



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



**IMPLEMENTAÇÃO DE UM CONTROLADOR
FUZZY COMO ESTABILIZADOR DE POTÊNCIA
EM MÁQUINA SÍNCRONA**

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

Henrique Simas

Florianópolis, Janeiro de 1996

Implementação de um Controlador Fuzzy como Estabilizador de Potência em Máquina Síncrona

Henrique Simas

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

Banca Examinadora:

Prof. Guilherme Bittencourt
Orientador do Curso

Eng. Álvaro Arioni Paladino
Orientador Empresa Reivax

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

Prof^a. Janette Cardoso, Avaliadora

Jean do Amaral Lima, Debatedor

Rodolfo Pinto da Luz, Debatedor

Agradecimentos

Este mesmo projeto não teria sido bem sucedido se não fosse algumas ajudas importantes, dentre elas posso destacar: Os professores do curso de engenharia de controle e automação industrial, que através de seus conhecimentos mostraram o caminho para minha formação como engenheiro.

Em especial aos professores Aguinaldo Silveira e Silva, Raul Guenther, Czeslau Barczak e Guilherme Bittencourt, professores orientadores dos trabalhos realizados nos laboratórios por terem me ensinado a ver que a engenharia é além do que é ministrado em sala de aula.

A REIVAX Automação e Controle, com todos os seus engenheiros e funcionários, por terem aberto a oportunidade de implementar um projeto de caráter de pesquisa de base, para futuros empreendimentos desta empresa.

A meus amigos, parentes, colegas e minha noiva, que de uma forma ou de outra, sempre estiveram a meu lado nos momentos mais difíceis, através de apoio e compreensão.

A meus pais e avós por terem sido minha fonte inspiradora e de exemplo para que eu seguisse a engenharia como opção de vida, e por terem sempre me apoiado de forma incondicional nos momentos de dificuldade com os estudos.

E a DEUS, esta força que rege o mundo e todas as coisas sobre ele, por ter me dado o dom de ser engenheiro e mostrado que a vida pode ser do jeito que quisermos. Basta lutar e vencer os desafios que ela nos põe a prova.

Resumo

O trabalho a seguir apresentado, tem o objetivo de mostrar as características referentes a teoria, implementação e ajustes de um controlador baseado em lógica fuzzy.

Os controladores fuzzy tem se apresentado nos últimos anos como uma linha de pesquisa promissora, dado os resultados obtidos em pesquisas recentes nesta área. Isto motivou a empresa REIVAX Automação e Controle a desenvolver uma ferramenta de estudo de controladores fuzzy como PSS(power system stabilizer), estabilizadores de potência em máquina síncrona.

Como resultado final deste trabalho, foi implementado um programa na linguagem Microsoft C, FuzSsd.C, no qual esta construído um controlador fuzzy com todas as alternativas de operações fuzzy, limites entre outras. Caracterizando-se como um controlador fuzzy didático, com um código de fácil acesso e reusável, útil em estudos em outras aplicações.

Nos capítulos subseqüentes são mostradas todas os itens estudados e utilizados na redação nos temas de cada um, demonstrando de uma forma mais clara a possível os tópicos mais importantes.

Índice

	página
Agradecimentos	III
Resumo	IV
Índice	V
Introdução	1
Capítulo I	
O Processo de Geração de Potência	4
1.1 Introdução	4
1.2 O Funcionamento da Máquina Síncrona	4
1.3 O software SSD	7
1.4 Conclusão	8
Capítulo II	
Teoria da Lógica Fuzzy	9
2.1 Introdução	9
2.2 Noções Fundamentais Sobre Conjuntos e Função de Pertinência	9
2.3 Sub-Conjuntos Fuzzy	11
2.3.1 Relação de Dominância	12
2.3.2 Operações em Sub-Conjuntos Fuzzy	13
2.3.3 Conjunto de sub-conjuntos fuzzy para para M e E finitos	16
2.3.4 Produto e Soma Algébricos de Sub-Conjuntos Fuzzy	16
2.4 Grafos Fuzzy e Relações Fuzzy	17
2.4.1 Grafos	17
2.4.2 Relações	18
2.5 Conclusão	22
Capítulo III	
Controladores Fuzzy	24
3.1 Introdução	24
3.2 Pensar Fuzzy	24
3.3 Configuração do Controlador Fuzzy	29
3.3.1 Operações de Intersecção	30
3.3.2 Operações de União	31
3.3.3 Operações de Defuzzificação	31
3.3.4 Operações de Alteração das Vizinhanças (Hedge)	36
3.4 Conclusão	37
Capítulo IV	
Metodologias de Concepção e Implementação de Controladores Fuzzy	38
4.1 Introdução	38

Índice

4.2 A Aplicabilidade de Controladores Fuzzy Frente aos Controladores Clássicos	38	
4.3 Metodologias de Implementação de Controladores Fuzzy	40	
4.3.1 Algoritmos Genéticos	40	
4.3.2 Redes Neuro-Fuzzy	40	
4.3.3 Equivalência do Controlador Fuzzy ao PI	40	
4.3.4 Controladores Fuzzy baseados na Relação de Não-Linearidades	41	
4.4 O Software FIDE	42	
4.5 Conclusão	43	
Capítulo V		
Placas de Aquisição	45	
5.1 Introdução	45	
5.2 Placa ADA 10	45	
5.3 Placa DAX	46	
5.4 Placa ADA 16	47	
5.5 Esquema de Conexão das Placas	48	
5.6 Conclusão	48	
Capítulo VI		
O Software FuzSsd	49	
6.1 Introdução	49	
6.2 Especificação do Programa	49	
6.3 Funcionamento do FuzSsd	50	
6.3.1 Utilizando o FuzSsd	53	
6.4 Conclusão	65	
Capítulo VII		
Simulações	66	
7.1 Introdução	66	
7.2 Simulações de Variação de Referência	66	
7.3 Simulações da Reatância Externa	67	
7.4 Comparação Entre Ajustes de Defuzzificação	69	
7.5 Conclusão	70	
Conclusão Geral	71	
Bibliografia	73	
Anexos		
Anexo I	Descrição das Funções Implementadas no Programa FuzSsd	75
Anexo II	Relatório de Ajustes do Controlador Fuzzy	80
Anexo III	Diagrama de Blocos da Máquina Síncrona	82

Introdução

A física e a matemática sempre estiveram juntas de modo a que nós pudéssemos entender as relações e influências dos diversos sistemas encontrados dentro de nosso mundo.

Teoremas, equações, definições, teses, hipóteses, corolários, entre outras formas de expressar o nosso entendimento do mundo, se tornaram as ferramentas palpáveis que podemos utilizar para pesquisar determinado problema, mesmo assim estas nunca foram suficientes para representação de um sistema por completo.

Com o avanço da ciência e com ela o aumento da complexidade dos sistemas, ficamos perante problemas sérios como representação matemática e física destes mesmos, pois exigem cada vez mais precisão dentro de um emaranhado de fluxos de informações e dados que interagem entre si.

Esta complexidade pode ser observada em nosso cotidiano, como o universo da computação e sua relação com o controle de sistemas automáticos. Um exemplo disto é o controle de flaps de um avião, observe quantos elementos terão que ser levados em consideração, quantas variáveis serão utilizadas e quão precisa terá que ser a ação de controle para que numa manobra do avião não ocorram problemas que acarretem acidentes.

Ainda podemos destacar a economia, uma matéria tão influente em nossas vidas, onde a modelagem matemática de sistemas dentro desta área se torna cada vez mais complexa, a medida que se adiciona em equações mais e mais fatores. Claro que o modelo matemático fica mais preciso, entretanto perde-se na sua complexidade e tempo de cálculo para se chegar em uma resposta final.

Em engenharia não poderia ser diferente, podemos observar isto em áreas de pesquisa de novas tecnologias, pois há casos que necessita-se de um profundo conhecimento do sistema dentro de uma ótica da modelagem matemática. Em certos casos, se torna uma tarefa trabalhosa, e em praticamente todos os casos existe a necessidade de se linearizar este modelo, restringindo-o a uma condição específica de funcionamento. Esta restrição leva as vezes a

sub-utilização de um equipamento, que por consequência interfere nos custos finais de um determinado projeto.

Entende-se como lógica fuzzy, uma proposta de representação de sistemas que não leva em consideração e sua modelagem matemática é a lógica fuzzy.

A lógica fuzzy foi apresentada pela primeira vez em 1960 num trabalho de Lotfi Zadeh, cujo objetivo era a tentativa da representação do pensamento humano

em uma máquina (computador). Por isso este tipo de lógica foi aplicada inicialmente em sistemas especialistas e de tomadas de decisões.

A grosso modo a lógica fuzzy tem a característica da representação de um sistema através do conhecimento do mesmo por parte de um operador experiente. Isto viabiliza uma representação mais precisa das não-linearidades e possibilita tomadas de decisões mais precisas para determinadas situações. O funcionamento de um algoritmo fuzzy é baseado na estimação de um estado do sistema a partir de uma série de palavras, por exemplo: uma temperatura pode estar quente ou fria, e uma ação sobre a mesma poderá ~~de tal forma~~ representar a ação do operador com experiência perante a mesma situação encontrada.

Uma fatia das aplicação da lógica fuzzy que tem se destacado nos últimos anos, é a aplicação em controladores de sistemas, isto por que com o advento do microprocessador e o conseqüente aumento da velocidade de processamento dos computadores, a partir dos anos 70 e 80, pode-se viabilizar a implementação de controladores fuzzy, pois os mesmos necessitam de uma base de hardware que seja suficiente para todas as operações necessárias.

Dentro do contexto mundial, os japoneses tem se dedicado de forma expressiva ao desenvolvimento de equipamentos que utilizam controladores fuzzy. Dentre estas aplicações destacam-se, a aplicação em eletrodomésticos (máquinas de lavar louças, máquinas fotográficas), controladores em base de hardware para serem programados e utilizados em microcomputadores, entre outras descritas em pesquisas encontradas nos artigos do IEEE [Bimal 94] [Hasan 94] [Homaiifar 95] [Tang 94] [Zhang 95].

Nestes artigos do IEEE foram estudadas várias aplicações de controladores fuzzy em máquina síncrona. A máquina síncrona faz parte do processo de geração de eletricidade de uma usina. Estes estudos culminaram num projeto de avaliação de desempenho dos controladores fuzzy em máquinas síncronas [Simas 95], na REIVAX Automação e Controle, executado por este que vos escreve. Com resultados

satisfatórios foi então definido um projeto final de curso que consistia na implementação de um controlador fuzzy, sendo este utilizado como estabilizador de potência (PSS, power system stabilizer), o qual faz parte de uma das malhas de controle da máquina síncrona, para servir como ferramenta de estudos de lógica fuzzy e controladores fuzzy, aplicados a este processo específico.

Neste relatório são apresentados elementos que fizeram parte deste projeto destacando-se os caminhos, as tarefas executadas e todos os tópicos estudados para se implementar um software e qual servirá de ferramenta de estudos, e num futuro próximo, de base para a implementação real de um controlador fuzzy para estabilizadores de potência de máquinas síncronas.

Para caracterizar este trabalho, foi feito um estudo aprofundado da lógica fuzzy, ressaltando aspectos matemáticos, teóricos e de aplicações em controladores, culminando na implementação na linguagem Microsoft C, de um controlador Fuzzy para se fazer simulações e comparações dentre controladores tipicamente utilizados

como estabilizadores de potência.

Este relatório está dividido em capítulos como segue abaixo:

Capítulo I =>O processo de Geração de potência. Neste capítulo irão ser apresentados os tópicos relevantes no processo de geração de potência representado pela máquina síncrona, identificando os elementos que serão determinantes no momento de se implementar um controlador.

Capítulo II =>Teoria da Lógica Fuzzy. Com o intuito de formar uma base teórica sobre lógica fuzzy este capítulo apresenta num contexto matemático os conceitos mais importantes, os quais poderão ser relevantes ao se implementar um controlador fuzzy.

Capítulo III =>Controladores Fuzzy. O capítulo de controladores fuzzy tem o objetivo de mostrar internamente como funciona um controlador fuzzy. Numa forma bem detalhada e minuciosa, serão apresentadas as operações, elementos e a forma de ajuste dos parâmetros, objetivando ao final um bom entendimento quanto a sua implementação.

Capítulo IV =>Metodologias de Concepção e Implementação de Controladores Fuzzy. Neste capítulo é feito um apanhado dos tópicos estudados durante a fase de estudos de controladores fuzzy, irão ser destacados aqui os mais recentes métodos para concepção e implementação de controladores fuzzy.

Capítulo V =>Placas de Controle Utilizadas na Implementação do Projeto. Este capítulo tem o objetivo de informar (sobre uma visão de usuário), o funcionamento e como foram dispostas no processo as placas utilizadas nas simulações do projeto do controlador fuzzy.

Capítulo VI =>O Software FuzSsd. Neste capítulo é apresentado as características do software implementado como ferramenta de estudos de controladores fuzzy. Tem como objetivo ensinar e demonstrar o funcionamento do mesmo na aplicação a máquina síncrona e como aplicá-lo em outros processos.

Capítulo VII =>Simulações. Na parte final deste relatório, são apresentadas as simulações feitas e as comparações entre controladores convencionais em funcionamento (atualmente nos PSS), estabilizadores de máquinas síncronas, e o controlador fuzzy implementado.

Anexos => => Nos anexos ^{apresentam-se} são apresentadas uma descrição sucinta do funcionamento das funções implementadas no software de controle fuzzy.

Onde são apresentadas: as funções implementadas no softw de controle; o relatório de ajustes do controlador e o diag. de blocos da maq sínc,

Capítulo I

O Processo de Geração de Potência

1.1 Introdução

O progresso da ciência trouxe-nos muitas vantagens e conforto. Podemos ver isso em nosso cotidiano, quando realizamos determinadas tarefas, ou em uma fábrica onde numa produção, os tempos de manufatura estão se tornando cada vez menores e portanto mais eficientes. *tempos mais eficientes?*

Mas praticamente todo este progresso necessita de uma fonte, a qual o mantenha funcionando, no caso, a eletricidade.

A eletricidade tem importância marcante em nossas vidas e nós nem percebemos, o não ser é claro, quando a mesma falta.

Sendo assim a engenharia criou meios de suprir estas necessidades projetando usinas hidroelétricas, termoelétricas, subestações, torres, postes e todo aparato necessário para que este bem não falte e seja de boa qualidade. Dentro deste contexto é necessário se fazer um controle da distribuição de eletricidade pelas regiões, tal que todas as necessidades sejam satisfeitas, e um controle sobre as potências geradas para que uma usina não seja sobrecarregada ou fique sub-utilizada.

