Para uma faixa de valor até 0.4 volts o sistema funcionou, mas com muito tempo de percurso (acima de 10 minutos) e com a faixa de voltagem de 0.2 volts (figura 14b), o controlador se desestabiliza, ou seja, praticamente não sai do lugar e fica repetindo posições no eixo x.

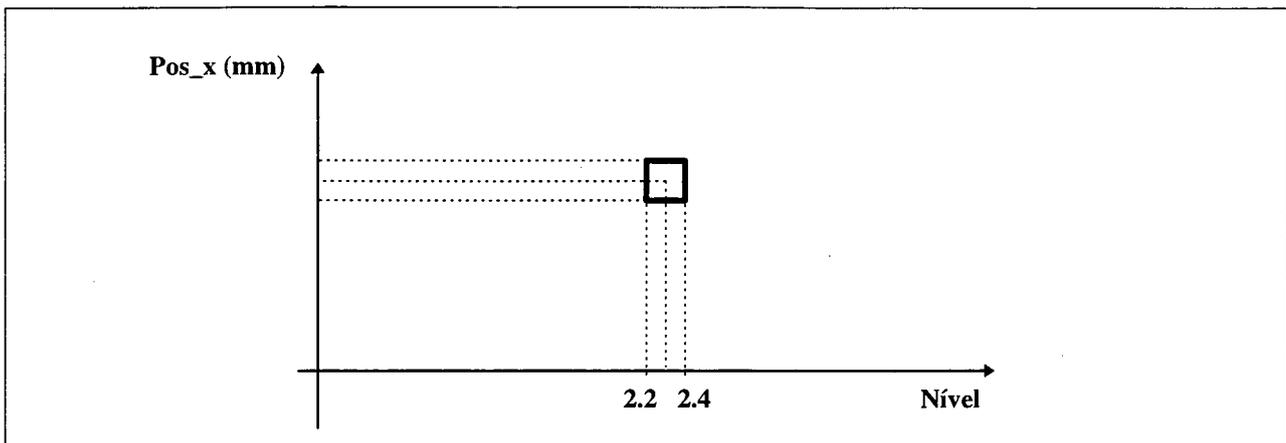


Figura 14b - Gráfico da região de instabilidade do seguidor de superfícies.

4.6 - Avaliação dos Experimentos

Para poder-se analisar os resultados, os seguintes procedimentos foram seguidos:

1 - Realização dos ensaios:

- Procedeu-se uma bateria de ensaios para cada tipo de controlador difuso proposto conforme itens 4.4.3, 4.4.4 e 4.4.5.

2 - Obtenção do traçado das superfícies reais por meio de método manual.

3 - Comparação das curvas obtidas com as curvas reais.

4 - Medição dos erros por meio de um paquímetro com precisão de 0.05 mm.

As curvas dos traçados acima, foram desenhadas com um plotter próprio da máquina X-Y-Z, que foi fixado na base do eixo z, como é mostrado na figura 4.18. Desta forma, ao mesmo tempo que o seguidor começa o processo do traçado é desenhada a linha da superfície apalpada pelo seguidor.

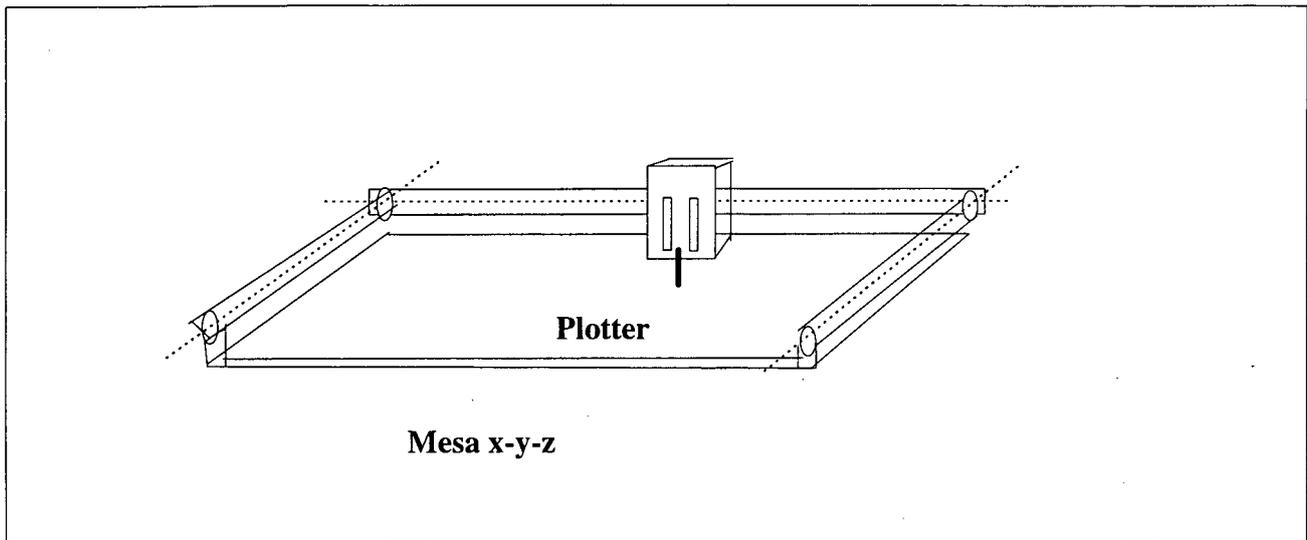


Figura 4.15 - O plotter.

Em todos os traçados obtidos anteriormente deve-se observar que a máquina X-Y-Z contribui com imprecisões da ordem do centésimo do milímetro, quando do avanço ou recuo de cada passo dos motores (segundo dados do fabricante).

4.7 - Os Objetos Utilizados e as Curvas Reais

A figura 4.16a e 4.16b mostram, respectivamente, a superfície plana (linear) e a superfície curva (não linear) dos objetos que foram utilizados nos ensaios.

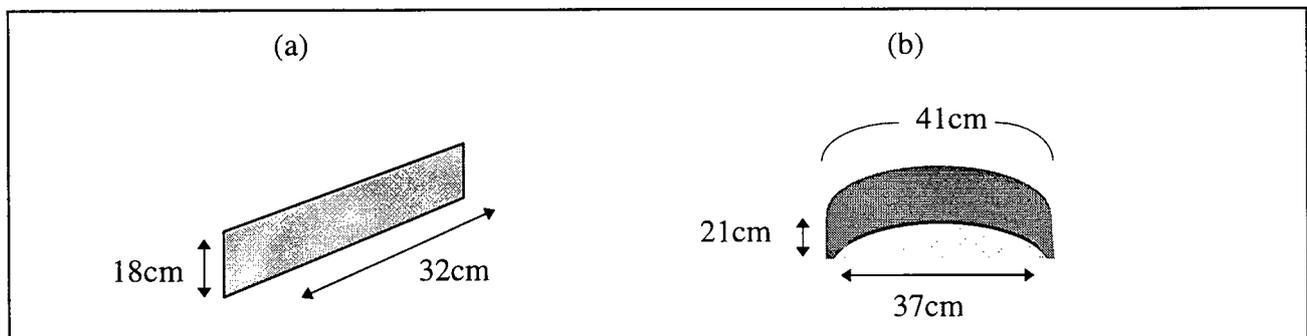


Figura 4.16a e 4.16b - Superfícies utilizadas para análise do controlador difuso do seguidor.

Devido as limitações da máquina X-Y, os percursos foram limitados a um deslocamento de 250 mm para ambos os tipos de superfície.

As figuras 4.16 e 4.17, a seguir, mostram a função real, $F(D)$, que representa o traçado da superfície de cada objeto.

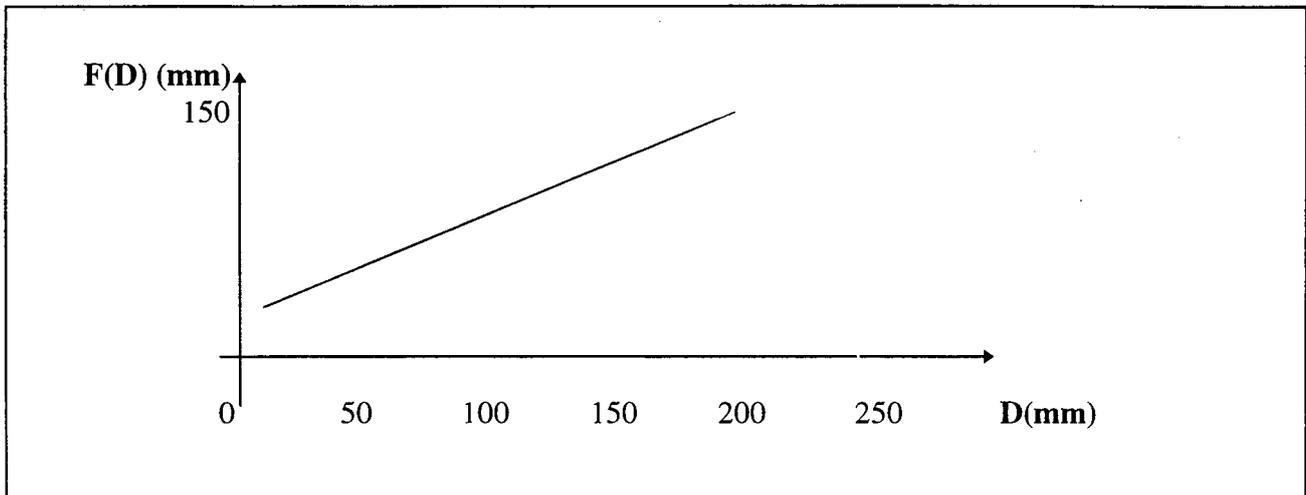


Figura 4.17 - Função característica do contorno da rampa (plano inclinado)