Nestes controles de potências destacam-se as máquinas síncronas, processos os quais estão englobados as turbinas, rotores, estatores e linhas de transmissão, onde é gerada a potência para ser distribuída.

O sistema constituído pela máquina síncrona possui várias malhas de controle, onde podemos destacar as malhas de controle de velocidade da turbina e as malhas de estabilização de potência.

A estabilização de potência é o objeto de estudo deste projeto, onde no estabilizador de potência, também conhecido como PSS(Power System Stabilizer), é aplicado um controlador baseado em lógica fuzzy.

Neste capítulo são apresentados os tópicos mais relevantes no processo de geração de potência, destacando o PSS e sua utilidade na máquina síncrona.

1.2 O Funcionamento da Máquina Síncrona

Um processo de geração de eletricidade é muito complexo, ^P para simplificar um pequeno esquema é apresentado na figura 1.1 .

deveria ser mais completo e mais resumido!

No processo de geração de potência(figura 1.1), um fluxo de água, provido por uma barragem (represa) e controlado por uma comporta, escoo por uma tubulação e é injetado numa câmara, onde se encontram as pás da turbina hidráulica. A força do fluxo faz com que as pás da turbina girem.

A rotação produzida pelas pás é transmitida por meio de um eixo até um gerador de potência, o qual a partir da rotação do rotor faz gerar no estator uma tensão trifásica.

Esta eletricidade é então transmitida para as linhas de distribuição.

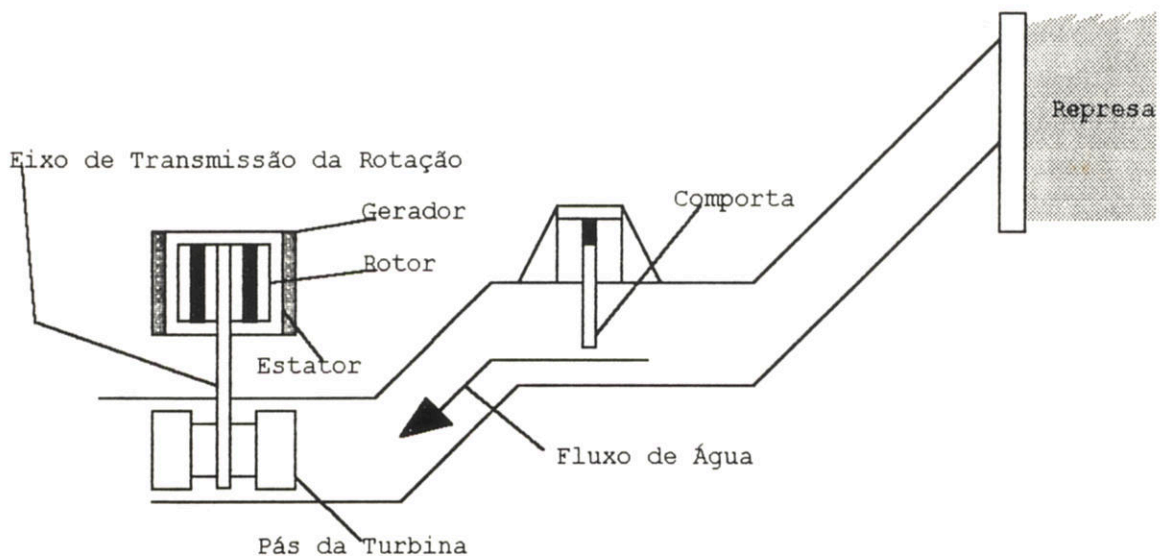


figura 1.1) Processo de Geração Elétrica

Uma das malhas de controle do sistema da máquina síncrona é mostrada na figura 1.2 . Nesta malha um regulador de velocidade atua sobre a comporta(Cpt), a qual vai fazer variar o fluxo de água que incide sobre as pás da turbina, isto acarreta uma variação da rotação da turbina hidráulica como descrito na figura 1.1, fazendo variar a potência gerada no rotor, esta variação de potência mecânica faz variar a potência elétrica gerada no estator.

Outro modo de variar a velocidade é a alteração do ângulo das pás da turbina. Este caso não foi objeto de estudo deste trabalho.

A diferença de potências cria uma aceleração da turbina, positiva quando a potência elétrica for menor do que a potência mecânica da turbina e negativa no caso contrário.

Esta malha de controle por possuir características mecânicas e fluídicas envolvidas, se torna muito lenta a ação de controle da comporta em relação e variação de potência gerada na saída da máquina síncrona. Isto é decorrido devido a constante de tempo resultante do sistema, a qual, será alta .

As consequências desta oscilação Δw , apesar de serem de baixa amplitude(< 10E-5), acarretam variações bruscas nas tensões de campo(Efd figura 1.3).

X
quais
oscila
coes?

Na figura 1.2 o termo $1/2H*s$, representa a inércia da turbina da máquina síncrona.

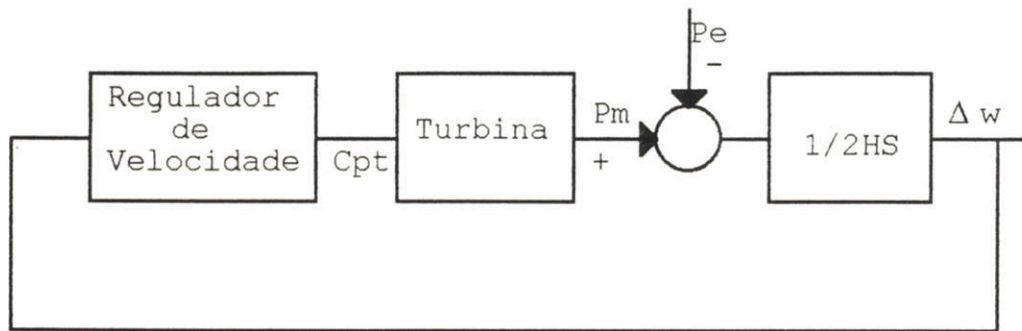


figura 1.2) Malha de controle da potência elétrica

A rotação de rotor induz uma corrente de campo no estator, que por sua vez é tratada e transmitida como tensão através das linhas de distribuição.

Esta tensão do estator é controlada por uma ponte tiristorizada, como mostra a figura 1.3 . Esta malha tem a característica de ter uma constante de tempo muito pequena, portanto produz respostas rápidas a variações no chaveamento dos tiristores.

O fato da constante de tempo do controle das tensões de campo serem pequenas, e este último sensível a variação de velocidade Δw , gera um problema de controle, que é o de variação da potência interna muito grande, debilitando a performance do gerador de potência.

Estas variações de potência estão também relacionadas com as variações de velocidade o que resulta numa dinâmica de controle muito oscilatória e pouco amortecida, causando debilidade na performance do gerador.

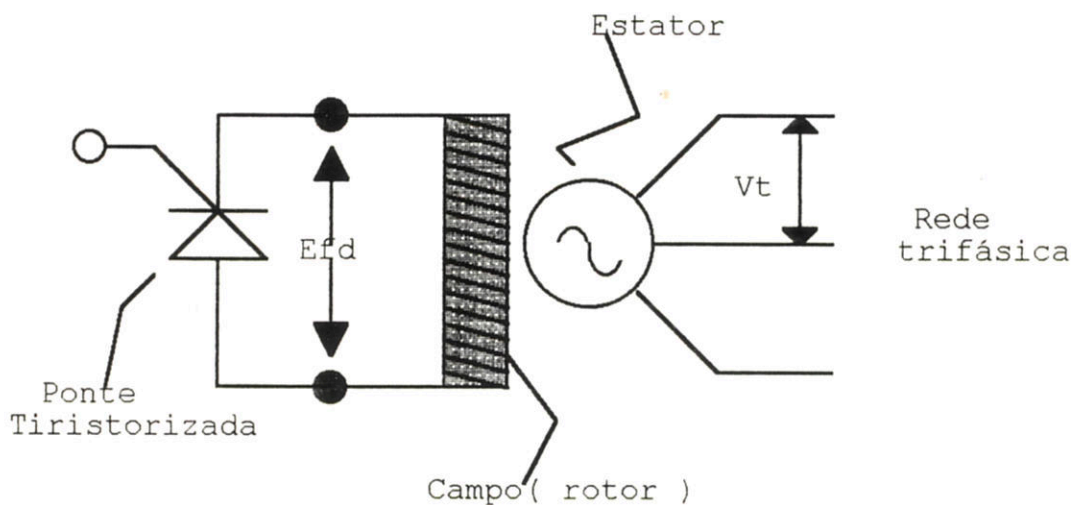


figura 1.3) Esquema de controle da tensão de campo por ponte tiristorizada

me-cho!

ou faz-se necessário, então,

Então se faz necessário o controle da variação de velocidade w através de um sistema que atue juntamente com a malha de controle de tensão de campo, de modo a amenizar estas oscilações.

O sistema que vai amenizar as oscilações é denominado estabilizador de potência (PSS Power System Stabilizer). O PSS age a partir da leitura da variação de potência elétrica ou velocidade do gerador, grandezas estas que são diretamente proporcionais. Veja figura 1.4.

Outra importante característica que interfere no desempenho do sistema de potência, é no momento em que há uma variação de carga do sistema, este evento acarreta uma variação de reatância externa, a qual não é controlada pela malha de controle de potência, portanto sendo também necessário considerar este parâmetro no momento de se ajustar o estabilizador de potência da máquina síncrona.

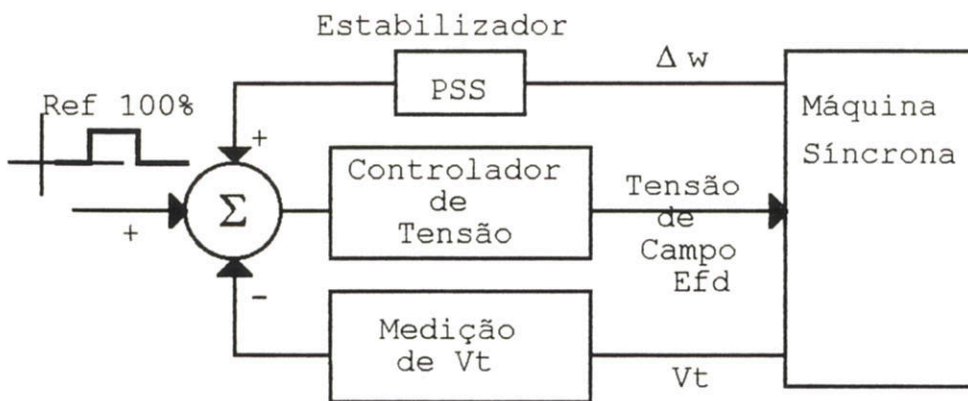


figura 1.4) Conexão do PSS ao processo do sistema de potência.

A figura 1.4 mostra como é conectado o PSS ao sistema de potência. A malha composta pela Medição de V_t e Controlador de Tensão é chamado de regulagem de tensão, pois, a referência "ref" indica a tensão de saída da máquina síncrona desejada. O PSS atua como regulador da variação da velocidade do rotor do gerador (Dw). Δw

A máquina síncrona descrita na figura 1.4 representa todas as malhas de controle e processo da turbina hidráulica demonstrados anteriormente. A máquina síncrona tem a característica de ser um processo de sétima ordem, multivariável e não-linear.

O PSS utilizado tipicamente se baseia em dois avanço-atraso com um ganho.

1.3 O software SSD

Com o intuito de desenvolver uma ferramenta de estudos de controladores em máquinas síncronas, a Reivax implementou um software ~~nomeado~~ denominado SSD (Simulador de Sistemas Dinâmicos).

O SSD foi implementado em Microsoft C, e tem, dentre outras possibilidades a condição da execução de ensaios de testes com uma máquina síncrona, definida dentre seus muitos blocos.

Esta máquina síncrona do SSD é a que foi utilizada para se fazer os testes e simulações do controlador fuzzy implementado. Isto foi possível também devido a possibilidade da comunicação do SSD com dados vindos externamente, lidos através da placa ADA 10 e envio de dados através da placa DAX.

O SSD não apenas pode ser utilizado em sistemas de potência como em qualquer outro processo em que seja necessária uma simulação, pois sua linguagem baseada em blocos pré-definidos, facilita de forma expressiva a construção dos modelos.

1.4 Conclusão

A capítulo teve como objetivo descrever a máquina síncrona em seus aspectos mais relevantes, destacando sempre o funcionamento dos componentes e sua importância dentro do sistema de geração elétrica, em uma usina hidroelétrica.

As pesquisas de novas tecnologias em sistemas de potência estão cada vez mais avançadas, com projetos de novas malhas de controle e CPU's mais dedicadas, tudo para que tenhamos em nossas casas a eletricidade tão necessária em nossas vidas.

Evitar sempre
que possível este
tipo de frase !!!

isto é comumente chamado de trajetória circular

Capítulo II

Teoria da Lógica Fuzzy

2.1 Introdução

Como em tudo o que estudamos, sempre se faz necessário um conhecimento teórico sobre o assunto em questão, isto cria um embasamento científico, que será útil para obtermos as ferramentas necessárias para aplicarmos da melhor forma possível este conhecimento.

Com este intuito o capítulo presente tem o objetivo de apresentar os teoremas e definições sobre a lógica fuzzy. Objetivando principalmente a elucidação dos principais conceitos aplicáveis em controladores.

Este estudo poderá ser necessário quando se for implementar ou criar seu próprio controlador (ou outra aplicação) utilizando a lógica Fuzzy [Kaufmann 75] [Klir 81] [Dubois 80].

No próximo capítulo partiremos para as discussões sobre a implementação e metodologias de construção dos controladores baseados em lógica fuzzy.

2.2 Noções fundamentais sobre conjuntos e função de pertinência.

Seja E um conjunto de elementos xi e A um sub-conjunto tal que:

$$A \subset E;$$

Seja $\mu_A(x_i)$ uma função que indica a pertinência (1) ou não (0) de um elemento do conjunto A ao conjunto E. Esta função é definida da seguinte forma:

$$\mu_A(x_i) = 1 \text{ se } x \in A$$

$$= 0 \text{ se } x \notin A;$$

Seja, por exemplo :

$$E = \{ x_1, x_2, x_3, x_4, x_5 \} \text{ e}$$

Acho que devia ser do conjunto E ao subconjunto A

Não interessa aqui.



$$A = \{x_2, x_3, x_5\}$$

então:

$$\mu_A(x_1) = 0, \mu_A(x_2) = 1, \mu_A(x_3) = 1, \mu_A(x_4) = 0, \mu_A(x_5) = 1;$$

Podemos definir outro sub-conjunto, \bar{A} , como sendo o conjunto de elementos do conjunto E que não estão presentes em A, então:

$$\bar{A} = \{x_1, x_4\}$$

aplicando a função de pertinência μ , então:

$$\mu_{\bar{A}}(x_1) = 1, \mu_{\bar{A}}(x_2) = 0, \mu_{\bar{A}}(x_3) = 0, \mu_{\bar{A}}(x_4) = 0, \mu_{\bar{A}}(x_5) = 0;$$

Se \bar{A} é o complemento de A em relação a E então:

$$A \cap \bar{A} = \emptyset,$$

$$A \cup \bar{A} = E.$$

Agora dados dois conjuntos A e B, sendo que A e B estão contidos em E, podemos definir a operação de intersecção como:

$$A \cap B \Rightarrow \mu_{[A \cap B]}(x) = 1 \text{ se } x \in A \text{ e } B$$

$$= 0 \text{ se } x \notin A \text{ e } B$$

portanto podemos concluir que

$$\mu_{A \cap B}(x) = \mu_A(x) * \mu_B(x) \text{ para este caso; e a união como:}$$

$$A \cup B \Rightarrow \mu_{[A \cup B]}(x) = 1 \text{ se } x \in A \text{ ou } B$$

$$= 0 \text{ se } x \notin A \text{ ou } B;$$

Continuando o exemplo anterior e definindo o conjunto B como $\{x_1, x_3, x_5\}$, temos:

esta notação confunde. Não devia ser $\mu_{[A \cap B]}(x)$?

não está faltando um equivalente a isto para a união.

$$A = \{(x1|0),(x2|1),(x3|1),(x4|0),(x5|1)\}$$

$$B = \{(x1|1),(x2|0),(x3|1),(x4|0),(x5|1)\};$$

onde A e B são representados pelo par $(x_i | \mu(x_i))$.

Agora podemos calcular:

$$\mu_{A \cap B}(x):$$

$$A \cap B = \{(x1|0),(x2|0),(x3|1),(x4|0),(x5|1)\}$$

$$\mu_{A \cup B}(x):$$

$$A \cup B = \{(x1|1),(x2|1),(x3|1),(x4|0),(x5|1)\}$$

2.3 Sub-conjuntos Fuzzy

Extendendo agora um pouco mais o nosso campo de visão, vamos imaginar que um elemento x_i do conjunto E não pertença totalmente ao conjunto A, ou seja, não necessariamente a função $\mu(x_i)$ tenha um valor 1 ou 0, e sim, possa variar entre no intervalo $[0,1]$, que é uma condição análoga a uma porcentagem de pertinência do elemento de A em relação ao conjunto E.

Portanto o elemento x_i de E pode não ser um elemento de A ($\mu(x_i)$ exatamente zero), pode ser elemento de A um "pouco" ($\mu(x_i)$ próximo a 0), pode ser um elemento médio de A ($\mu(x_i)$ próximo a 0.5), pode ser "praticamente" elemento de A ($\mu(x_i)$ próximo de 1), ou finalmente pode ser completamente elemento de A ($\mu(x_i)=1$).

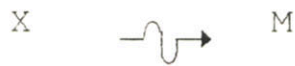
Então um exemplo do conjunto A escrito da forma $(x_i | \mu(x_i))$, poderá ser apresentado como abaixo:

$$A = \{(x1|0.2),(x2|0),(x3|0.3),(x4|1),(x5|0.8)\}$$

Portanto um sub-conjunto fuzzy A de E é escrito de acordo com o par:

$$\{(x_i | \mu_A(x_i)), \forall x_i \in E,$$

onde $\mu_A(x_i)$ é a função que determina o grau pertinência do elemento de x_i em A em relação ao conjunto E. Esta função $\mu_A(x_i)$ tem seu mapa definido através de uma função membro $M_A(x_i)$ que varia de 0 a 1.



$$\mu_a(x_i)$$

Podemos observar que se $M=\{0,1\}$ teremos uma função característica da lógica booleana, ou seja, sim ou não, que no caso, comportando-se a mesma como uma função não-Fuzzy.

Outros exemplos de funções M podem ser assim descritos

$$M=[0, 1];$$

$$M=\{0, 1/2, 1\};$$

$M=\{0, 0.25, 0.5, 0.75, 1\}$, a escolha é feita de acordo com a aplicação.

2.3.1 Relação de Dominância

Dado $v=(kv_1, kv_2, \dots, kv_n)$ e

$u=(ku_1, ku_2, \dots, ku_n)$ para $n=1, 2, 3,$

dizemos que u domina v , escrevendo como

$$u \geq v,$$

se e somente se

$$ku_1 \geq kv_1, ku_2 \geq kv_2, \dots, ku_n \geq kv_n.$$

se $ku_i > kv_i \forall ku_i \in u$ e $kv_i \in v, i=1, 2, \dots, n$

u é estritamente dominante,

$$u > v,$$

exemplo:

$$u = (7, 3, 0, 5)$$

$$v = (4, 2, 0, 3)$$

Como $7 \geq 4$, $3 \geq 2$, $0 \geq 0$, $5 \geq 3$, então $u \succeq v$.

2.3.2 Operações em Sub-conjuntos Fuzzy

A partir dos conceitos apresentados acima, nesta seção serão demonstrados as operações entre sub-conjuntos fuzzy mais utilizadas e descritas em [Kaufmann].

Seja E um conjunto de elementos xi, M uma função membro associada, e A e B dois sub-conjuntos fuzzy de E. Podemos então definir as operações abaixo entre os sub-conjuntos A e B:

2.3.2.1 Inclusão - Nós dizemos que A esta incluso em B se:

$$\forall xi \in E : \mu_A(x) < \mu_B(x),$$

ou simplesmente

$$A \subset B.$$

2.3.2.2 Igualdade - A e B serão iguais se e somente se

$$\forall xi \in E : \mu_A(x) = \mu_B(x),$$

ou

$$A = B.$$

2.3.2.3 Desigualdade - A desigualdade existe quando a igualdade não é satisfeita, ou seja

$$\forall xi \in E : \text{existe pelo menos um xi de A e B é tal que}$$



$$\mu_A(xi) \neq \mu_B(xi). \text{então } A \neq B.$$

2.3.2.4 Complementação - Dada a função M [0 1] B pode ser escrito como complemento de A, se:

$$\mu_B(xi) = 1 - \mu_A(xi)$$

então $B = \overline{A}$ ou $\overline{A} = B$,

então pode-se concluir também que $\overline{(\overline{A})} = A$.

2.3.2.5 Intersecção - definida por $A \cap B$. Compreende ~~em~~ um conjunto que contenha elementos que apareçam mesmo tempo no conjunto A e B. Descrevemos a intersecção da seguinte forma



$$\forall x_i \in E : \mu_{A \cap B}(x_i) = \text{MÍNIMO}(\mu_A(x_i), \mu_B(x_i))$$

Esta operação é mais conhecida por AND fuzzy [Cox 95], segundo a definição de Lofti Zadeh.

obs: ao invés de se escrever MÍNIMO pode-se simplificar e utilizar apenas MIN, ou \wedge no lugar da vírgula ou dentro dos parenteses, como exemplo abaixo:

isto
atrapalha

$$\mu_{A \cap B}(x_i) = (\mu_A(x_i) \wedge \mu_B(x_i)).$$

2.3.2.6 União - definida por $A \cup B$. Compreende em um conjunto que contenha todos os elementos que apareçam em A e B, descrevemos a união da seguinte forma

$$\forall x_i \in E : \mu_{A \cup B}(x_i) = \text{MÁXIMO}(\mu_A(x_i), \mu_B(x_i)).$$

Esta operação permite introduzir o OR fuzzy.

obs: ao invés de se escrever MÁXIMO pode-se simplificar e escrever apenas MAX ou \vee , este último também escrito de acordo com a forma de notação do Mínimo.

2.3.2.7 Soma Disjuntiva - Definida pela união das intersecções de $A \bar{B}$ e $\bar{A} B$ dispostos da seguinte forma:

$$A \oplus B = (A \cap \bar{B}) \cup (\bar{A} \cap B);$$

2.3.2.8 Diferença - A diferença entre Sub-conjuntos Fuzzy é definida por

$$A - B = A \cap \bar{B}.$$

2.3.2.9 Distância de Hamming - Seja E finito, a distância de Hamming é um escalar que indica a distancia relativa entre dois sub-conjuntos fuzzy. É definida por

$$d(A, B) = \sum_{i=1}^n |\mu_A(x_i) - \mu_B(x_i)|.$$

2.3.2.10 Distância de Hamming Relativa - É definida por

$$\delta(A, B) = (1/n) * d(A, B),$$

onde n é o número de elementos de E .

3.2.2.11 Distância Euclidiana - É definida por

$$e(A,B) = \sqrt{\sum |\mu_A(x_i) - \mu_B(x_i)|^2}$$

2.3.2.12 Distância Euclidiana Relativa - É definida por

$$\varepsilon(A,B) = (1/n) * e(A,B),$$

onde n é o número de elementos de E .

2.3.2.13 Medição de Entropia de um Sub-conjunto Fuzzy - A entropia nos indica o grau de desordem entre os elementos de um sub-conjunto Fuzzy.

Para a medição da entropia, primeiro definimos o elemento $\pi_A(x_i)$ para todo $x_i \in A$, então

$$\pi_A(x_i) = \frac{\mu_A(x_i)}{\sum_{i=1}^N \mu_A(x_i)},$$

onde N é o número de elementos de A .

Então a entropia é definida por:

$$H = (1/\ln(N)) * \left(\sum \pi_A(x_i) * (\ln(\pi_A(x_i))) \right).$$

2.3.2.14 Decomposição - Um sub-conjunto fuzzy pode ser decomposto em um produto de sub-conjuntos por um coeficiente α_i :

$$A = \text{MAX}[\alpha_1.A\alpha_1, \alpha_2.A\alpha_2, \dots, \alpha_n.A\alpha_n],$$

$$0 < \alpha_i < 1 \quad i = 1, 2, \dots, n$$

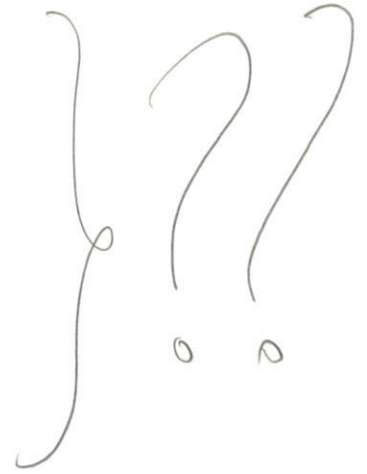
$$\mu_{A\alpha_i}(x_i) = 1 \quad \text{se } \mu_A(x_i) > \alpha_i$$

para que
sirva?

$$0 \quad \text{se } \mu_A(x_i) < \alpha_i$$

exemplo:

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & \\ 0 & 0.2 & 0.5 & 0.7 & 1 & = \text{MAX} ((0.2) \cdot (\begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0 & 1 & 1 & 1 & 1 \end{array}) , \\ & & & & & (0.5) \cdot (\begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0 & 0 & 1 & 1 & 1 \end{array}) , \\ & & & & & (0.7) \cdot (\begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0 & 0 & 0 & 1 & 1 \end{array}) , \\ & & & & & (1) \cdot (\begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0 & 0 & 0 & 0 & 1 \end{array}) . \end{array}$$



2.3.3 Conjunto de sub-conjuntos fuzzy para M e E finitos

Seja, por exemplo, $E = \{x_1, x_2, x_3\}$;

então define-se um conjunto $P(E)$ como sendo o conjunto de todos os sub-conjuntos possíveis de E , como segue abaixo:

$$P(E) = \{ \emptyset, \{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{x_1, x_2, x_3\} \}.$$

[isto porque neste caso $M = \{0, 1\}$]

Observamos que em $P(E)$ $2^3 = 8$ elementos.

Se considerarmos $E = \{x_1, x_2\}$ e $M = \{0, 1/2, 1\}$, $P(E)$ terá 9 elementos,

$$P(E) = \{ \{(x_1|0), (x_2|0)\}, \{(x_1|0.5), (x_2|0.5)\}, \dots \}$$

ou seja $3^2 = 9$, que portanto nos permite concluir que o número de Sub-conjuntos possíveis, $P(E)$, é indicado por m^n , onde m é o número de elementos de M e n é o número de elementos de E .

2.3.4 Produto e Soma Algébricos de Sub-conjuntos Fuzzy

Seja E um conjunto Fuzzy, $M[0, 1]$, A e B dois sub-conjuntos de E , o produto algébrico $A.B$ é dado por

$$\forall x_i \in E \quad \mu_{A.B}(x_i) = \mu_A(x_i) \cdot \mu_B(x_i),$$

A soma algébrica é definida por:

$$\forall x_i \in E \quad \mu_{A+B}(x_i) = \mu_A(x_i) + \mu_B(x_i) - \mu_{A \cdot B}(x_i).$$

2.4 Grafos Fuzzy e Relações Fuzzy

2.4.1 Grafos

Uma forma de representação de sub-conjuntos fuzzy, é através de grafos, os quais podem aparecer na forma de tabelas ou gráficos.

Os grafos fuzzy são constituídos de dois ou mais sub-conjuntos fuzzy,

As tabelas formam uma matriz, onde os elementos representam o grau de pertinência das variáveis colocadas nas linhas e colunas de cada conjunto. Outra forma de tabela é pelo grafo de Berge, onde são colocadas cores nos elementos da matriz, para destacar o grau de pertinência das variáveis dos conjuntos dispostos nas linhas e colunas da mesma.

Demonstração:

Sejam dois conjuntos E_1, E_2 , sendo que os elementos x pertencem a E_1 e os elementos y a E_2 . O par (x, y) define o produto $E_1 \times E_2$

O sub-conjunto fuzzy G é tal que

$$\forall (x, y) \in E_1 \times E_2 : \mu_G(x, y) \in M,$$

onde M é uma função membro no intervalo $[0, 1]$ do conjunto $E_1 \times E_2$, se isto é satisfeito então G é uma função gráfica fuzzy.