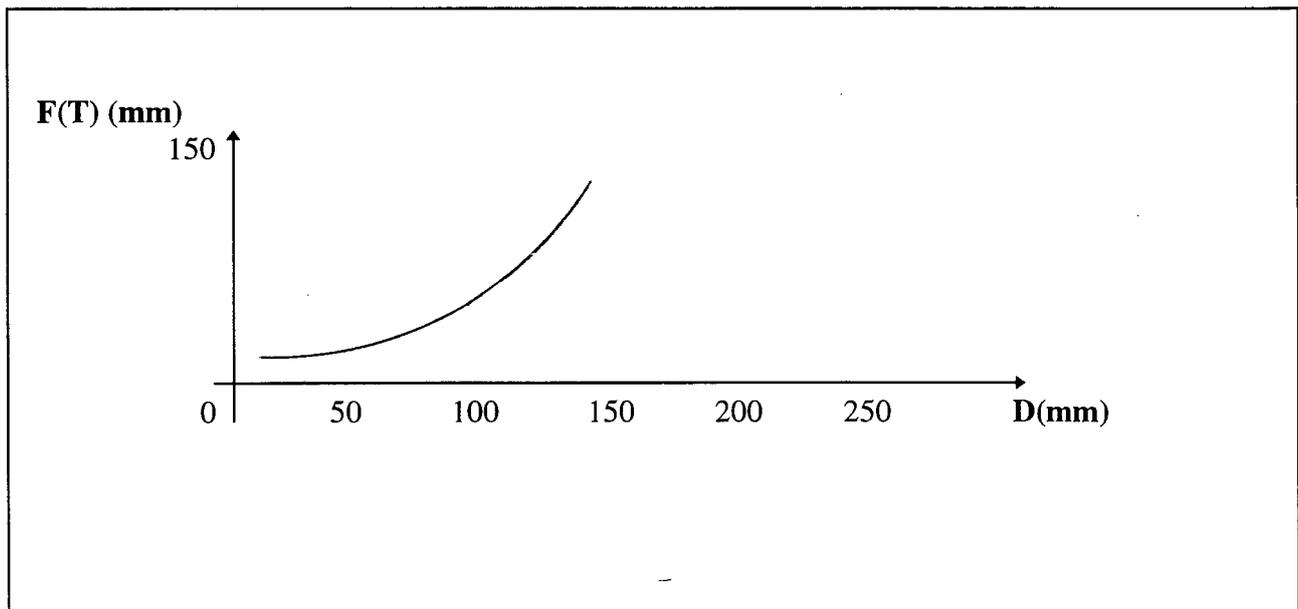


Figura 4.18 - Função característica do contorno do objeto curvo

4.8 - Conclusão

Os erros e os tempos de percursos, vistos nas tabelas anteriores, permite dizer que para os três controladores, não houve grandes diferenças nos resultados, portanto, o uso de três regras de controle para o seguidor, por facilidade, pode ser utilizado.

No contorno do plano inclinado, o erro máximo encontrado foi de 2,35 ($\pm 0,05$) mm e no traçado não linear o erro máximo foi de 1,90 ($\pm 0,05$) mm.

Conforme os resultados vistos nas tabelas e gráficos anteriores, o seguidor de superfícies proposto, mostrou na prática, que é possível constituir um sistema que acompanhe a forma de superfícies de objetos com um erro que não ultrapasse mais que um valor absoluto de tempo de 7 segundos de percurso e de 0.15 mm para o erro de posição.

CAPÍTULO 5

CONCLUSÕES E RECOMENDAÇÕES

5.1 - Conclusões do Trabalho

Devido aos equipamentos com que se construiu o sistema, alguns problemas surgiram, como vibração da mesa x-y, vibração da barra seguidora, vibração da própria bancada de ensaios, atraso nos comandos dos motores de passo devido a configuração da máquina, que quando recebe um comando para avançar ou retornar um passo em apenas um dos eixos, tem que ser processado os passos de todos os eixos.

O uso do resistor como sensor, só possibilita ler a posição no instante que a barra seguidora está percorrendo a peça, contudo, se fosse usado um sensor fotoelétrico (utilizado em câmeras de vídeo), por exemplo, poderia-se obter os dados relacionados com a posição antes de ser iniciado o processo de controle. Com o uso desse tipo de sensor, a barra seguidora, que também contribui com erros, seria suprimida e muito seria ganho em termos de desempenho.

Ainda, pelos resultados obtidos e descritos nos itens anteriores, pode-se perceber que o controlador difuso produziu resultados comparáveis com as curvas reais, e os erros resultantes são suficientemente pequenos se o sistema for aplicado ao controle de processos que não exijam precisão de posicionamento superior à obtida.

O seguidor difuso proposto mostrou na prática, que o uso do controle difuso para dispositivos que tenham problemas de imprecisão é de solução muito mais simples, visto que ao invés de se adquirir equipamentos mais precisos e portanto caros, pode-se utilizar uma faixa de valores permissível em que o processo de controle consiga contornar os problemas dessas imprecisões.

Por exemplo, no caso do seguidor de superfícies proposto ao ser restringida a faixa de valores da área difusa continuamente, em que a posição x seja, ainda, aceitável (figura 14), chega-se a um certo valor, conforme visto no item 4.4, que o controle se desestabiliza. Desta forma, uma faixa de valores difusa, válida, contorna o problema sem maiores complicações ou custos. De outro lado, se

suprimir-se a área difusa, o controle não consegue responder adequadamente as necessidades do projeto, ou seja, contornar com precisão as superfícies de objetos como as dos ensaios aqui efetuados.

No uso do dispositivo seguidor de superfícies como controlador de posição, para aplicações mais elaboradas, pode-se conseguir melhorar o seu desempenho através da substituição de equipamentos pouco velozes e da substituição do sensor.

5.2 - Recomendações

Em certos casos, é recomendável que os motores tenham grande velocidade, podendo-se assim, possibilitar aplicações que também dependam desse fator.

Na hipótese de que venha-se a aplicar um dispositivo semelhante ao seguidor de superfícies proposto, por exemplo, em suspensão ativa [Barbieri91], fresamento [Schrader66], [Dallas76], solda [Schrader66], etc., pode-se pensar nas sugestões acima, que também fazem parte da conclusão deste trabalho, como um passo para melhorar a confiabilidade do sistema.

“Para obter-se uma boa solda, deve-se levar em consideração uma série de fatores como o ponto de temperatura adequada, o tipo de material a ser soldado entre outros.

...Uma soldagem feita em uma mesa que gira para colocar a peça na **POSIÇÃO** mais precisa e vantajosa são chamados dispositivos para solda” [Schrader66].

“ Quanto ao fresamento de peças metálicas, por exemplo, o mesmo tipo de superfície pode muitas vezes ser fresado de várias maneiras ou **POSIÇÃO**. Por exemplo, superfícies planas podem ser usinadas por fresamento cilíndrico paralelo, fresamento lateral, ou fresamento por faceamento. Os métodos usados para fresagem, tem que levar em consideração a **forma da peça** e a **posição da superfície**” [Schrader66].

Para aplicação em suspensão ativa, “analisa-se o comportamento dinâmico de um veículo automotivo através de um modelo mecânico onde são consideradas a rigidez das molas e coeficientes de amortecimento dos amortecedores.

As finalidades conflitantes, segurança (suspensão dura) e conforto (suspensão macia), envolve a necessidade do conhecimento do comportamento próprio e excitado do veículo, para que possam ser estabelecidos tipos de controle e otimizações do sistema. Estes conflitos tem sido contornados utilizando-se novos tipos de suspensões (ativas, semi-ativas, etc.) e a posição $x \pm dx$ em amortecedores e molas é fundamental para o bom funcionamento de uma suspensão automotiva” [Barbieri91].

Em outros processos, como por exemplo, no fresamento [Dallas76], na suspensão ativa [Barbieri91], solda [Schrader66], cirurgias de precisão monitoradas, ou outras semelhantes, algumas adaptações deverão ser feitas, mantendo-se, porém, a estrutura geral do sistema.

Trabalhos realizados com outros tipos de posicionadores mostram que essas hipóteses são válidas e as aplicações possíveis. Em [Barczak 94], faz-se o controle de um posicionador em apenas uma direção, com aplicação em usinagem.

Por outro lado, conforme informações obtidas no Hospital Universitário (HU) (Laboratório de Técnica Operatória e Cirurgia Experimental), o transplante de pele é uma situação em que envolve o operador humano em um caso de muita imprecisão, pois todo o processo de retirada da pele tem que ser feito manualmente com um dispositivo com uma lâmina, semelhante a um “fatiador de queijo” (**dermátomo**). Desta forma, seria possível fazer uma pesquisa para verificar o desenvolvimento de um dispositivo que substituísse o **dermátomo**, aparelho utilizado na medicina para fazer a retirada da pele, manualmente, ao fazer o transplante e que conseguisse fazer a retirada de camadas de pele com boa precisão.

REFERÊNCIAS BIBLIOGRÁFICAS

[AD92a] Analog Devices. Data Converter Reference Manual, Vol I, 1992.

[AD92b] Analog Devices. Data Converter Reference Manual, Vol II, 1992

[ADI-T88] ADI-Treiber. Isert-Electronic. Dez., 1988.

[Altrock93] Altrock, Contantin V.. Fuzzy Logic. Band 1 - Technologie, 1993.