Abaixo na figura 2.1 temos uma representação de um grafo por tabela:

Seja $E_1 = \{x_1, x_2, x_3\}$ e

$$E_2 = \{y_1, y_2, y_3, y_4, y_5\}$$

	y1	y2	y3	y4	y5
x1	0	0	0.1	0.3	1
x2	0	0.8	0	0	1
x3	0.4	0.4	0.5	0	0.2

figura 2.1) Representação de um grafo

Generalizando o conceito de grafos podemos determiná-lo pelo produto n-dimensional de:

$\exists x \exists x_2 \dots \exists x_n$ tal que $x(i) \in E_i \quad i = 1, 2, \dots, n$,

$\forall (x(1), x(2), \dots, x(n)) \subset E_1 \times E_2, \dots, E_n :$

$\mu(x(1), x(2), \dots, x(n)) \subset M$.

2.4.2 Relações

Seja P um conjunto de n conjuntos fuzzy e M a função membro, a relação enésima fuzzy é o sub-conjunto P indicado pelos os valores de M .

A figura 2.1 se mostra como uma relação dos conjuntos E_1 e E_2 (P), onde cada relação correspondente entre x_i e y_i é indicada por um número $\in M [0, 1]$.

Então uma relação entre sub-conjuntos fuzzy pode ser determinada através de equações contínuas como o exemplo descrito abaixo:

Seja $E_1 = E_2 = R$ onde $R(-\infty, \infty)$, x e $Y \in R$,

$$\begin{aligned} \mu_R(x,y) &= 0 && \text{se } y > x \\ &= (1/(1+(1/(x-y)))) && \text{se } y < x, \end{aligned}$$

Uma forma de notação da relação $E_1 \times E_2$ é através do símbolo \mathcal{R}

$$x_i \in E_1 \quad y_i \in E_2 \implies x \mathcal{R} y$$

2.4.2.1 Projeção de uma relação: A projeção de relação é dividida em:

i) Primeira projeção - definida por

$$\mu^{(1)}(x) = \bigvee \mu_{\mathcal{R}}(x,y), \text{ conjunto dos máximos de cada coluna;}$$

ii) Segunda Projeção - definida por

$$\mu^{(2)}(x) = \bigvee \mu_{\mathcal{R}}(x,y), \text{ conjunto dos máximos em cada linha;}$$

iii) Projeção global - definida por

$$h(R) = \bigvee \bigvee \mu_{\mathcal{R}}(x,y) \text{ ou}$$

$$\bigvee \mu_{\mathcal{R}}(x,y), \text{ máximo entre os conjuntos resultantes}$$

da primeira e segunda projeção.

Se a projeção global é igual a 1 a relação é dita normal. Se a projeção global é menor que 1 a relação é dita subnormal.

2.4.2.2 Elementos de uma Relação Fuzzy

2.4.2.2.1 Suporte de uma Relação: Suporte S de uma relação é o conjunto de pares (x,y) tal que $\mu_{\mathcal{R}}(x,y) > 0$,

$$S(\mathcal{R}) = \{(x,y) / \mu_{\mathcal{R}}(x,y) > 0\}$$

2.4.2.2.2 Envoltente de uma Relação: Seja \mathcal{R} e \mathcal{L} duas relações tal que:

$$\forall (x,y) \in E_1 \times E_2 : \mu_{\mathcal{R}}(x,y) < \mu_{\mathcal{L}}(x,y),$$

dizemos então que \mathcal{L} é um envoltente de \mathcal{R} , ou seja, $\mathcal{R} \subset \mathcal{L}$.

2.4.2.2.3 União de duas Relações: Uma união de duas relações é definida por

$$\mathcal{R} \cup \mathcal{L}$$

e descrita como abaixo,

$$\mu_{\mathcal{R} \cup \mathcal{L}}(x,y) = V(\mu_{\mathcal{R}}(x,y), \mu_{\mathcal{L}}(x,y)).$$

2.4.2.2.4 Intersecção de duas Relações: A intersecção é definida por

$$\mathcal{R} \cap \mathcal{L}$$

e descrita como abaixo,

$$\mu_{\mathcal{R} \cap \mathcal{L}}(x,y) = \wedge(\mu_{\mathcal{R}}(x,y), \mu_{\mathcal{L}}(x,y)).$$

2.4.2.2.5 Produto e Soma Algébrica de duas Relações: O produto algébrico entre duas relações é descrito por:

$$\mathcal{R} \cdot \mathcal{L}$$

e definido pela expressão :

$$\mu_{\mathcal{B} \cdot \mathcal{L}}(x,y) = \mu_{\mathcal{B}}(x,y) \cdot \mu_{\mathcal{L}}(x,y),$$

A soma algébrica é descrita por

$$\mathcal{B} + \mathcal{L}$$

e é definida pela expressão:

$$\mu_{\mathcal{B} + \mathcal{L}}(x,y) = \mu_{\mathcal{B}}(x,y) + \mu_{\mathcal{L}}(x,y) -$$

$$\mu_{\mathcal{B}}(x,y) \cdot \mu_{\mathcal{L}}(x,y)$$

2.4.2.2.6 Complemento de uma Relação: O complemento de uma relação é descrito por $\overline{\mu_{\mathcal{B}}(x,y)}$, e definido como abaixo,

$$\forall (x,y) \in E_1 \times E_2 : \overline{\mu_{\mathcal{B}}(x,y)} = 1 - \mu_{\mathcal{B}}(x,y),$$

2.4.2.2.7 Disjunção de duas Relações: A disjunção é descrita por $\mathcal{B} \cup \mathcal{L}$ e definida segundo a expressão abaixo,

$$\mu_{\mathcal{B} \cup \mathcal{L}}(x,y) = (\mu_{\mathcal{B}} \wedge \mu_{\mathcal{L}}) \vee (\overline{\mu_{\mathcal{B}}} \wedge \overline{\mu_{\mathcal{L}}}).$$

2.4.2.2.8 Composição de duas Relações

Seja R_1 uma relação $X \times Y$ e R_2 uma relação $Y \times Z$, composição de duas relações é descrita por

$$\mu_{R_1 \circ R_2} / R_1 \in X \times Y \quad R_2 \in Y \times Z \text{ tal que}$$

$$\mu_{R_1 \circ R_2}(x,z) = \bigvee (\mu_{R_1}(x,y) \wedge \mu_{R_2}(y,z))$$

onde (y de R_1 = y de R_2).

A operação de composição é efetuada de acordo com as regras da multiplicação entre duas matrizes, entretanto ao invés de se multiplicar os elementos encontra-se o menor, e, ao invés da soma final para encontrar o resultado procura-se o máximo dentre os mínimos achados anteriormente.

2.4.2.2.9 Sub-conjunto Comum de Nível α em uma Relação: Este subconjunto é definido por:

$$G_2 = \{(x,y) / \mu_R(x,y) > \alpha\} \quad \alpha \in \text{ao intervalo } 0 \text{ a } 1$$

2.4.2.2.10 Teorema da Decomposição de uma Relação :

A decomposição consiste em separar a relação em um somatório fuzzy (OU, Máximo) de produtos escalares de sub-relações por um coeficiente α . A decomposição é descrita por:

$$\mathcal{R} = \bigvee \alpha \cdot \mathcal{R}_\alpha, \quad 0 < \alpha < 1, \text{ onde}$$

$$\mu_{\mathcal{R}_\alpha}(x,y) = 1 \text{ se } \mu_{\mathcal{R}}(x,y) > \alpha$$

$$0 \text{ se } \mu_{\mathcal{R}}(x,y) < \alpha$$

2.4.2.3 Propriedades das Relações Fuzzy

Seja $E_1 = E_2 = E = \{A, B, C, D, E\}$ e $M [0, 1]$. A relação $\mathcal{R} \times \mathcal{R}$ é dada por $\mu_{\mathcal{R}}(x,y)$ para todo par x,y . Partindo deste pressuposto podemos definir as seguintes propriedades de uma relação:

2.4.2.3.1 Simetria : Uma relação é dita simétrica se e somente se :

$$\forall (x,y) \in E \times E : \mu_{\mathcal{R}}(x,y) = \mu_{\mathcal{R}}(y,x);$$

2.4.2.3.2 Reflexividade : Uma relação é dita se :

$$\forall (x,y) \in E \times E : \mu_{\mathcal{R}}(x,x) = 1;$$

2.4.2.3.2 Antireflexividade : Uma relação é antireflexiva se :

$$\forall (x,y) \in E \times E : \mu_{\mathcal{R}}(x,x) = 0;$$

2.4.2.3.3 Transitividade : A transitividade é definida por :

$$\forall (x,y),(y,z),(x,z) \in E \times E :$$

$$\mu_{\mathcal{R}}(x,z) > \bigvee (\mu_{\mathcal{R}}(x,y) \wedge \mu_{\mathcal{R}}(y,z));$$

2.4.2.4 Relações de pré-Ordem

Uma relação é dita de pré-ordem Fuzzy se nela forem encontradas as seguintes propriedades:

- * Transitividade
- * Reflexividade.

Teorema) Se uma relação é transitiva e reflexiva(isto é de pré-ordem), então podemos afirmar que,

$$\mathcal{R} = \mathcal{R}^2, k = 1, 2, 3, \dots$$

onde $\mathcal{R}^2 = \mathcal{R} \oplus \mathcal{R}$, composição de \mathcal{R} e \mathcal{R} .

2.4.2.5 Relação de Similitude

Uma relação é dita de Similitude quando esta relação apresenta as seguintes propriedades:

- * Transitividade;
- * Reflexividade;
- * Simetria.

Exemplo:

\mathcal{R}

	x1	x2	x3	x4	x5	x6	x7
x1	1	a1	a1	a1	a1	a1	a1
x2	a1	1	a2	a2	a2	a2	a2
x3	a1	a2	1	a3	a3	a3	a3
x4	a1	a2	a3	1	a4	a4	a4
x5	a1	a2	a3	a4	1	a5	a5
x6	a1	a2	a3	a4	a5	1	a6
x7	a1	a2	a3	a4	a5	a6	1	.
.
.
.

Se $0 < a_1 < a_2 < a_3 < \dots < a_k < 1$; esta relação é dita de similitude, pois se verificam as três propriedades necessárias.

2.5 Conclusão

Neste capítulo foi apresentado um apanhado geral dos conceitos, definições e teoremas que abrangem a lógica fuzzy. O objetivo principal aqui é ambientar-se dentro do conceito da lógica fuzzy, pois a mesma, aparentemente não parece ser de clareza imediata.

A complexidade e as dificuldades para um melhor entendimento dos tópicos

OTIMO !!

aqui descritos, podem ser estudados nas bibliografias colocadas no final deste relatório [Dubois 80] [Klir 81] [Kaufmann 75].

* Obviamente que nem tudo o que está aqui será utilizado nos controladores fuzzy, entretanto podem servir de base para aplicações em outras áreas científicas, onde se deseje aplicar a teoria da lógica fuzzy.

* Dado todo este embasamento teórico temos agora a oportunidade de ~~se~~ prosseguir no estudo da lógica fuzzy. Os passos seguintes foram direcionados a pesquisa nas publicações do IEEE ([Bimal 94] [Hasan 94] e [Tang 94]) e livros, [Cox 95], as ~~quais~~ ^{quais} apresentavam resultados ~~de~~ de simulações e de ~~conexão~~ ^{uso} de controladores Fuzzy em sistemas de potência.

Estas pesquisas serviram de base para a decisão da implementação de simulações do controlador fuzzy, e avaliação de sua performance dentro das condições exigidas.

Capítulo III

Controladores Fuzzy

3.1 Introdução

Lógica fuzzy como já apresentado se distingue de outros modelos matemáticos do conhecimento, por representar a incerteza dos parâmetros de um sistema.

Não são precisas das c/exatidão

Esta característica a torna muito útil em aplicações onde as variáveis envolvidas não tem valores exatos e possuem um grau de incertezas. Dentre estas aplicações destacam-se a biologia, economia, inteligência artificial, e engenharia.

Em engenharia a lógica fuzzy é aplicada basicamente em controladores de processos, e vem nos últimos anos se destacando perante os demais tipos de controladores como o PID, por apresentar resultados comparativos satisfatórios, incentivando o desenvolvimento de ferramentas e equipamentos baseados na mesma.

A única característica que ainda inviabiliza uma melhor aceitação do controlador fuzzy, é o fato de ainda não ter sido comprovada sua estabilidade matematicamente. Fica assim aqui uma boa dica de futuros trabalhos que podem ser bem reconhecidos pela comunidade científica. !!

Neste capítulo são apresentados os elementos que formam um controlador fuzzy, da forma mais esplanada possível associando os elementos estudados no capítulo anterior.

dos ministros?

3.2 Pensar Fuzzy

Quando se tem uma variável e esta é utilizada em um controlador convencional, se faz necessário que ela seja a mais exata possível, dentro dos limites exigidos de precisão do controlador e do processo a ser controlado.

A vantagem do controlador fuzzy é a de representar a incerteza de uma medição, associando a medição a uma palavra que expresse o estado do sistema, esta palavra é chamada de variável fuzzy.

Para um indivíduo a temperatura de um forno pode estar "morna", já para outro a mesma pode estar um pouco mais quente, como representar isto? *mudar!*

A resposta é simples, observemos que será necessário indicar um limite para cada variável fuzzy que expresse o estado, no exemplo poderíamos representar graficamente os limites como na figura 3.1 .

Já imaginou uma prensa "pesada"?

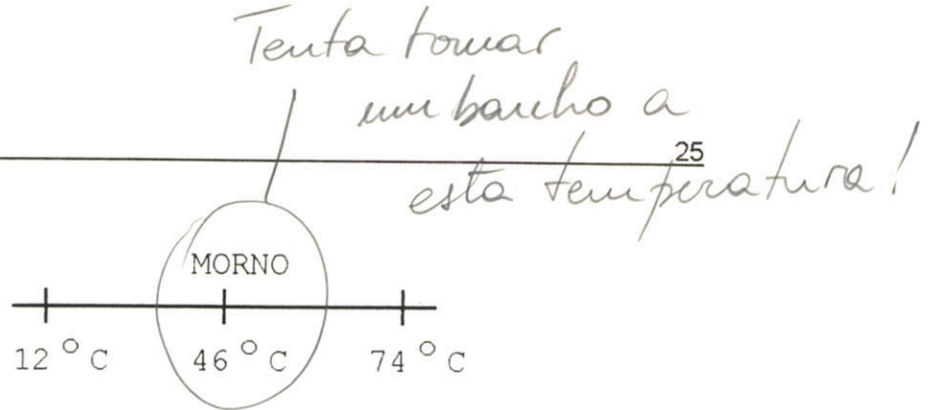


figura 3.1) Exemplo da representação dos limites para temperatura morna de um forno

Observando o desenho acima, podemos ver que se a temperatura for 46° C afirmamos que a mesma é morna. E quando nos aproximamos dos limites, como poderemos representar a temperatura "Morna" sabendo que ela já não é mais tão morna.

Para isto associa-se um percentual, ou melhor, um grau de pertinência o qual adotamos a letra grega μ para indicar o quanto pertence um dado estado do sistema a variável fuzzy. Neste caso, podemos de nosso exemplo ficaria assim:

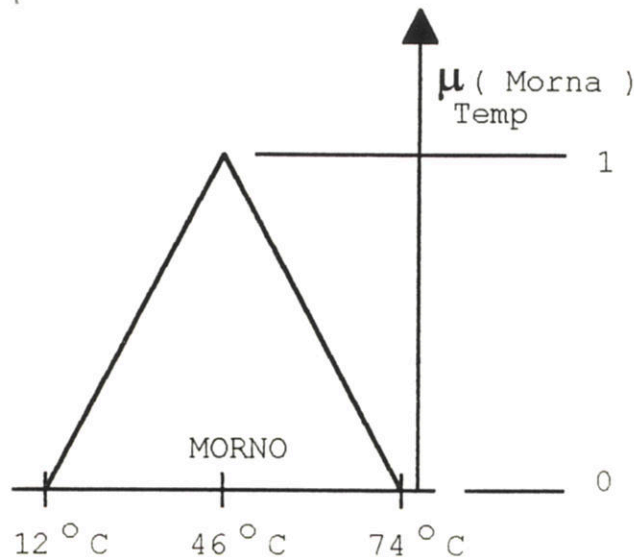


figura 3.2) Associação de um grau de pertinência a temperatura "morna"

Portanto agora podemos associar um grau de pertinência a um estado representado por uma variável do sistema, no caso acima temperatura morna. Esta função resultante na figura acima é chamada função membro fuzzy da temperatura, para este exemplo, ou simplesmente sub-função fuzzy da variável temperatura. As sub-funções fuzzy, por definição também podem ser chamadas de funções de pertinência de uma variável do controlador.

Este grau de pertinência pode variar de zero, quando uma dada temperatura não pertence a sub-função fuzzy, até 1 quando a temperatura esta localizada no ponto de máximo da sub-função.

Com estes conceitos analisados, agora pode-se incluir arbitrariamente quantas sub-funções fuzzy se queira, de forma a alcançarmos a estabilidade, a dinâmica desejada e a abrangência do controlador fuzzy.

Generalizando as sub-funções fuzzy para todo o intervalo, podemos definir como serão construídas as sub-funções das variáveis para alcançarmos as características indicadas acima. Para tanto, em nosso exemplo temos a temperatura de um forno como uma variável fuzzy, e temos que definir o conjunto das sub-funções e seus respectivos intervalos. Na figura 3.3 temos o exemplo de uma função fuzzy para a temperatura de forno.



figura 3.3) Arranjo dos limites para um conjunto de sub-funções fuzzy da variável temperatura

Este tipo de função fuzzy é construída para cada variável que fará parte do controle do sistema. No caso da máquina síncrona foram construídas funções fuzzy para o erro, velocidade do erro, que são as variáveis do processo disponíveis, e ação de controle do estabilizador de potência (PSS) fuzzy.

Tendo em mãos as representações das variáveis envolvidas no processo, agora podemos definir como são executados os cálculos a partir das entradas de modo a chegarmos a saída do controlador fuzzy.

Num controle fuzzy as variáveis de entrada são chamadas de antecedentes e as de saída conseqüentes. Para tanto, são executadas operações fuzzy entre os antecedentes, chamadas de intersecção, e a partir da avaliação por uma base de regras calcula-se os conseqüentes através das operações de união, e por fim calcula-se a saída em termos físicos do controlador fuzzy para o processo.

A operação entre antecedentes inicia com a fuzzificação, a qual consiste em ler as variáveis do processo e encontrar as sub-funções fuzzy e seus respectivos graus de pertinência. Um exemplo pode ser visto na figura 3.4 abaixo onde para uma temperatura lida, pode-se avaliar o estado do forno pelos graus de pertinência e nas sub-funções fuzzy.

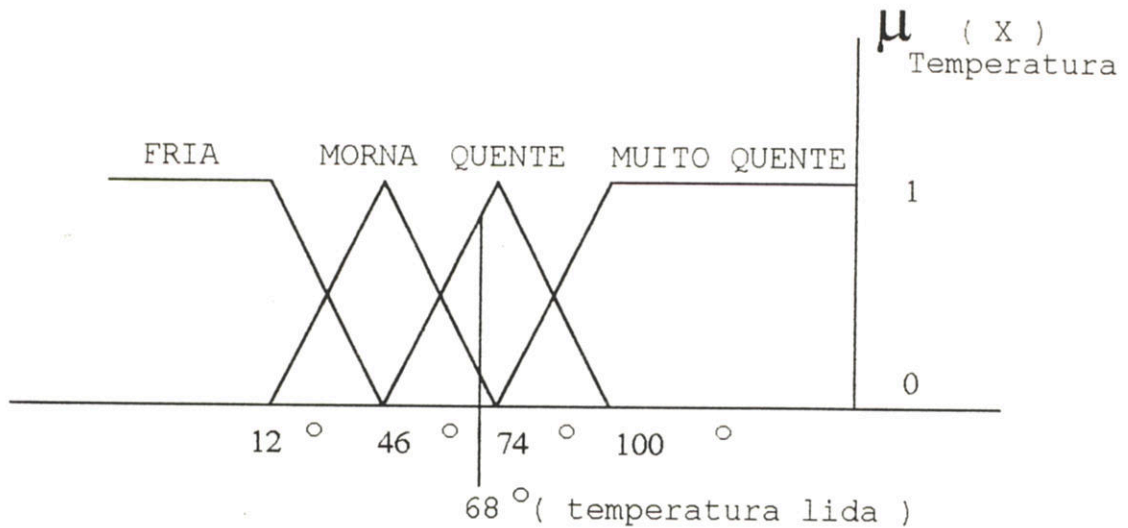


figura 3.4) Fuzzificação =>Leitura e determinação das funções e graus de pertinência a que pertencem os dados

A partir desta leitura do controlador fuzzy, no caso a temperatura é Morna com grau de pertinência $\mu(\text{Morna})$ igual a 0.2 e Quente com grau de pertinência $\mu(\text{Quente})$ igual a 0.8. Avalia-se através de uma base de regras quais serão as sub-funções da ação de controle que serão utilizadas no cálculo da saída do controlador.

		Variação de Temperatura		
		Baixa	Média	Alta
T e m p e r a t u r a	Fria	Baixa	Média	Alta
	Morna	Baixa	Média	Alta
	Quente	Média	Baixa	Média
	Muito Quente	Alta	Média	Baixa

Ação de Controle

*falta lógica.
não é dito
que a quer!!*

figura 3.5) Base de Regras

A variação de temperatura constitui de outra entrada suposta disponível para este exemplo.

A base de regras funciona de acordo com a expressão IF.< Antecedentes >. THEN.< Conseqüentes >. Por exemplo, pela base de regras da figura 3.5 :

IF temperatura FRIA AND Variação da temperatura MÉDIA THEN Ação de Controle MÉDIA

Suponha que a variação de temperatura tenha sido Média com grau de pertinência $\mu=0.3$ e Baixa com grau de pertinência 0.4 .

Pela base de regras temos:

IF temperatura MORNA AND Variação da temperatura MÉDIA THEN Ação de Controle MÉDIA

IF temperatura MORNA AND Variação da temperatura BAIXA THEN Ação de Controle BAIXA

IF temperatura QUENTE AND Variação da temperatura MÉDIA THEN Ação de Controle BAIXA

IF temperatura QUENTE AND Variação da temperatura BAIXA THEN Ação de Controle MÉDIA

Encontramos então na base de regras que a saída em tensão do aquecedor do forno deve ser Média , Média, Baixa, Baixa. Para cada sub-função resultante, o grau de pertinência é encontrado através intersecção entre os antecedentes. Normalmente é utilizado o mínimo grau de pertinência[Bimal 94][Hasan 94](esta é uma operação de intersecção), entre as funções que a determinaram através da base de regras. No exemplo a primeira sub-função Média para a saída foi determinada de acordo com a primeira expressão IF AND THEN, descrita acima, como os graus de pertinência de seus antecedentes são 0.2 e 0.3, pega-se o menor grau de pertinência e associa-se a sub-função Média que fica com grau de pertinência 0.2 .No caso das demais fica, Média com $\mu = 0.4$, Baixa com $\mu= 0.4$ e Baixa com $\mu = 0.3$.

Para o cálculo da saída precisamos de apenas uma representante de cada sub-função da saída. No exemplo foram

avaliadas duas sub-funções de saída de controle Baixa e duas para a saída Média de controle. Neste caso sendo a mesma sub-função, aplica-se a operação fuzzy de União entre estes conseqüentes similares, normalmente elimina-se uma e conserva-se aquela que possuir o maior grau de pertinência. Então para o cálculo da saída serão escolhidas as sub-funções fuzzy da ação de controle Baixa com $\mu = 0.4$ e Média com $\mu = 0.4$.

A operação fuzzy entre antecedentes(intersecção) e conseqüentes(união) até agora descrita pode ser resumida na expressão abaixo.

$$\mu U(.) = \text{Máximo}(\text{Mínimo}(\mu \text{Temp}(.), \mu \text{VarTemp}(.)))$$

Com as sub-funções de saída de tensão definidas partimos para a parte final do cálculo da ação de controle, denominada defuzzificação.

A defuzzificação consiste no cálculo da saída de controle real, para isto é determinada uma figura como é mostrada na figura 3.6, construída a partir das sub-funções

fuzzy da ação de controle limitada pelos seus respectivos patamares definidos pelos seus graus de pertinência(Média $\mu=0.4$ e Baixa $\mu=0.4$, para o exemplo apresentado).

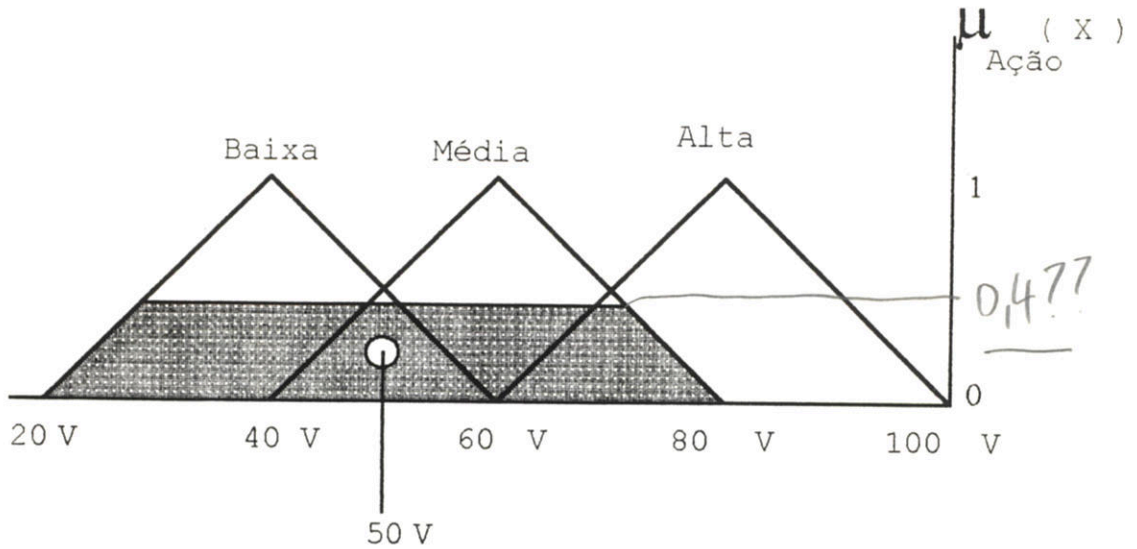


figura 3.6) Área de defuzzificação indicando o Centróide



Para o cálculo da saída de controle pode ser utilizado o método do centróide da figura resultante(em hachurado) que, neste exemplo, produz uma saída de controle de aproximadamente 50 Volts

Toda esta seqüência de cálculos é executada a cada passo de interação de controle, por isso o controle fuzzy necessita de um suporte de hardware rápido em aplicações que requeira uma ação de controle em tempos menores.

3.3 Configuração do Controlador Fuzzy

94]: Numa malha de controle o controlador fuzzy tem a seguinte configuração[Bimal

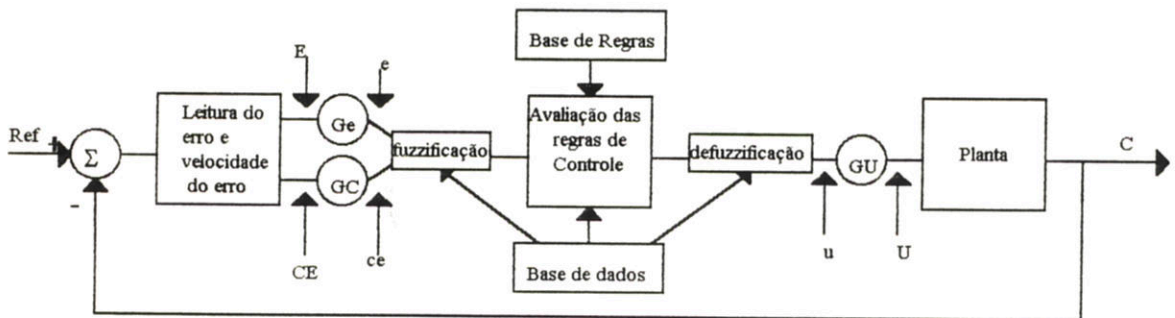


figura 3.7) Representação esquemática de um controlador Fuzzy

A base de dados não mencionada anteriormente, serve para definir os limites das sub-funções de cada variável, base de regras, tipos de operações de intersecção e consequentes e métodos de fuzzificação e defuzzificação.

O bloco descrito na figura 3.7 tem como características a leitura do erro e velocidade do erro, mas dependendo da aplicação a leitura dos dados poderão ser um aumento da população em aplicações biológicas, níveis de pressão em pacientes, em sistemas de diagnósticos médicos, entre outros.

Para cada operação de antecedentes (intersecção, AND) e consequentes (união, OR), podemos alterar o método destas operações de formas ^{adequadas} a virem ficar mais convenientes a nossa aplicação. Porém a alteração tem carácter individual e não existe uma metodologia que comprove que esta ou aquela operação são mais adequadas a certo tipo de situação, cabe a nós munidos de intuição e conhecimento do sistema escolhermos uma delas. _{os projetista,}

As operações a seguir estão descritas em [Cox 94] e são apresentadas na forma em que foram estudadas.