[Barbieri91] Barbieri, N., Zampieri, D. E. Análise do Comportamento Dinâmico de um Veículo Automotivo para Variações dos Parâmetros das Suspensões. A.E.A. - Associação Brasileira de Engenharia Automotiva. Seminário de “Segurança Veicular”, 6 a 7 de junho, 1991.

[Barczak93] Barczak, Czeslau L., Controle Digital de Sistemas Dinâmicos Projeto e Análise. Grucon - Universidade Federal de Santa Catarina, 1993.

[Barczak94] Barczak, Czeslau L., Rosa, Leonardo C., Martin, Carlos A., Krambeck, Cleverson P. A Fuzzy Controller For N.C. Machine Tool Positioning System. IASTED 94 International Conf., Ottawa, 1994.

[Bezdek79] Bezdek, James C. Pattern Recognition With Fuzzy Objective Function Algorithms. Utah State University. Logan, Utah, 1979.

[Biggerrstaff86] Biggerrstaff, Ted J. Systems Software Tools. Prentice-Hall, 1986.

[Brady82] Brady, Michael, Hollerbach, J. L., Johnson, T. L., Perez-L., T., Mason, M. T. Robot Motion: Planning and Control. Cambridge, Massachusetts and London, England, 1982.

[Brubarker91] Brubarker, David L., Ph.D., President the Huntington Group. Fuzzy logic works with degrees of truth. Computer Design, March, 1, pp 98-102, 1991.

[Chang94] Chang, Z., Tan, J. Linearity and A Tuning Procedure For Fuzzy Logic Controllers. Transactions, vol. 37(3), pp. 973-979, 1994.

[Chieh93] Chieh, Chieh-Li, Chen, Pey-Chung. A Pneumatic Model-following Control System Using a Fuzzy Adaptive Controller. Automática, vol. 29, nº 4, pp.1101-1105. Printed in Great Britain, 1993.

[Dallas76] Dallas, D. B. Tool and Manufacturing Engineers Handbook. Society of Manufacturing Engineers, 1976.

[David91] David, A. K., and Rongda, Z. An Expert System With Fuzzy Sets for Optimal Planning, IEEE Trans. on Power Systems, feb. 1991, pp. 59-65.

[Dubois80] Dubois, Didier. Fuzzy Sets And Systems (Theory and Applications). Academic Press-United States of America, 1980.

[Filho93] Filho, Humberto Ferasoli, Pegararo, Renê, Franchin, Marcelo N., Saito, J. Hiroki. Um Sistema Multiprocessador de Imagens Utilizando Processadores Digitais de Sinais. XIII Congresso da Sociedade Brasileira de Computação - Anais - Florianópolis, 1993.

[Franklin86] Franklin, F., G., Powell J. David, E-N, Abbas. Feedback Control of Dynamic Systems. Addison-Wesley Publishing Company, 1986.

[Galichet95] Galichet, Sylvie and Foulloy. Fuzzy Controllers: Synthesis and Equivalences. IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 3 Nº 2, MAY, 1995.

[Gomide92] Gomide, F., Gudwin, R., Netto, M. L. Andrade. Controle de Processos por Lógica Fuzzy. Unicamp - FEE - DCA, Campinas - SP, 1992.

[Gomide94] Gomide, F., Gudwin, R. Modelagem, Controle, Sistemas e Lógica Fuzzy. Unicamp - FEE - DCA, Campinas - SP. SBA Controle & Automação/vol. 4 nº3, set/out, 1994.

[Graham88] Graham, Ian and Jones, Peter Llewelyn. Expert Systems Knowledge, Uncertainty and Decision. London-New York, 1988.

[Han93] Han, Jia-Yuan and McMurray, Vicent. Transactions On Systems, Man, and Cybernetics. IEEE, vol. 23, nº 01, pp 277-285, 1993.

[Hassan91] Hassan, M. A. M., Malik, O. P., and Hope, G. S. A Fuzzy Logic Based Stabilizer for a Synchronous Machine, IEEE Trans. on Energy Conversion, pp. 407-413, sept. 1991.

[IK88] Interface-Karte 4.0. in Positionier + Koordinatensteuerungen. Isert-Electronic. März., 1988.

[Jamshidi93] Jamshidi, Mohammad, Vadiie, Nader, Ross, Timothy J. Fuzzy Logic and Control. Software and Hardware Applications. University of New Mexico, 1993.

[Kandel86] Kandel, Abraham. Fuzzy Mathematical Techniques With Applications. Florida State University, 1986.

[Kaufmann75] Kaufmann, A. Introduction to the Theory of fuzzy subsets, Vol. 1, Academic Press, New York, 1975.

[Kickert76] Kickert, W. J. M., and Lemke H. R. Van Nauta. Application of Fuzzy Controller in a Warm Water Plant. Automática, vol. 12, pp. 301-308, Pergamon Press, Printed in Britain, 1976.

[Kickert78] Kickert, Walter J. M. Fuzzy Theories on Decision-making. Printed in the Netherlands, 1978.

[Klein91] Klein, Rolf Dieter., Schinner, Alexander Dipl. Ing. mc-Fuzzylab r1.1. Munich Germany, 1991.

[Kosko92] Kosko, Bart. Neural Networks and Fuzzy Systems A Dinamical Systems Approach to Machine Inteligence. University of Southern California. Prentice-Hall International-Inc., 1992.

[Kosow72] Kosow, Irving I. Máquinas Elétricas e Transformadores. Editora Globo, 1972.

[Kuo82] Kuo, Benjamin C. Sistemas de Controle Automático. Prentice Hall do Brasil, 1982.

[Lee90] Lee, Chuen Chien. Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I. I.E.E.E. Manuscript received May 27, 1988; revised July 1, 1989. This work was supported in part by NASA grant NCC-2-275 and AFOSR Grant 89-0084, 1990.

[Lee90] Lee, C. C. Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Parte II. IEEE, mar./apr., 1990.

[Li88] Li, Y. F. and Lau, C. C. Development of Fuzzy Algorithms for Servo Systems. I.E.E.E. International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988.

[López94] López, José R. A., Gutierrez, Elena A., Rosa, Leonardo C. A Fuzzy Logic Based Approach to Machine-Tool Control Optimization. By CRC Press, Inc, pp.523-539, 1994.

[Madrid90] Madrid, Marconi Kolm. Controle de Tráfego de Elevadores com Inteligência Artificial Aplicando Lógica fuzzy. Primeiro SAI - Simpósio de Automação Integrada - CEFET-PR, 16 a 20 de julho, 1990.

[**Magalhães86**] Magalhães, Maurício Ferreira. Software Para Tempo Real. Campinas: Editora da Unicamp, S.P. pp 2-5, 1986.

[**Mamdani74**] Mamdani, B.E. Application of fuzzy algorithms for control of simple dynamic plant. PROC. IEE, vol.121, nº 12, december, 1974.

[**Nakamiti94**] Nakamiti, G., Freitas, R., Prado, J., Gomide, F. Inteligência Artificial Distribuída Nebulosa. Décimo Congresso Brasileiro de Automática. Sexto Congresso Latino Americano de Controle Automático. Rio de Janeiro, 19 a 23 de setembro, 1994.

[**Ogata93**] Ogata, Katsuhito. Engenharia de Controle Moderno. Segunda Edição. Prentice-Hall do Brasil, 1993.

[**Palm93**] Palm, R., Hellendoorn, H., Reinfrank, M. Fuzzy-Control für Echtzeitanwendungen: Einführung und Übersicht. pp. 107-129, 1993.

[**Runkler94**] Runkler, Thomas, Kothe, Saman H. R., Glesner, Manfred. Volle Pulle im Rückwärtsgang. Fahrzeugmanövrierhilfe mit Fuzzy-Logik. Elektronik 15, 1994.

[**Salant90**] Salant, M., A. Introdução à Robótica. MAKron Books, 1990.

[**Serafin79**] Serafin, Donald, Buncke, Harry J. Microsurgical Composite Tissue Transplantation. London, 1979.

[**Schrader66**] Schrader, F. G., Morris, J. L., Leachs, J. L. Processos de Fabricação e Materiais para Engenheiros. Programa de Publicações Didáticas. Agência Norte Americana para o Desenvolvimento Internacional - USAID, Rio de Janeiro, 1966.

[**Sibigtroth92**] Sibigtroth, James M. Implementing Fuzzy Expert Rules in Hardware. AI Expert, April, 1992.

[**Silva95**] Silva, D. D., Lücke, H. A. H., Barczak, C. L. Um Seguidor Baseado em um Controlador Difuso Dedicado. II Simpósio Brasileiro de Automação Inteligente. Curitiba, setembro, 1995.

[**Slotine91**] Slotine, J. E. and W. Li. Applied Nonlinear Control, New Jersey: Prentice Hall, 1991.

[**SUCES85**] Sociedade dos Usuários de Computadores e Equipamentos Subsidiários, Dicionário de Informática, Livros Técnicos e Científicos Editora S.A., 1985.

[**Sugeno85**] Sugeno, Michio. An Introductory Survey of Fuzzy Control Information Sciences vol.36, pp. 59-83, 1985.

[**Tilli93**] Tilli, Thomas. Fuzzy-Shell Für Windows version 1.2, 1993.

[**Togai90**] Togai, Maski Ph.D, chairman and CEO, Togai Infralogic. An Example of fuzzy logic control. Computer Design, March, 1, pp 93-97, 1990.

[**Togai91**] Togai Infralogic, Inc. Artificial Intelligence On A Chip. Document ID: TIL-READ-01E. Last Revised: Tuesday, September 10, 1991. Revision 1.2, 1991.