3.3.1 Operações de Intersecção:

Como operações de intersecção podemos apresentar os seguintes tipos de operações fuzzy:

3.3.1.1 Mínimo, também conhecido como mínimo de Zadeh,

$$\mu_{\text{resultante}} = \text{Mínimo}(\mu_A, \mu_B);$$

3.3.1.2 Média

$$\mu_{\text{resultante}} = (\mu_A + \mu_B) / 2 ;$$

3.3.1.3 Média ao quadrado

2

$$\mu_{\text{resultante}} = ((\mu_A + \mu_B) / 2) ;$$

3.3.1.4 Raiz da Média

1/2

$$\mu_{\text{resultante}} = ((\mu_A + \mu_B) / 2) ;$$

3.3.1.5 Produto

$$\mu_{\text{resultante}} = \mu_A * \mu_B;$$

3.3.1.6 Bounded Sum

$$\mu_{\text{resultante}} = \text{Máximo}(0, \mu_A + \mu_B - 1);$$

3.3.1.7 AND de Yager, k é um parâmetro a ser setado.

$$\mu_{\text{resultante}} = 1 - \text{Mínimo}\left(1, \left((1 - \mu_A)^k + (1 - \mu_B)^k \right)^{1/k}\right)$$

3.3.2 Operações de União:

Como operações de união podemos apresentar os seguintes tipos de operações fuzzy:

3.3.2.1 Máximo, também conhecido como máximo de Zadeh

$$\mu_{\text{resultante}} = \text{Máximo}(\mu_A, \mu_B)$$

Média, Média ao quadrado e raiz da média são análogas a intersecção.

3.3.2.2 Produto

$$\mu_{\text{resultante}} = \mu_A + \mu_B - (\mu_A * \mu_B)$$

3.3.2.3 Bounded Sum

$$\mu_{\text{resultante}} = \text{Mínimo}(1, \mu_A + \mu_B)$$

3.3.2.4 OR de Yager, k é um parâmetro a ser setado.

$$\mu_{\text{resultante}} = \text{Mínimo}\left(1, \left(\mu_A^k + \mu_B^k \right)^{1/k}\right)$$

3.3.3 Operações de Defuzzificação

As operações de defuzzificação são calculadas a partir da figura resultante das operações com os conseqüentes, como demonstrado na figura 3.6, portanto para estudarmos os métodos mais utilizados será apresentado somente a figura resultante (em hachurado).

As operações aqui descritas foram estudadas no livro [Cox 94] e são as seguintes:

3.3.3.1 Centróide

Calcula o centro de gravidade da figura, através da equação:

$$U = \frac{\int X \cdot \mu(X) dX}{\int \mu(X) dX}$$

o seu editor de equações ã funciona?

A demonstração gráfica é de acordo com a figura 3.6

3.3.3.2 Média do Máximo

No patamar formado pela sub-função com o maior grau de pertinência, calcula-se o ponto médio neste patamar.

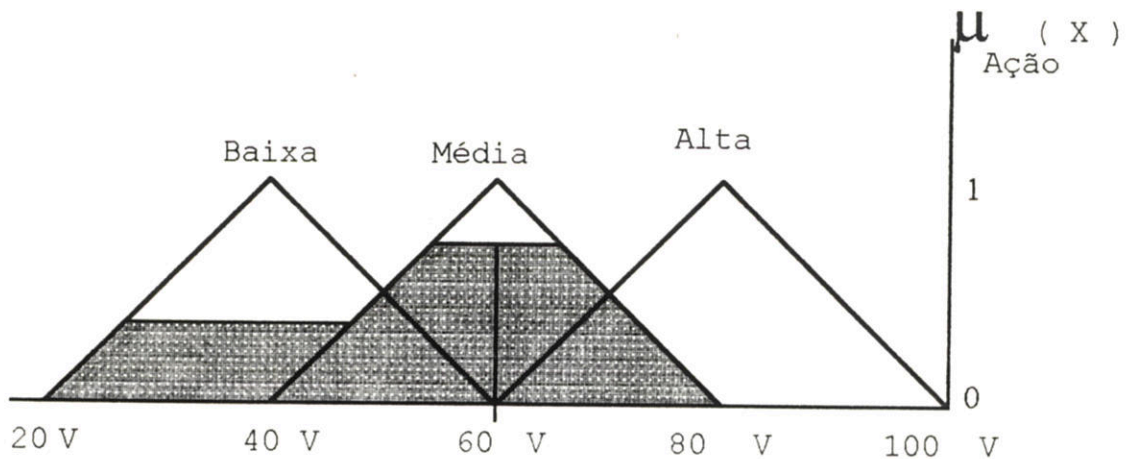


figura 3.8) Cálculo da Defuzzificação pela Média do Máximo

Modificando um pouco o exemplo, observamos na figura 3.8 que o ponto médio coincide com o centro da sub-função Média da ação de Controle, fato este que nem sempre acontece.

3.3.3.3 Máximo a Direita

Da sub-função da ação de controle com maior grau de pertinência calcula-se a saída pelo valor referente do patamar do lado direito

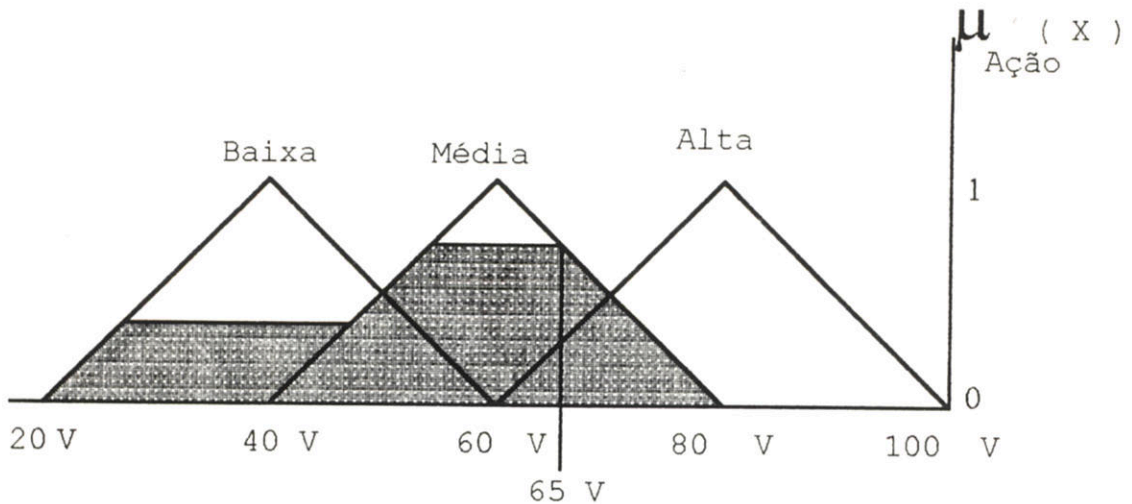


figura 3.9) Cálculo da Defuzzificação pelo Máximo a Direita

Em nosso caso, como descrito na figura 3.9 a ação de controle seria aproximadamente 65 Volts

3.3.3.4 Máximo a esquerda

A saída de controle é calculada de forma análoga ao máximo a direita, é claro que agora ao invés da direita usamos a esquerda.

Pela figura 3.10 a saída de controle será 55 V.

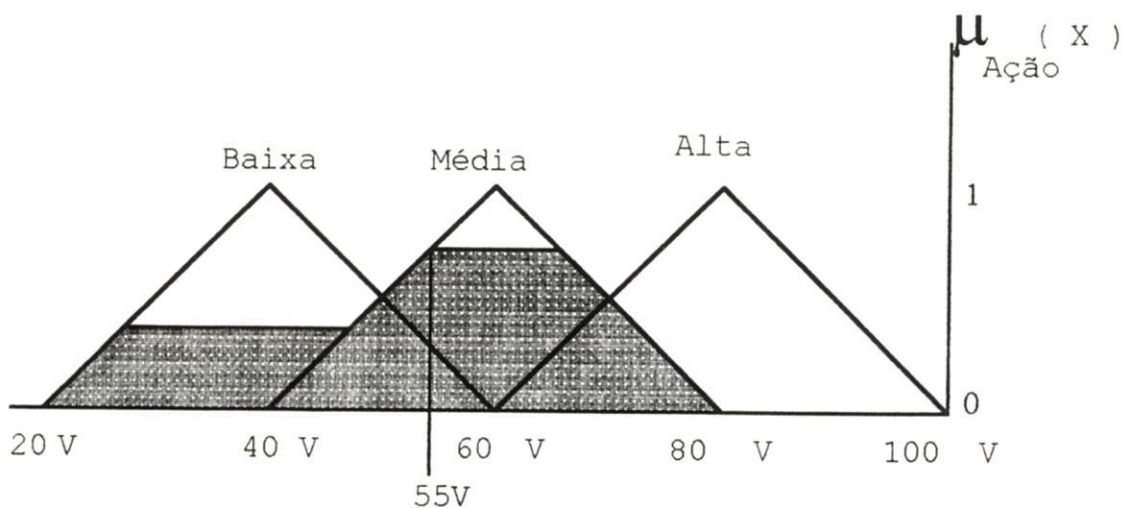


figura 3.10) Cálculo da defuzzificação pelo Máximo a Esquerda

3.3.3.5 Centro do Máximo

Neste caso a ação de controle é calculada pelo centro da sub-função com o maior grau de pertinência.

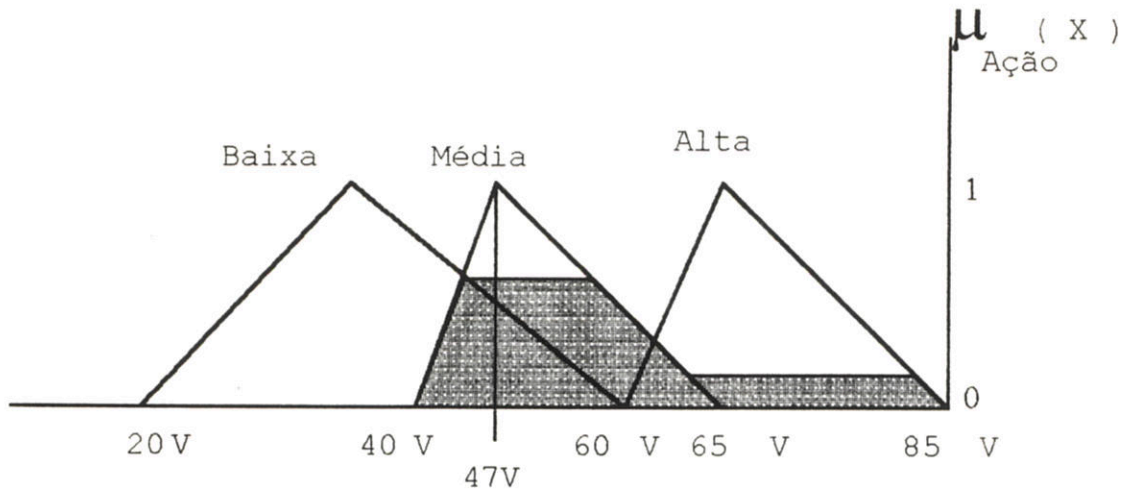


figura 3.11) Cálculo da Defuzzificação pela Centro do Máximo

3.3.3.6 Média do Centro dos Máximos

A saída de controle é calculada pela média dos centros das duas sub-funções fuzzy com maiores graus de pertinência.

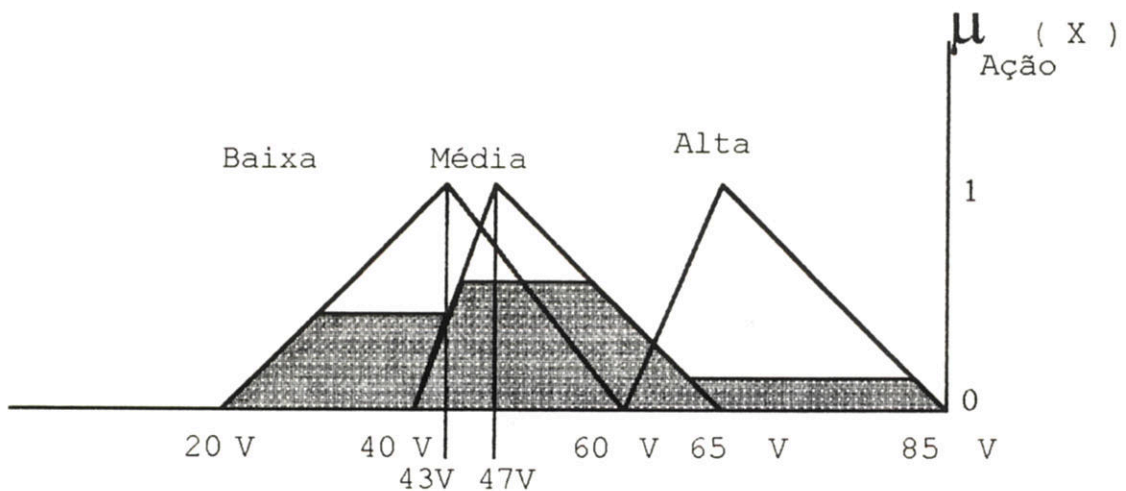


figura 3.12) Cálculo da Defuzzificação pelo Centro dos Máximos

Neste exemplo da figura 3.12 os dois centros são 43 V e 47 V, que resultam numa média igual a 45 V, que será a ação de controle calculada.

3.3.3.7 Centróide dos Máximos

Utilizando a mesma equação aplicada anteriormente para o centróide, aqui calcula-se o centróide da figura formada pelas duas sub-funções com os dois maiores graus de pertinência.

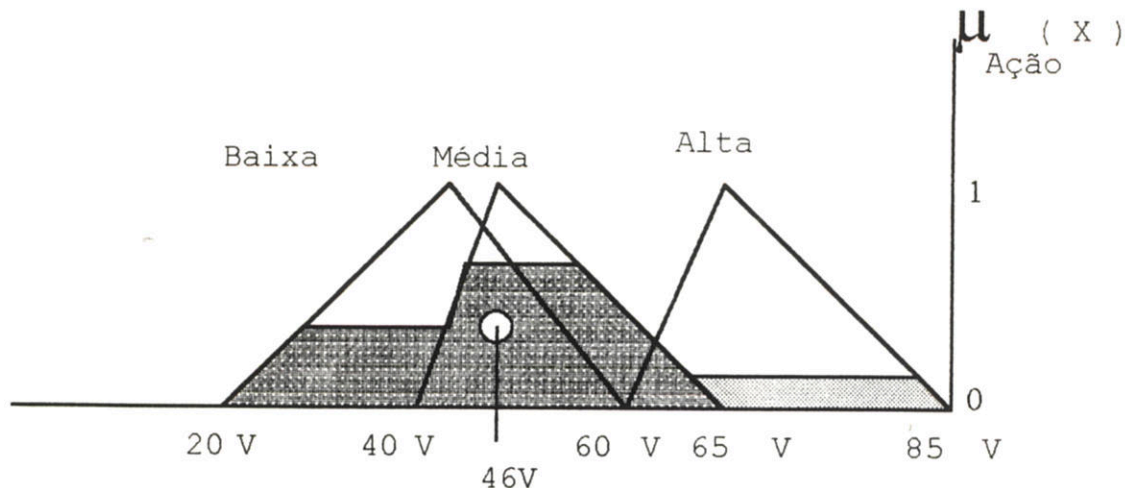


figura 3.13) Cálculo da Defuzzificação pelo Centróide dos Máximos

Neste caso acima as duas sub-funções em hachuriado mais escuro são utilizadas para o cálculo do centróide por terem o maior grau de pertinência.

3.3.3.8 TVFI(True Value Flow Inference) [Apronix 94]

Este método consiste em calcular a ação de controle pelos centros das sub-funções fuzzy da saída e seus respectivos graus de pertinência. É a média ponderada onde a ponderação dos valores dos centros é dado pelo grau de pertinência respectivo.

Do exemplo do forno tínhamos como sub-funções fuzzy:

Ação de Controle Baixa $\mu = 0.3$ e

Ação de Controle Média $\mu = 0.4$

Supondo que o centro da sub-função Baixa seja 25 V e da sub-função Média 44 V então a ação de controle fica:

$$\begin{aligned}
 U &= \frac{0.4 * 44 + 0.3 * 25}{0.4 + 0.3} \\
 &= 35.8571 \text{ Volts}
 \end{aligned}$$

3.3.4 Operações de alteração das vizinhanças(Hedges)

Este tipo de operação consiste em alterar o formato da sub-função fuzzy, de formas a condicioná-la melhor a aplicação desejada.

Esta alteração do formato é produzida pela colocação de um expoente, na função que forma a sub-função fuzzy. Como as sub-funções fuzzy variam entre 0 e 1, não se alterará a imagem da função, entretanto as laterais que eram retas do tipo $y=a*x + b$ formando o triângulo já visto agora se tornam curvas de acordo com o expoente (k) selecionado, tornando a equação para a forma:

$$y = (a*x + b)^k$$

Estas curvaturas são identificadas por nomes, de acordo com a variação de abertura, como apresentado nas figuras 3.14 e 3.15

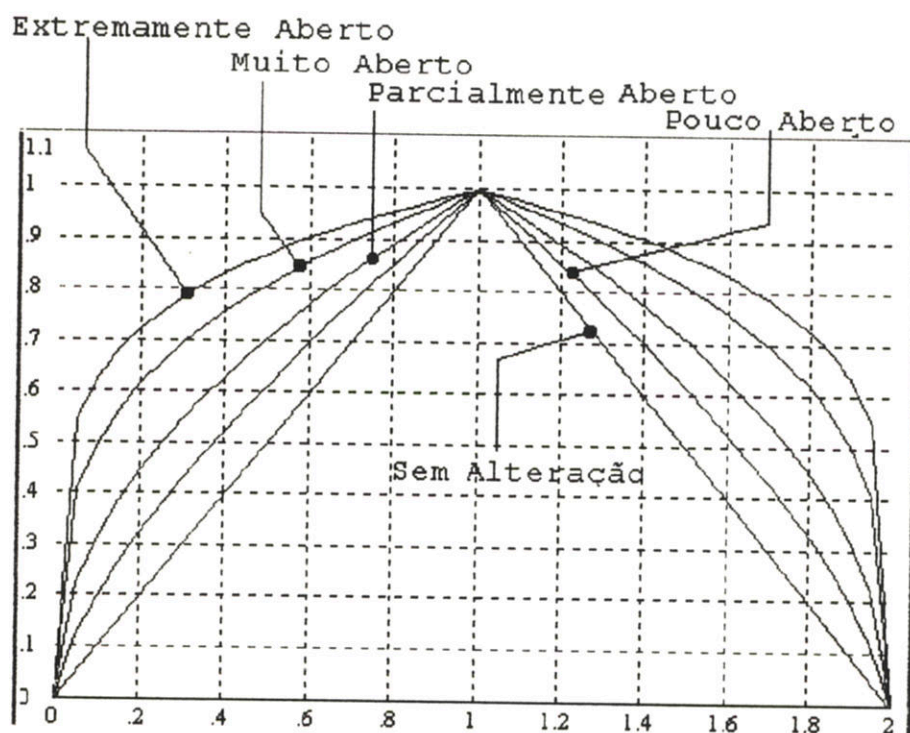


figura 3.14) Alteração do expoente $0 < k < 1$

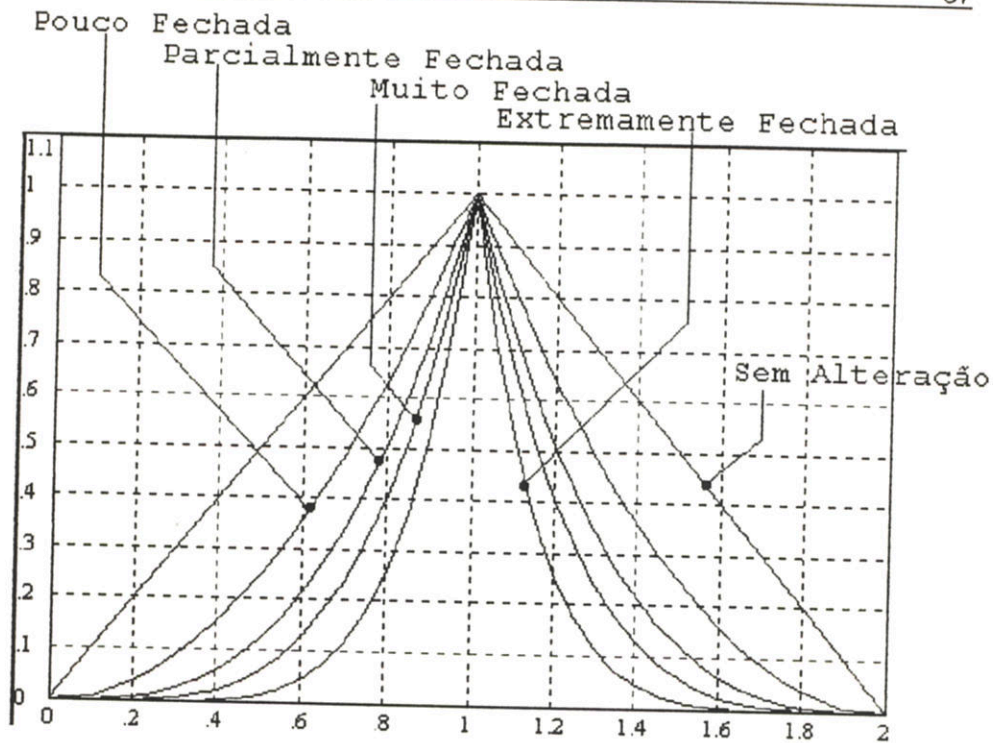


figura 3.15) Alteração do expoente $k > 1$

A operação de hedge pode ser utilizado num processo de melhoria da precisão do controlador fuzzy, aumento ou diminuição da sensibilidade do controlador a perturbações e variações de referência, e alterações na dinâmica de controle.

3.4 Conclusão

Os controladores fuzzy são de fácil entendimento, mas é preciso um sentimento, que só é captado quando estudado bem detalhadamente. Neste capítulo ~~foi~~ ^{foram} apresentados os diferentes elementos que fazem do controlador fuzzy, uma promissora linha de pesquisa em processos não-lineares e multivariáveis.

A modelagem e implementação do controlador fuzzy pode variar de ~~uma~~ ^{problema} pessoa para ~~suas~~ ^{problema} necessidades.

No próximo capítulo, será apresentado um estudo feito sobre as pesquisas de concepção e modelagem de controladores fuzzy.

Capítulo IV

Metodologias de Concepção e Implementação de Controladores Fuzzy

4.1 Introdução

Ao se dominar os conceitos básicos da lógica fuzzy e os elementos do controlador fuzzy, se faz necessário agora uma apresentação dos métodos mais utilizados para a concepção e implementação dos mesmos.

Neste capítulo estarão indicados os trabalhos mais importantes no campo da implementação de controladores fuzzy, apresentando os conceitos básicos utilizados em cada metodologia, indicando a fonte de pesquisa estudada.

4.2 A aplicabilidade de controladores Fuzzy frente ^{aos} controladores clássicos

Ao necessitarmos de um controlador para atuar em um determinado processo, necessariamente precisaremos de um ~~back-ground~~, ~~o qual possamos~~ entender, com uma boa precisão, a dinâmica e limitações associadas ao mesmo. Este tipo de conhecimento também se estenderá a análise de sensores e atuadores utilizados na implementação das malhas de controle.

Podemos observar isto através de figura 4.1 [Apronix 94], onde ~~é~~ ^{há} apresentado os passos para implementação de um controle fuzzy e ~~para~~ ^{de} um controle clássico.

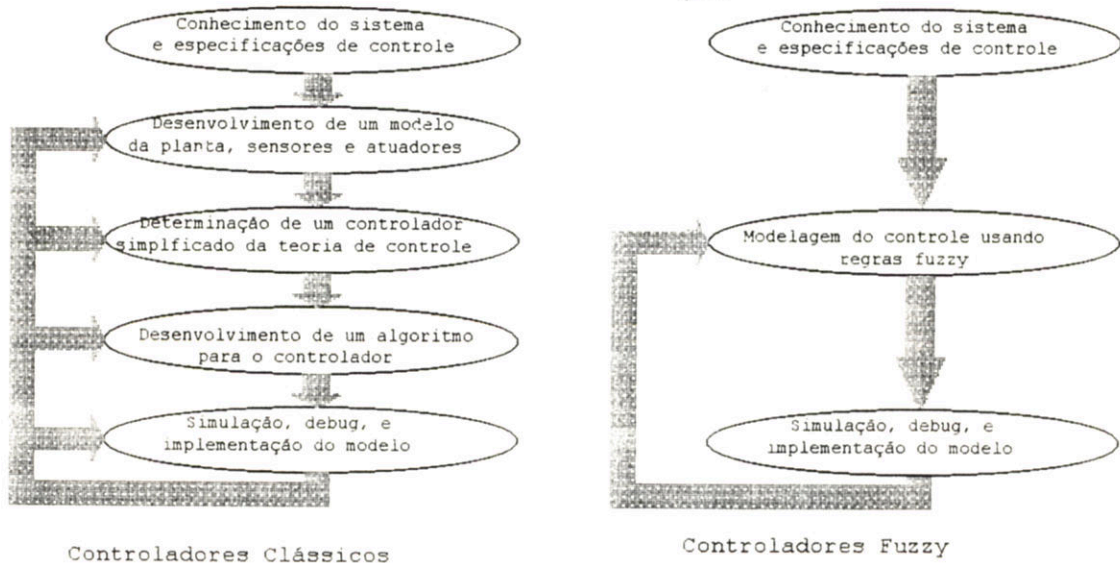


figura 4.1) Passos de implementação de um controle fuzzy e clássico

Todas estas características citadas não são ainda suficientes, ainda faltaria a análise linear de todo o sistema de controle, nos seus devidos pontos de equilíbrio. Esta tarefa pode em certos casos dificultar bastante a implementação de um controlador clássico como um PID, gerando atrasos nos cronogramas e uma série de limitações que terão que ser impostas a este controlador, para que seus objetivos no controle de malha fechada sejam alcançados.

Ao contrário da técnica utilizada na implementação de controladores típicos, o controlador fuzzy tem sua implementação baseada na intuição e conhecimento por experiências do processo onde será conectado.

Este conhecimento aproximado tem como vantagens a diminuição da carga matemática necessária para implementação e execução da malha de controle, e a não necessidade de se avaliar com alto nível de detalhamento os sensores, atuadores e sistemas de auxílio numa planta de controle.

Isto por que, como já foi dito nos capítulos anteriores, a lógica fuzzy vem como uma ferramenta destinada a representação de incertezas e imprecisões de um sistema.

Podemos então concluir que este controlador possui algumas características como abaixo:

- *-> Representação de incertezas
- *-> Tratamento por inferências das incertezas
- *-> Independência da dinâmica dos sistemas
- *-> Alto grau de processamento matemático

O preço que se paga pela implementação de um controlador fuzzy num processo, é a da necessidade de uma base de hardware tal que a velocidade de processamento seja compatível com as necessidades de cálculo interno, pois para um cálculo da saída de controle pode-se perder um tempo consideravelmente longo, relativo ao processo, devido a operações como defuzzificação, por exemplo. Por isso as primeiras aplicações de controladores fuzzy foram em sistemas de controle de temperatura, por estes serem lentos e não necessitarem de grandes velocidades na dinâmica de controle.

Outra característica que dificulta a implementação do controlador fuzzy é ^{a necessidade} o conhecimento do processo onde será conectado, por ~~base de~~ um operador que tenha experiência. Um bom dimensionamento do controlador fuzzy vai depender do quanto se conhece, se tem sentimento, sobre a malha de controle.

Mas hoje em dia com o avanço da tecnologia de microprocessadores, e a consequente diminuição dos tempos de processamento, pode-se aplicar o controlador fuzzy em sistemas onde se requer uma velocidade de cálculo grande. Sem esquecer que existem linhas de pesquisa que são direcionadas para amenizar a carga de conhecimento do processo, criando meios de ajustes automáticos dos parâmetros e definição das operações fuzzy

4.3 Metodologias de implementação de controladores fuzzy

Os controladores fuzzy não possuem uma metodologia sólida para sua implementação. O que existe são formas de se melhorar a intuição do projetista, no momento de ajustar os parâmetros do controlador.

Este fato pode ser um problema sério se não temos experiência sobre o sistema onde pretendemos implantar um controlador fuzzy.

Abaixo estão descritas as linhas de pesquisas atuais para a concepção de controladores fuzzy, dando destaque para a análise de comparação com o controlador PI e o método de determinação das funções de pertinência, pela relação entrada saída de uma não-linearidade.

4.3.1 Algoritmos Genéticos

A técnica de algoritmos genéticos foi desenvolvida com o objetivo da geração de número aleatórios, baseados em condições pré-definidas. Para isto utiliza conceitos de genética(cromossomos, população, etc) para executar suas tarefas.