[Tomsovic92] Tomsovic, K. A Fuzzy Linear Programming Approach to the Reactive Power/Voltage Control Problem, IEEE Trans. on Power Systems, pp. 287-293, feb. 1992.

[Ursino89] Ursino, Mauro, Giammarco, Patrizia di, and Belardinelli, Enzo. A Mathematical Model of Cerebral Blood Flow Chemical Regulation. IEEE, Transactions on Biomedical Engineering, vol. 36, n.2, pp. 183 - 201, february 1989.

[Viot93] Viot, Greg. Fuzzy Logic in C. Dr. Dobb's Journal, February, pp 40-49, 1993.

[Wang92] Wang, Zhenyuan and Klir George J. Fuzzy Measure Theory. Plenum Press. New York and London, 1992.

[Yager94] Yager, Ronald R. On The Theory Of Approximate Reasoning. SBA Controle & Automação/vol. 4 n^o 3/setembro-outubro, 1994.

[Zadeh65] Zadeh, L. A. Fuzzy Sets. Control, n^o 3, pp. 338-354, 1965.

[Zadeh92] Zadeh, L. A. Interpolative Reasoning in Fuzzy Logic and Neural Network Theory. IEEE, First Int. Conf. on Fuzzy Systems, San Diego, CA., March 1992.

[Zimmermann85] Zimmermann, H.. Fuzzy Set Theory and Its Applications. Kluwer-Nijhoff Publishing-USA, 1985.

[Zimmermann93] Zimmermann, Hans - Jürgen. Fuzzy Logic. Band 2 - Anwendungen, 1993.

APÊNDICE A

O PROCESSADOR DE LÓGICA DIFUSA FC110

A.1 - Introdução

As vantagens que a lógica difusa apresenta, ganhou o interesse de vários especialistas e empresas no mundo todo. Devido as perspectivas positivas no desenvolvimento da técnica difusa, não é surpresa que várias companhias estão oferecendo processadores difusos dedicados. As mais proeminentes são a Togai Infralogic (Irvine, CA) e Omron (Kyoto, Japão). [Klein91]

A.2 - O Processador FC110

A Togai Infralogic está comercializando o FC110, processador de lógica difusa, reunindo em sua arquitetura tecnologia CMOS RISC a 10Mips, com um conjunto de instruções especializadas para avaliação de base de regras difusa.

O FC110 DFP (Digital Fuzzy Processor), possui as seguintes características, conforme descrito no manual da Togai Infralogic:

- . Avaliação eficiente de bases de conhecimento;
- . Suporte de programação de alto nível;
- . 100.000 avaliações de regras por segundo;
- . 60.000 saídas determinísticas por segundo 100% avaliadas;
- . Diretamente suporta mais de 800 regras;

- . Até 256 argumentos IF e THEN;
- . Arquitetura de microprocessador e conjunto de instruções;
- . Faz interface diretamente a microprocessadores comuns;
- . 64 bytes de RAM ,interna, dividida com o equipamento anfitrião;
- . 256 bytes de memória mapeadas I/O externa;
- . 128K bytes de espaço de endereço da base de conhecimento;
- . Tecnologia CMOS de baixa potência;
- . Arquitetura completa Harvard;

O processador fuzzy digital é um processador de alta performance VLSI desenhado para aplicações específicas de lógica difusa em tempo real. Sua configuração de único chip o faz pequeno o suficiente para aplicações específicas sensíveis, enquanto sua grande base externa de conhecimento o faz poderoso o suficiente para complexos sistemas controlados com lógica difusa.

A arquitetura do FC110 é otimizada para avaliação de base de conhecimento da lógica difusa. Sua capacidade de interface permitem ao usuário explorar todo o espectro de possíveis arquiteturas de sistema, desde a operação como um processador solitário até o uso como um co-processador.

O FC110 suporta bases de conhecimento com até 128K bytes de informação de funções de pertinência ou de regras. O formato das regras difusas permite um número variável de afirmações IF e THEN. As funções de pertinência são também flexíveis, tendo até 256 elementos na função de pertinência e 8 bits de resolução.

O processador é acomodado em uma embalagem plástica de 68 pinos portadora do chip, usando 56 pinos de sinal e 12 pinos de força e aterramento. A figura A.1 apresenta um sistema difuso com a utilização do processador FC110.

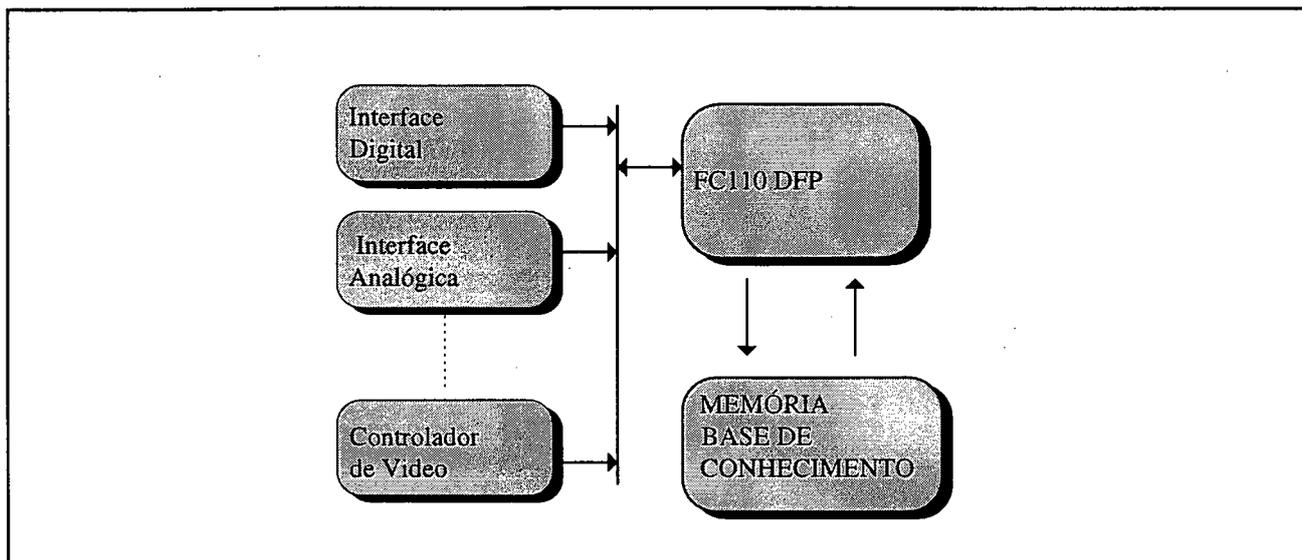


Figura A.1 - Arquitetura Básica com o FC110.

O sistema apresentado na figura A.1, representa a arquitetura básica de um sistema difuso, com um processador específico para avaliar lógica difusa. Entretanto, há outras arquiteturas que devido ao tipo de aplicação diferenciam-se da arquitetura apresentada, contudo o modelo básico é visível mesmo nas aplicações mais complexas.

Assegur, são apresentadas outras arquiteturas com o FC110. Note que nas figuras A.1, A.2, A.3, e A.4, são mostrados, respectivamente, o modelo básico, a arquitetura de um microcontrolador baseado no FC110, a arquitetura de um microcomputador baseado no FC110 e a arquitetura de multiprocessamento utilizando um número “n” de processadores difusos.

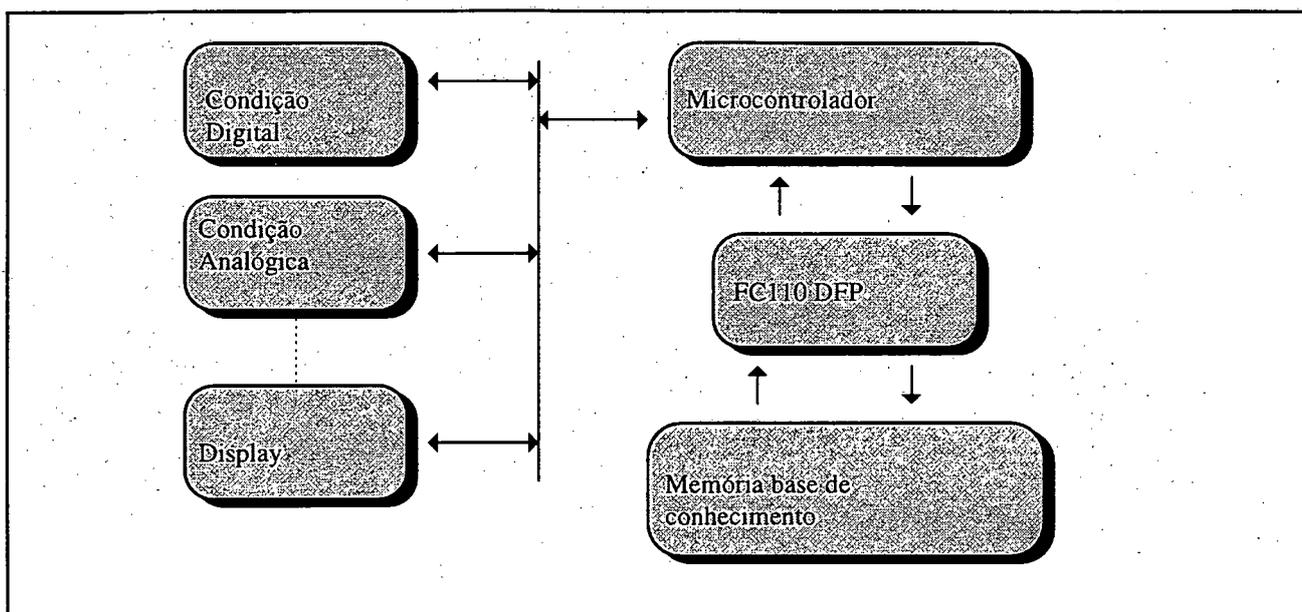


Figura A.2 - Microcontrolador baseado no FC110.

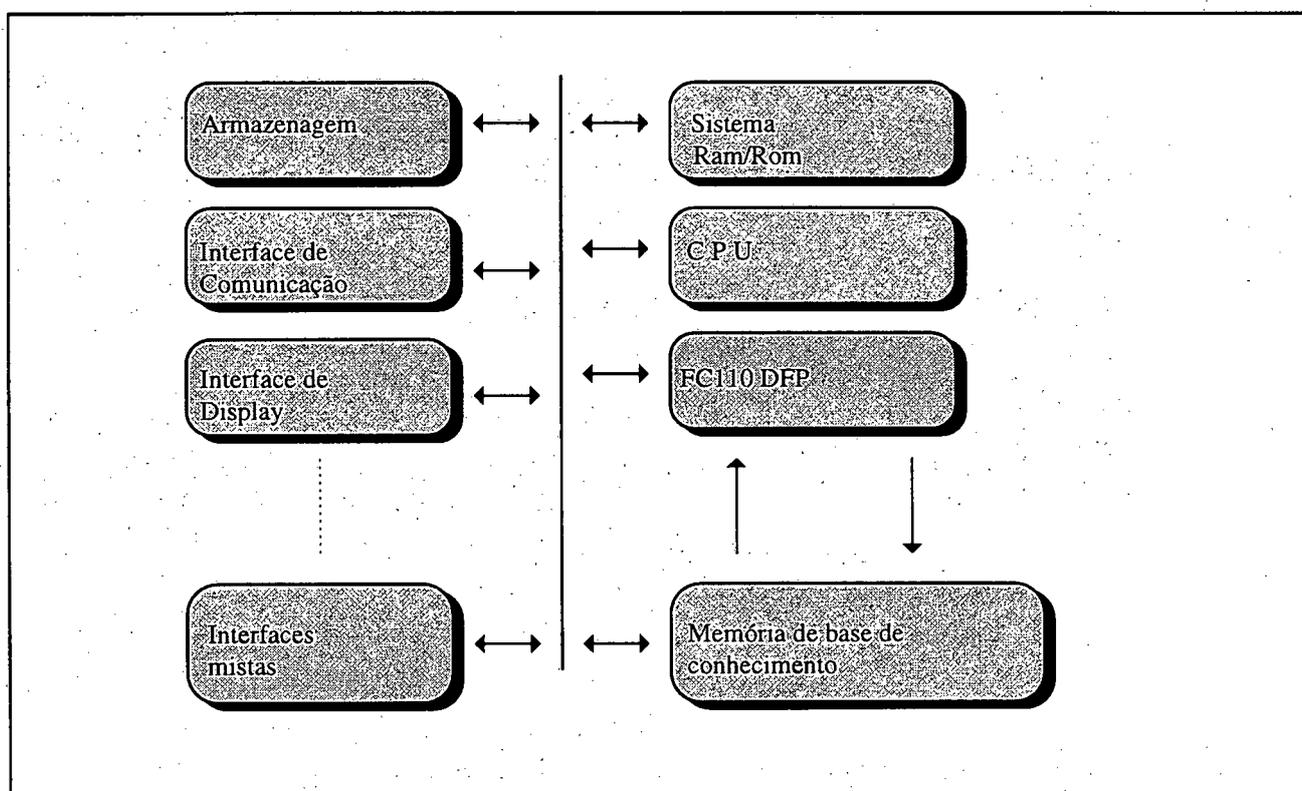


Figura A.3 - Arquitetura de microcomputador baseado no FC110.

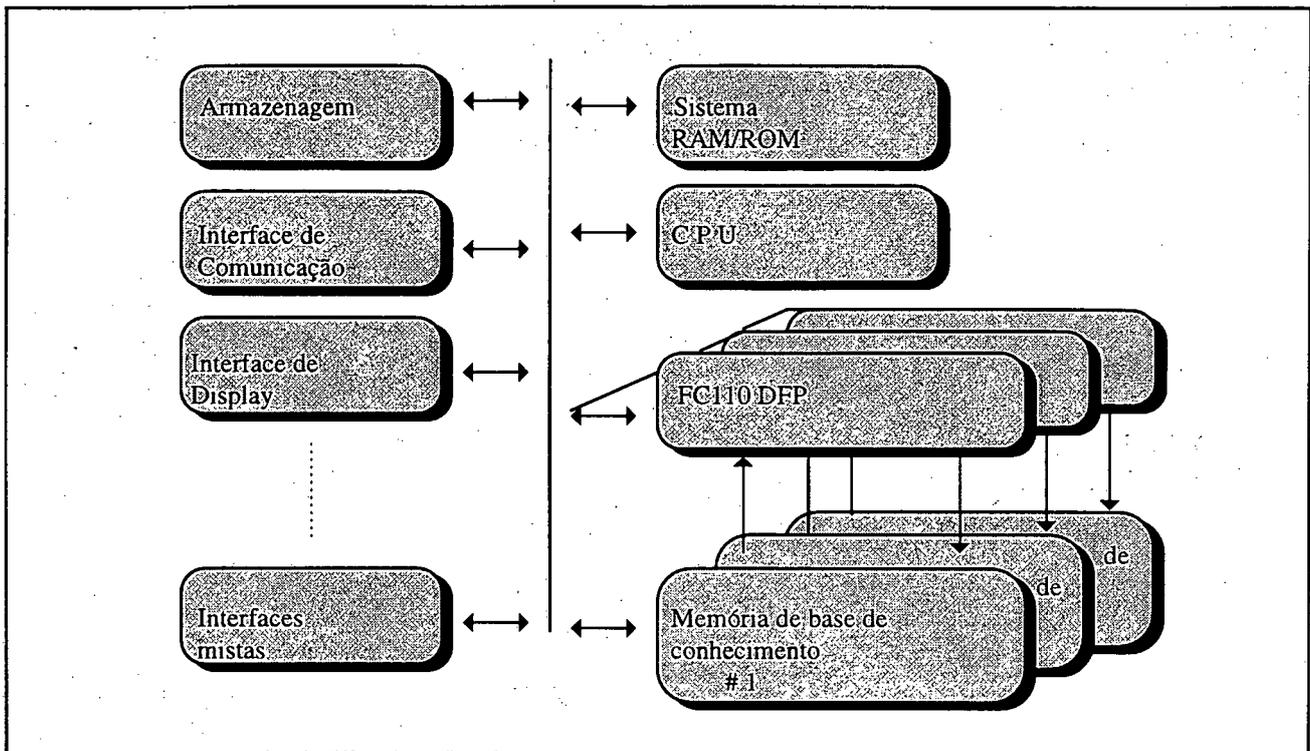


Figura A.4 - Arquitetura de multiprocessamento com o FC110.

A.2.1 - A Interface de Memória de base de Conhecimento

A base de regras difusas e todas as instruções usadas para controlar o FC110 residem na memória base de conhecimento. A memória base de conhecimento pode ser tanto ROM, EPROM ou RAM. A escolha do tipo de memória é inteiramente dependente do tipo de sistema em que o FC110 é especificado.

A interface base de conhecimento com o FC110 consiste em duas vias ("buses"), a via de endereço da base de conhecimento de 16 bits e a via de dados também de 16 bits.

A.2.2 - A Interface Para Sistemas Baseados no FC110

Um sistema geral é o mostrado na figura 1.1 (capítulo 1). O processador de sistema comunica-se com o FC110 através de uma via de interface de sistema de 20 pinos. Seis desses pinos são parte da via de endereço de sistema. Os outros oito pinos pertencem a via de dados de sistema. Os seis pinos restantes realizam vários controles e funções.

A.2.3 - A RAM de Dados Partilhados Interna

Dados e comandos são passados entre o sistema e o FC110 por meio de uma RAM, interna ao processador, de dados partilhados. A RAM de dados partilhados é mostrada no diagrama de blocos do FC110 (figura A.5).

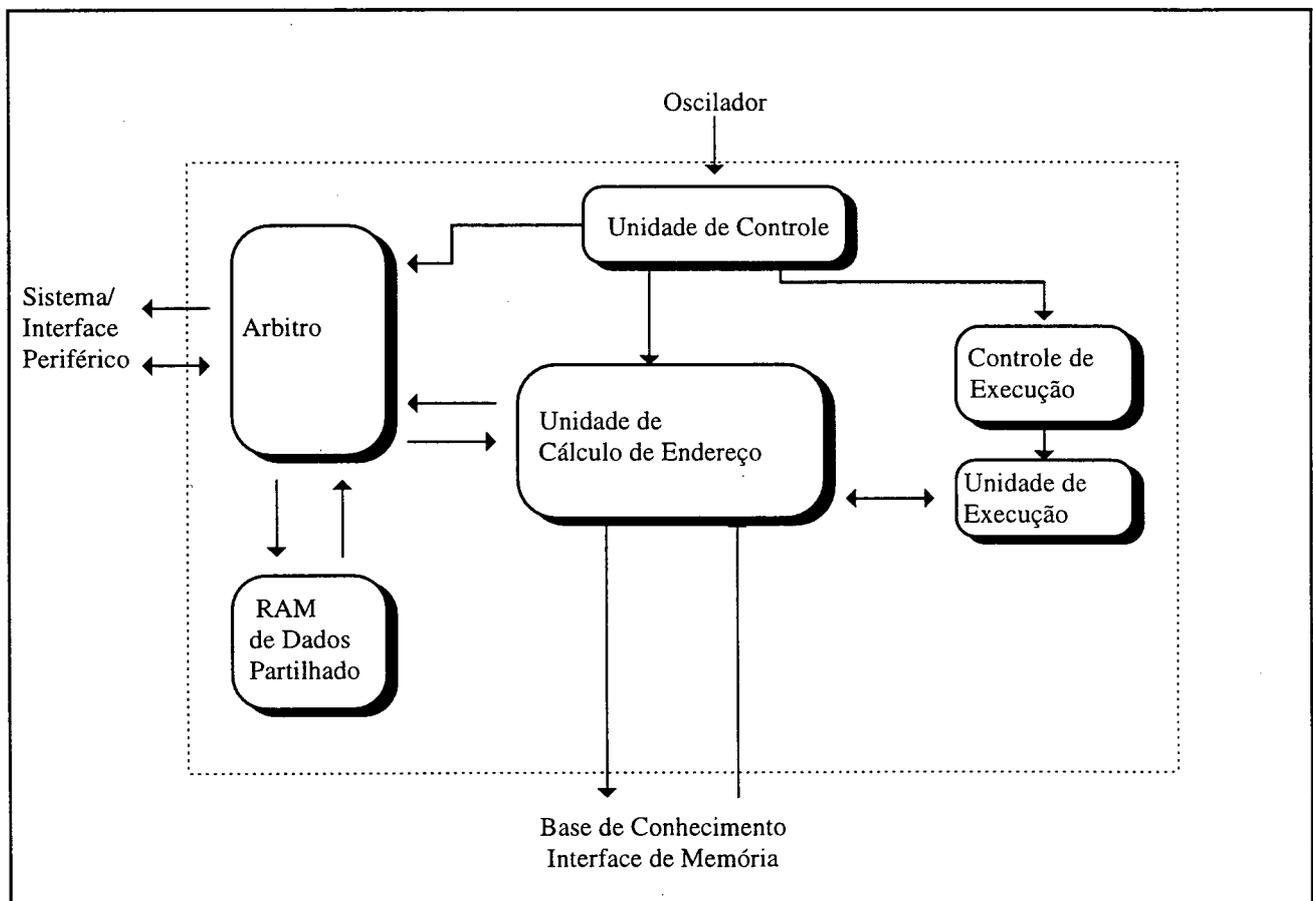


Figura A.5 - Diagrama de Blocos do FC110. [Klein91]

A RAM é organizada como uma memória de leitura/escrita de 256 bytes. O FC110, por si mesmo, tem acesso a todos os 256 bytes da RAM de dados partilhados. Estas 256 locações de memória são geralmente usadas para armazenamento de cálculos imediatos e a pilha de programa.

O FC110 usa um esquema arbitrário interno para assegurar a integridade dos dados sendo transferidos para e da RAM de dados partilhados. O esquema arbitrário permite ao primeiro pedido, tanto pelo sistema quanto pelo FC110, ser atendido primeiro, e faz qualquer pedido posterior esperar até que o primeiro seja acabado. Se pedidos simultâneos são feitos, a prioridade é dada ao FC110.

A.2.4 - A Interface Periférica

O FC110 não pode apenas aceitar transferências de dados de sistema como um “bus” escravo, mas pode também iniciar transferências de dados de sistema como um “bus” mestre, via um ciclo de acesso periférico. Os primeiros 256 bytes de espaço de endereço da interface de sistema, como visto pelo FC110, começando em 8000 hexa, são usados para acesso periférico pelo FC110. Armazenar dados nestas locações inicia um ciclo de escrita periférico, isto significa que dados FC110 vão ser armazenados fora do processador (“off-chip”). Carregar dados destas locações inicia um ciclo de leitura periférico. Um ciclo de leitura periférico vai transferir dados de fora do processador para o próprio FC110.

O mapa de memória para a RAM de dados partilhados do ponto de vista do FC110 e da interface do sistema é mostrado na figura A.6. Nesta figura, Mux e Nmux significam, respectivamente, modo multiplexado e modo não multiplexado para a interface do sistema.

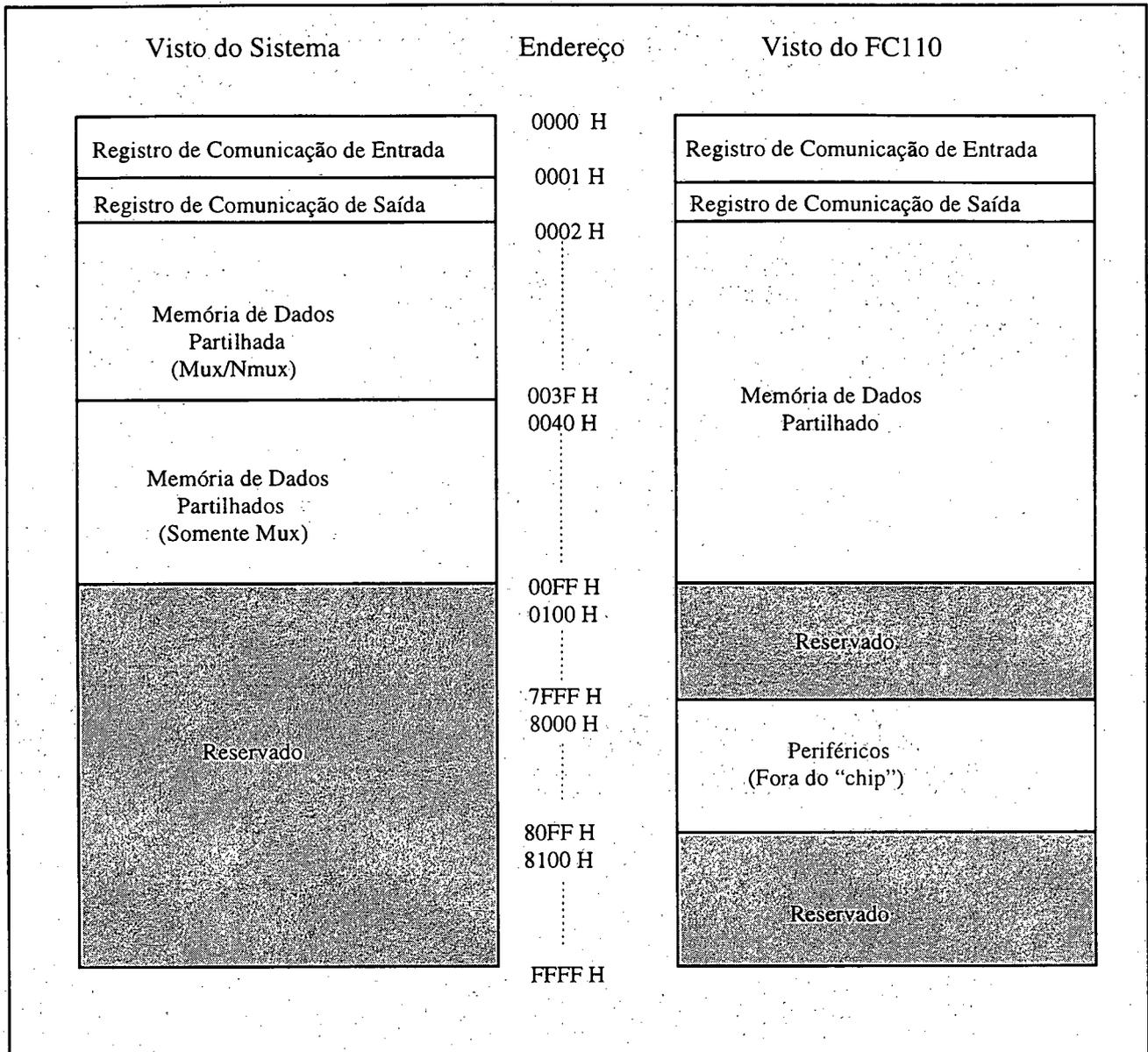


Figura A.6 - Mapeamento de Endereços da Interface do Sistema e Periféricos. Nota: Acesso para os endereços reservados podem causar resultados indefinidos [Klein91].

A.2.5 - Resumo do Conjunto de Instruções do FC110

O FC110 tem muitas das mesmas instruções de microprocessadores tradicionais. Entretanto, para rapidamente avaliar bases de regra da lógica difusa, várias instruções especializadas foram adicionadas. Um resumo do conjunto de instruções é dado na tabela A.1 a seguir.

Mnemônico	Descrição	Mnemônico	Descrição
ADD	Adição aritmética com vai um	MULT	Multiplicação Aritmética
CMP	Comparação Aritmética	NOP	Sem Operação
DEFUZ	Desfuzzyficação	RHSC	Avalia o Lado Direito (porção "THEN") de uma regra usando o método do centróide
DIV	Divisão Aritmética	RHSH	Idem RHSC, usando o Método da altura
FZAND	Conectivo "AND" Difuso	RTS	Retorna da Subrotina
FZOR	Conectivo "OR" Difuso	STORE	Armazena para registro ou Memória
HALT	Para e espera por uma Mensagem do Sistema	SUB	Subtração Aritmética com Empréstimo
JMP	Desvia Para Novo Alvo		
JSR	Desvia Para Subrotina		
LHS	Avalia a Parte Esquerda (Porção "IF") de uma Regra		
LOAD	Carrega um Registrador ou Memória		
MODE	Define a Operação e modo funcional do FC110		

Tabela A.1 - Resumo das Instruções do FC110. [Klein91]

O FC110 tem seis instruções explicitamente para avaliação de regra em lógica difusa. A instrução "LHS" é usada para avaliar o lado esquerdo (porção "IF") de uma regra. Esta instrução

pode diretamente suportar até 256 argumentos. A instrução “RHSC” é usada para avaliar o lado direito (porção “THEN”) de uma regra, usando o método centróide de desfuzzyficação. A instrução “RSHS” também avalia o lado direito de uma regra, mas usa o método da altura de desfuzzyficação. As instruções “FZAND” e “FZOR” são usadas para criar regras complexas com até mais argumentos “IF”, ou conectivos “AND” e “OR”. A instrução “DEFUZ” realiza a operação final de desfuzzyficação nos cálculos do lado direito (RHSC e RSHS).

A.2.6 - Modos de Endereçamento do FC110

Muitas das instruções listadas na tabela A.1, operam em múltiplos modos de endereçamento. O FC110 suporta seis tipos diferentes de modos de endereçamento. Os modos de endereçamento estão resumidos abaixo.

1- Modo de Endereçamento Imediato

No modo de endereçamento imediato, o operando de memória é contido na palavra da instrução.

2 - Modo de Endereçamento Direto - Memória

Neste modo de endereçamento, o endereço do operando de memória é contido na palavra imediatamente seguindo a palavra de instrução corrente.

3 - Modo Direto - Registrador

No modo de endereçamento direto em registrador o operando é encontrado na fonte e/ou registros de destino que residem no FC110.

4 - Modo Indexado Direto - Memória

Neste modo de endereçamento, o endereço do operando de memória é encontrado pela adição dos conteúdos de um registrador de índice ao endereço base do operando. O endereço base do operando é contido na palavra imediatamente seguindo a palavra de instrução corrente.

5 - Modo Pré-Incremento Indireto de Memória de Dados

No modo de endereçamento indireto pré-incremento de memória de dados, o endereço de memória do operando é encontrado incrementando um registrador fonte e usando seus conteúdos como o endereço de memória do operando.

6 - Modo de Endereçamento Pós-Decrementado Indireto de Memória de Dados

Neste modo de endereçamento, o endereço de memória do operando é encontrado em um registrador destino. O registrador destino é então decrementado.

Nota:

Uma análise mais completa, sobre os modos de endereçamento do FC110, o conjunto de instruções, e como programar o processador, encontra-se no Guia do Conjunto de Instruções do FC110.

A.2.7 - Metodologia de Transferência de Dados do Sistema

Esta parte, descreve o método sugerido para coordenar a operação do sistema com aquele do FC110 e para a transferência de dados entre o FC110 e o sistema anfitrião.

O sistema anfitrião pode ler e escrever na RAM de dados partilhados a qualquer hora, como permitido pelo árbitro a bordo do FC110. É desejável, entretanto, coordenar a transferência de dados, tal qual observações medidas (dados de entrada de regra) e conclusões de regras, com a avaliação da base de regra, ou bases de regra, armazenadas na memória base de conhecimento. O

FC110 tem vários mecanismos embutidos para a coordenação de transferência de dados, comandos e “status” entre o sistema e ele mesmo. A RAM de dados compartilhados contém duas locações especiais de endereços para facilitar a passagem de mensagens e comandos. Estas locações são também amarradas a sinais de controle para proposta de coordenação de comunicações.

A primeira destas locações é o registrador de entrada de comunicação (ICR) que está localizado no endereço 00H. Esta locação é usada para mandar mensagens do sistema para o FC110. Quando o sistema escreve uma mensagem para o ICR, o “flag” de comunicação pendente do FC110 (“P”) é ligado. Isto notifica o FC110 que uma mensagem está pendente. Se o FC110 está parado, via execução de uma instrução “HALT”, subindo a bandeira P vai causar um “re-start” do FC110 na locação de base de conhecimento imediatamente seguinte àquela da instrução “HALT”.

A segunda locação é o registrador de comunicação de saída (OCR), localizado no endereço 01H. Esta locação é usada para mandar mensagens do processador FC110 para o sistema anfitrião. Quando o processador difuso escreve para este endereço, o sinal externo “/INT” (interrupção) ao sistema anfitrião é ativado alertando o sistema para a mensagem pendente. Uma vez que o sistema leu o OCR, o sinal /INT retorna ao seu estado inativo.

O mecanismo final para a coordenação de comunicações é o sinal “/IDLE”. Este sinal externo é ativado quando o FC110 é parado via uma instrução “HALT” e é usado para notificar o sistema anfitrião que a execução de base de regra foi completada, e que o FC110 está esperando por entrada posterior. Uma vez que os dados são escritos pelo sistema anfitrião no ICR, o FC110 recomeça e o sinal /IDLE fica inativo. O método sugerido para a transferência de dados entre o FC110 e o sistema anfitrião é composto do pelos seguintes passos básicos:

Passo 1: Com o FC110 parado, o sistema anfitrião armazena dados de observação na RAM de dados compartilhados.

Passo 2: O sistema anfitrião escreve no ICR, causando a colocação do flag “P” pelo FC110 começando a execução da base de conhecimento e desativando /IDLE.

Passo 3: O FC110 (executando a base de conhecimento) lê o registrador ICR, ressetando o flag “P”.

Passo 4: O FC110 avalia regras, armazena os valores da conclusão na RAM de dados compartilhados.

Passo 5: O FC110 escreve no OCR, ativando “/INT”.

Passo 6: O FC110 lê o ICR para assegurar que o flag “P” esteja ressetado antecedendo a pausa (“halting”).

Passo 7: O FC110 executa a instrução “HALT”, ativa “/IDLE”, e para a execução.

Passo 8: O sistema anfitrião lê o “OCR” que faz com que o FC110 desative “/INT”.

Passo 9: O sistema anfitrião lê os valores da conclusão da RAM de dados compartilhados e retorna ao passo 1.

Para usar o FC110 DFP apropriadamente, é importante ter uma visão geral do processo inteiro de desenvolvimento de programas do FC110, da captura da base de conhecimento difusa até a verificação final da base de conhecimento do FC110.[03]

É recomendável, para maiores informações sobre o processador FC110, se referir ao manual de instruções da Togai Infralogic.

APÊNDICE B

A Placa mc-Fuzzylab

B1 - Introdução

Nesta parte apresenta-se algumas das características da placa mc-Fuzzylab utilizada neste trabalho. Coloca-se alguns aspectos sobre a configuração do “hardware” e sobre alguns “softwares” da placa.

B2 - O “Hardware”

O mc-Fuzzylab é uma placa de programação baseada no FC110 da Empresa Togai, que pode ser instalada em um computador tipo PC-AT. O mc-Fuzzylab pode, ainda, trabalhar como aparelho totalmente independente, sem o PC, quando a base de conhecimento for instalada em EPROM e não em RAM.

O processador FC110 forma o coração da placa e nela se encontram dois transputer-link. Com isso é possível instalar outras placas em rede.

A técnica difusa oferece, no caso dos problemas que não sejam exatos de se formular, os quais se sabe mais ou menos o que se tem que fazer, uma interessante possibilidade de solução. Como as regras difusas podem ser rapidamente analisadas, o mc-Fuzzylab pode ser utilizado para trabalhar em tempo real.

O mc-Fuzzylab está disposto numa placa projetada para ser conectada ao PC. A comunicação se realiza através de circuitos periféricos e uma área de memória que pode ser lida ou escrita.

Um conversor A/D com 8 canais e 8 bits por canal é utilizado na constituição física da placa. A taxa de conversão é de $2,5\text{v}/\mu\text{s}$ por canal. Um conversor D/A, também impregado na placa, com 4 canais opera igualmente com 8 bits e com a mesma taxa de conversão.

quais não podem ser descritas com sistema de regras. Um disassemblador também é parte integrante dos programas disponíveis da placa..

Ainda, com a placa mc-Fuzzylab, estão dispostos outros “softwares”. Para que possa-se testar o mc-Fuzzylab, apresenta-se um programa(DIAG110) que faz diagnóstico. Com este programa, as placas são procuradas e quando encontradas, suas memórias são testadas.

Outro “software” disponível é o DEBUG110. Este “software” permite que os estados periféricos do mc-Fuzzylab sejam observados no monitor, numa representação gráfica e permite ainda que sejam exercidas influências das entradas.

Para ser possível carregar programas “fuzzy” diretamente de programas próprios, os “softwares” DRVR110.C e DRVR110.h possibilitam não só fazer o carregamento da base de conhecimento mas, também, exercer a comunicação com o mc-Fuzzylab e o programa principal.

Para melhores detalhes, sobre o “hardware” ou “softwares” referir-se ao manual da placa **[Klein91]**.

APÊNDICE C

PROCEDIMENTO PARA O CONTROLE DO MECANISMO SEGUIDOR

C1 - Introdução

Neste apêndice são colocados os programas em linguagem c, feitos para controlar o seguidor difuso. Primeiramente, é apresentado o código referente ao controle com o processador FC110, denominado **Segue110.c**, que faz a comunicação dos dados provenientes do processamento difuso para o programa de controle **Seguidor.c**. Este código é o responsável pela comunicação entre o computador PC-AT e o meio externo.

C2 - Segue110.c

```
/*
 * file segue110.c
 * created by togai infralogic (r) fc110 linker version 2.3.0
 * tue nov 28 23:59:43 1995
 */

#include <stdio.h>
#include <stdlib.h>
#include <tilcomp.h>
#include <drvrl110.h>
#include <conio.h>
#include "segue110.h"

globals

int segue110_init (char *hexfilename, int processor)
{
    set_globals(processor)
    int fc110_init (char* ,int);
    return fc110_init (hexfilename, processor);
}
```

```
void seguidor_difuso (ubyte nivel, ubyte *posicao_x)
{
    locals

    init

    wait

    store(2, nivel);
    store(3, *posicao_x);

    fire(0);

    wait

    *posicao_x = (ubyte) load(3);

    cleanup
}
```

C3 - O Seguidor.c

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <drv110.h>
#include <tilcomp.h>
#include "segue110.h"
#include "8250.h"
#include "interface.h"
#ifndef __math_h
#include "math.h"
#endif
#ifndef __process_h
#include "process.h"
#endif
#ifndef __conio_h
#include "conio.h"
#endif
#ifndef __string_h
#include "string.h"
#endif
```

```

/*-----
   procedimentos de interface com maquina - nivel 0
   -----*/

#define velxy 6000 //configuracao.velxypu
#define veldz 21 //configuracao.veldz
#define velsz 21 //configuracao.velsz

struct configuracao {long int x0,y0,penup,pendown;
                    long int velsz,veldz,velxypd,velxypu;
                    long int lim_min_x,lim_min_y,lim_max_x,lim_max_y;} configuracao;

long int _n0_x, // valor atual do eixo x
         _n0_y, // valor atual do eixo y
         _n0_z, // valor atual do eixo z
         velx, // valor da velocidade do eixo x
         vely; // valor da velocidade do eixo y
long int lmx=15000, // limite maximo do eixo x
         lmix=0, // limite minimo do eixo x
         lmy=32000, // limite maximo do eixo y
         lmiy=0; // limite minimo do eixo y

/*
void espera(int tempo)
{
    int i,j;
    for(i=0;i<=tempo;i++)
        for(j=0;j<=3000;j++);
}
*/

// rotina para envio do comando pela porta serial
void command(char *l){
    char r;
    while(!data_ready(com1)); // espera pela confirmacao do comando anterior
    r=receive(com1);
    if(r!='0'){
        printf("\nerro na comunicacao - %c\n",r);
        exit(2);
    }
    while(*l){ // envia o proximo comando
        while(!thr_empty(com1));
        transmit(com1,*l);
        l++;
    }
}

```

```

}
}

```

// rotina para inicializar a porta serial e resetar a maquina

```

void inicializahardware(void){
    char *l;
    set_baud_rate(com1,_4800);
    set_parity_off(com1);
    set_1stop_bit(com1);
    set_char_length(com1,8);
    set_break_off(com1);
    disable_interrupt(com1);
    receive(com2);
    _n0_x = configuracao.x0;
    _n0_y = configuracao.y0;
    _n0_z = 0;
    velx=vely=configuracao.velxypu;
    l = "@07\r"; // utilizar os 3 eixos
    while(*l){
        while(!thr_empty(com1));
        transmit(com1,*l);
        l++;
    }
    command("@0r7\r"); // reseta a maquina
}

```

/*-- monta o comando de pen absolute respeitando os limites da maquina, para
entao manda-lo para a maquina -----*/

```

void _penabsolute(long int x ,long int y){
    char str[150],s2[150];

```

```

    _n0_x = x; _n0_y=y;

```

```

if((x<=lmx)&&(x>=lmix)&&(y<=lmy)&&(y>=lmy)&&(velx<=10000)&&(vely<=10000)&&(velx
>0)&&(vely>0)

```

```

    &&(_n0_z>=0)&&(_n0_z<=9000)&&(veldz<=10000)){

```

```

    strcpy(str,"@0m ");

```

```

sprintf(s2,"%ld,%ld,%ld,%ld,%ld,%ld,%ld,%ld\r",_n0_x,velx,_n0_y,vely,_n0_z,(long)21,(long)0,(l
ong)21);

```

```

    strcat(str,s2);

```

```

    command(str);

```

```

    }
    else{
        errosimples("posicao fora de limite");
        return;
    }
}

```

```

void main()
{
    char * msg1 = "o nivel de voltagem no canal ain0: ";
    char * msg2 = "faixa de voltagem baixa, aumentar pos_x e manter pos_y.";
    char * msg3 = "faixa de voltagem adequada, manter posicao do seguidor.";
    char * msg4 = "faixa de voltagem alta, diminuir pos_x e manter pos_y.";

    file *f1;
    file *f2;

    int coluna = 0;

    int ultimo_estado = 0; // 0 = nada, 1 = baixo, 2 = adequado, 3 = alta
    ubyte nivel, posicao_x, posicao_y;
    unsigned int x,y,xmaq,ymaq,zmaq;
    // char *led;
    unsigned char far *fuzzypr = ((char far*)0xdc000001);
    // float tabela[2] = { 1.0, 2.0 };

    {
        globals
        set_globals(0)

        if (segue110_init("segue110.hex",0)!=0)
        {
            printf("falha na inicializacao da placa fc110.");
            return;
        }

        nivel = 0; posicao_x = 0;
        outp( 0x300, 0xc0 );
        outp( 0x300, 0xc6 );
        pokeb(0xdc00,0x960,0x55);
        clrscr(); // limpa a tela

        // abre o arquivo (cria se nao existir)

```

```

f1=fopen("arq1.dat","wb");
f2=fopen("arq2.dat","wb");

// imprime cabecalhos nos arquivos
fprintf(f1, "\\t\\t\\tarquivo de voltagens\\r\\n\\n");
fprintf(f2, "\\t\\t\\tarquivo de mensagens\\r\\n\\n");

printf(msg1);
fprintf(f1,"%s\\r\\n", msg1);

x = wherex();
y = wherey();

configuracao.x0 = 0;
configuracao.y0 = 0;
configuracao.penup = 0;
configuracao.pendown = 0;
configuracao.velsz = 1000;
configuracao.veldz = 1000;
configuracao.velxypd = 7000;
configuracao.velxypu = 7000;
configuracao.lim_min_x = 0;
configuracao.lim_min_y = 0;
configuracao.lim_max_x = 20000;
configuracao.lim_max_y = 20000;

//resetar a maquina
inicializahardware();

while (!kbhit())
{
float volt;
// delay(500);
nivel = (ubyte)load(0x120); // carrega em nivel o byte de 0x920
volt = (nivel/5.0)/10.0;
gotoxy(x,y);
printf("%8.6f", volt);

// grava no arquivo
fprintf(f1,"%10.6f", volt);

if( (coluna += 10) > 70 )
{
coluna = 0;

```

```

    fprintf(f1, "\r\n");
}

printf(" volts \n" );
creol(); // limpa linha
// seguidor_difuso( nivel, &posicao_x, &posicao_y);
seguidor_difuso(nivel, &posicao_x);

// printf("posicao_x = %i posicao_y = %i nivel = %i\n", posicao_x, posicao_y, nivel);
printf("posicao_x = %i nivel = %i\n", posicao_x, nivel);

// nivel baixo
// if(volt<=2.0000)
//if(nivel<=112)
//if(posicao_x>=160)
//if(posicao_x>=32)
//if(posicao_x>=134)
if(posicao_x>144)
{
    creol();

    printf(msg2);
    if(ultimo_estado != 1)
    {
        fprintf(f2, "\r\n%s\r\n", msg2);
        ultimo_estado = 1;
    }
    pokeb(0xdc00, 0x960, 0x6);
    // delay(500);
    pokeb(0xdc00, 0x960, 0x00);
// incrementa a posicao do eixo x
//   _n0_x++;
//   _n0_x += 5;
    _n0_x += posicao_x/10;

}

else

// nivel medio
//   if(((volt>2.0000)&&(volt<2.2000)))
//   if(((nivel>112)&&(nivel<144)))
//if(((posicao_x>96)&&(posicao_x<160)))
//if((posicao_x>-32)&&(posicao_x<32))

```

```

//if(((posicao_x>122)&&(posicao_x<134)))
if(((posicao_x>=112)&&(posicao_x<=144)))

{
    clreol();

    printf(msg3);
    if(ultimo_estado != 2)
    {
        fprintf(f2, "\r\n%s\r\n", msg3);
        ultimo_estado = 2;
    }

    pokeb(0xdc00,0x960,0x18);
    // delay(500);
    pokeb(0xdc00,0x960,0x00);
// incrementa posicao do eixo y
//   _n0_y++;
   _n0_y += 10;
//   _n0_x = posicao_x;
}

else

// nivel alto
// if(volt>2.2000)
// if(nivel>=144)

//if(posicao_x<=96)
//if(posicao_x<=-32)
//if(posicao_x<=122)
//if(posicao_x>=160)
if(posicao_x<112)
{
    clreol();

    printf(msg4);
    if(ultimo_estado != 3)
    {
        fprintf(f2, "\r\n%s\r\n", msg4);
        ultimo_estado = 3;
    }

    pokeb(0xdc00,0x960,0x60);
    // delay(500);
}

```

```
    pokeb(0xdc00,0x960,0x00);
// decreta a posicao do eixo x
//    _n0_x--;
//    _n0_x -= 5;
    _n0_x -= posicao_x/10;

}
fprintf(f2, "posicao_x = %i nivel = %i\r\n", posicao_x, nivel);

// envia o comando para a maquina com os novos valores de _n0_x e _n0_y
    _penabsolute( _n0_x, _n0_y );
}

fclose(f1);
fclose(f2);
}
}
```

Os dois programas acima, constituem os programas de comunicação com a placa. O programa de controle (**Segue110.c**) e o programa de interface com o **PC-AT** para o meio externo (**Seguidor.c**).