Em controladores fuzzy as pesquisas de aplicação de algoritmos genéticos, estão focados nos seguintes tópicos:

- * Determinação de base de regras
- * Ajustes de limites das sub-funções
- * Otimização de parâmetros do controlador Fuzzy

As pesquisas podem ser encontradas em artigos do IEEE nas referências bibliográficas[Homaifar 95] [Ishibuchi 95] [Pernel 95].

4.3.2 Redes neuro-fuzzy

Dentre as várias aplicações de redes neurais, uma das que se destaca é a aplicação em controle de processos. O problema reside no treinamento de rede, que influencia na dinâmica da ação de controle.

Para melhorar o treinamento da rede é introduzido um controlador fuzzy o qual, antes da rede entrar em funcionamento no controle do sistema, ~~o mesmo~~ auxilia no treinamento e determinação das malhas da rede [Zhang 95] [Sharaf 94] [Chiaberge 95].

4.3.3 Equivalência do Controlador Fuzzy com o PI

Esta pesquisa ^{foi desenvolvida por} foi executada num artigo do IEEE [Galichet 95], e tem como objetivo representar um controlador fuzzy em termos de controlador PI.

Nesta pesquisa um conjunto de parâmetros de um controlador fuzzy é determinado, fixando estes como a base do estudo da equivalência.

O controlador fuzzy, deverá ter estas características:

*As sub-funções de entrada são triangulares, regularmente distribuídas, com uma única sobreposição entre as mesmas;

*As sub-funções de saída são simétricas e regularmente distribuídas;

*Base de Regras são aditivas, ou seja, há uma ordenação crescente na distribuição das sub-funções fuzzy, na base de regras em relação as entradas;

*Defuzzificação pelo método do centróide;

*Operadores de antecedentes e conseqüentes são mínimo e máximo respectivamente.

Se um controlador fuzzy tem ^{as} estas características acima, então a ação do controle é precisamente equivalente a um controlador PI. (Quê controlador PI?)

Note que para se encontrar uma equivalência entre fuzzy e um controlador PI, é necessário restringir de forma significativa os ajustes do controlador fuzzy. Podemos então destacar a liberdade que se tem para ajustar os parâmetros do controlador fuzzy, além da requerida para um PI, e portanto há a possibilidade de se melhorar mais as características dinâmicas da malha de controle.

4.3.4 Implementação de um Controlador Fuzzy baseado na relação de Não-Linearidades

Esta metodologia se baseia na função gráfica que relaciona a saída desejada do controlador, em relação a entrada do mesmo.

Isto pode ser exemplificado pela figura 4.2[Aptronix 94].

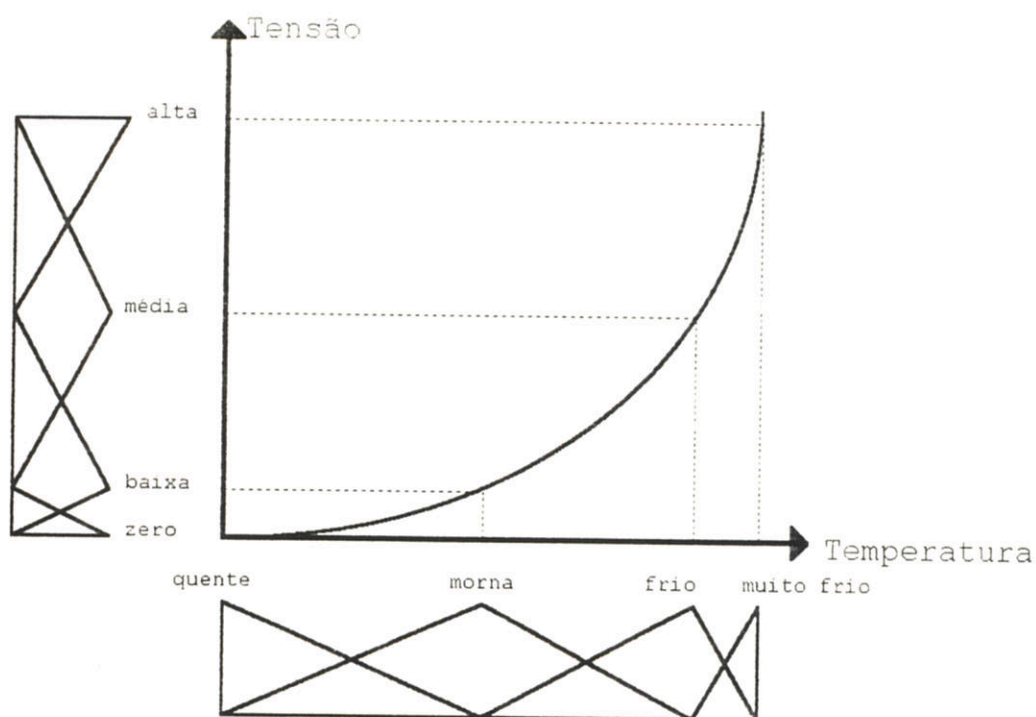


figura 4.2) Representação das não-linearidades para especificação de um controlador fuzzy

Observe que tendo-se a relação entrada saída desejada do controlador, pode-se definir com certa precisão como estarão distribuídas as sub-funções fuzzy da entrada, em relação as sub-funções da saída, assim como define-se a base de regras do controlador fuzzy. Para a figura acima a base de regras da relação entre a temperatura de um forno e a saída em tensão de um aquecedor fica:

IF Temperatura *Muito Fria* **THEN** Tensão *Alta*;

IF Temperatura *Fria* **THEN** Tensão *Média*;

IF Temperatura *Morna* **THEN** Tensão *Baixa*;

IF Temperatura *Quente* **THEN** Tensão *Zero*.

Isto é feito para cada variável de entrada do controlador que se relacione com outra ou outra. A intuição entra no momento de se definir as operações de inferência dos antecedentes, conseqüentemente, defuzzificação e base de regras.

4.4 Software FIDE [Apronix 94]

Um dos objetivos deste projeto era o de se estudar e aplicar o controlador fuzzy gerado pelo software FIDE (Fuzzy Inference Development Environment) na máquina síncrona.

Este software tem a característica de gerar em código C, um programa de controle fuzzy, a partir de uma sintaxe própria e de fácil entendimento.

Também pode-se criar em código C um processo, na forma de um bloco, com uma sintaxe de programação específica em relação ao FIDE, e ~~conecta-lo~~ ^{conectá-lo} ao controlador fuzzy do FIDE de modo a se fazer simulações.

Possui ferramentas de análise de limites da ação de controle, simulação passo-a-passo, construção de base de regras e modificação do formato triangular das sub-funções, através das técnicas dos hedges, que pode ser aplicado em trechos dos lados das sub-funções fuzzy.

Neste projeto esta ferramenta foi apenas estudada e avaliada sua performance. Com isto pode-se apresentar algumas características bastante vantajosas:

- * Boa apresentação gráfica(Windows);
- * Facilidade de ajustes;
- * Baixo tempo para a implementação de um controlador fuzzy;
- * Possibilidade de criação de um bloco em linguagem C com o controle fuzzy;
- * Possibilidade de inserção de blocos em linguagem C, para efeito de simulação direta no controlador fuzzy do FIDE.

Apesar destas vantagens, este software apresentou desvantagens percebidas por simulações.

A desvantagem mais relevante é o fato que as derivadas das curvas calculadas pelo software não são calculadas a tempos constantes, acarretando o aparecimento de degraus nas respostas gráficas. Este fato inviabilizou um estudo mais aprofundado quanto a simulações do PSS da máquina síncrona, pois o mesmo sistema como já dito é de sétima ordem, somado ao processo de medição e controle de velocidade, fica de nona ordem.

4.5 Conclusão

Neste capítulo foi apresentado um pouco sobre o que se está fazendo, para se melhorar as técnicas de implementação de controladores fuzzy. Uma miscelânea de métodos e ferramentas mais utilizadas ultimamente.

Poderíamos ter apresentado todos os trabalhos já realizados sobre metodologias de concepção e implementação de controladores fuzzy, entretanto, seria escrito um livro e não um capítulo de um relatório. Para maior conhecimento de quem desejar, basta estudar os artigos do IEEE e as literaturas apresentadas nas bibliografias indicadas na parte final deste relatório.

Apesar de ser ainda bastante recente a utilização do controle fuzzy, esta técnica de controle promete muito para o mercado de sistemas de controle, pois cada vez mais apresentados bons resultados, o que viabiliza muito seu desenvolvimento. A aceitação por completo somente se dará no dia em que algum pesquisador conseguir comprovar, matematicamente, a estabilidade do controlador fuzzy, pois mesmo sabendo-se muito, é di-

fácil afirmar que uma aplicação neste ou naquele processo será estável, ou com a dinâmica já pré-determinada.

Capítulo V

Placas de Aquisição

5.1 Introdução

No projeto do controlador fuzzy, foi necessário fazer a conexão entre dois microcomputadores, no primeiro ficaria o software SSD funcionando como máquina síncrona, no segundo o programa FuzSsd.C, onde estão implementadas as rotinas de controle fuzzy.

Neste processo foi necessário tratar três dados: erro, velocidade de erro e ação de controle.

Para implementar a conexão entre os micro-computadores foram utilizadas três placas diferentes, uma ADA 10 para receber dados da ação de controle no processo da máquina síncrona, uma DAX para enviar os dados de erro e velocidade do erro do processo para o controlador fuzzy e uma ADA 16 que faz interface com o programa FuzSsd.C, adquirindo os dados do processo e colocando na saída a ação de controle fuzzy calculada.

A utilização das duas placas no processo se justifica pelo fato da placa ADA 10 não possuir as duas saídas necessárias e esta aplicação, portanto se fez necessário a utilização da placa DAX, pois esta última possui 4 saídas.

O capítulo a seguir tem a objetividade de apresentar as características principais de cada placa utilizada, no caso, as placas ADA 16, ADA 10 e DAX, destacando para cada uma o seu papel na utilização na malha de controle do PSS fuzzy. Ao final do capítulo é apresentado um esquemático das conexões entre os micros identificando cada placa os pontos de conexão.

5.2 Placa ADA 10

A placa ADA 10 é um projeto antigo da REIVAX, e na versão do software SSD utilizada, funciona como elemento de aquisição dos dados da ação de controle calculada pelo controlador fuzzy, sendo então necessária para este projeto.

A placa ADA 10 é um sistema compatível com PC /XT/AT, portanto de fácil utilização, pois para utilizá-la basta conectar ao microcomputador e indicar no software SSD o canal para enviar dados, e qual canal deseja-se receber dados.

Possui 8 canais de entrada analógica de 12 ou 14 bits com seleção de ganho programável de +0.1 volt, +1 volt e +- 10 volt. Podem ser adicionados circuitos sample-

hold nos 8 canais de entrada, para sincronização de leitura em mais de um canal. Um canal de saída analógica de 12 bits possui atenuação programável de ± 10 , ± 5 , ± 2.5 , 0 a 10 e 0 a 5 volts.

As 8 entradas digitais, tem níveis de até 40 volts, e uma entrada de medição de frequência, enquanto as 8 saídas digitais com níveis de 5 ou 12 volts são selecionáveis por estape.

Os temporizadores, em número de 2, são programáveis de 16 e 32 bits e são utilizados como geradores de interrupção, como geradores de frequência ou juntos aos controladores DMA incluídos na placa.

Possui controladores DMA de entrada e saída analógica. No controlador DMA de entrada analógica, existe a possibilidade de programação da sequência de aquisições para um ou mais canais com comutação automática de ganhos para cada canal. O tempo mínimo entre amostras é de 15 microsegundos para placas de 14 bits e 12.5 segundos para placas de 12 bits. Para o controlador DMA de saída analógica, pode-se programar uma sequência de conversões fixa ou cíclica. O tempo mínimo entre conversões é de 10 microsegundos.

No projeto do controlador fuzzy a placa ADA 10 foi utilizada para receber a ação de controle fuzzy, para o SSD, enviada pela ADA 16, conectada do controlador fuzzy do programa FuzSsd.C.

5.3 Placa DAX

Também compatível com IBM PC/XT/AT, tem sua maior utilização na saída de dados no programa SSD da REIVAX. A placa DAX tem um único propósito de envio de dados através de até 6 conversores (saídas) independentes de 12 bits atualizadas simultaneamente e protegidas por OPA de precisão soqueteados.

O tempo de conversão é menor que 9 microsegundos para cada canal.

Esta placa tem a possibilidade de programação de referência de tensão e atenuação.

A precisão (LSB) da placa pode ser melhorada através de ajustes.

As 8 faixas de tensão são divididas em duas configurações como segue abaixo:

Bipolar

V ref		10 V	7.5 V	5 V	2.5 V
	Valor Escrito	Tensão de Saída			
Saída	0	+10.000	+7.500	+5.000	+2.500
	4095	-9.995	-7.496	-4.992	-2.499

LSB		4.88 mV	3.66 mV	2.44 mV	1.2 mV
-----	--	---------	---------	---------	--------

Unipolar

V ref		10 V	7.5 V	5 V	2.5 V
	Valor Escrito	Tensão de Saída			
Saída	0	0	0	0	0
	4095	9.999	7.498	4.999	2.498
LSB		2.44 mV	1.83 mV	1.22 mV	0.61 mV

A placa DAX ainda possui duas saídas digitais independentes SD1 e SD2 programáveis simultaneamente. Estas são bufferizadas(drenam até 40 mA em nível baixo e podem fornecer até 40 mA em nível alto) em soquete. Possui ainda timer programável e gerador de frequências.

No projeto do controlador fuzzy, esta placa foi utilizada para fornecer as saídas de dados do processo da máquina síncrona (erro e velocidade do erro) para o controlador fuzzy no programa FuzSsd.c comunicando com a placa ADA 16.

5.4 Placa ADA 16

A placa ADA 16 é um produto da REIVAX compatível com Computadores IBM PC/XT/AT.

Esta placa foi projetada com tecnologia de circuitos integrados de última geração, mais técnicas de lógicas programáveis. O objetivo do projeto é o de obter-se um produto compatível com os concorrentes do mercado.

Configuração da placa ADA 16 pode ser feita pelo usuário, onde o mesmo pode ajustar ganhos e a forma de transmissão entre placas, ou pela fábrica na programação dos tipo de saídas, no caso diferenciais ou não, e o número de saídas, que poderão ser 2, 4 ou 8 saídas programáveis.

A placa ADA 16 possui 16 entradas analógicas simples ou 8 entradas analógicas diferenciáveis. Os ganhos de cada canal de entrada pode ser programados em 1 V/V, 10V/V, 100V/V e 1000V/V, setados independentemente para cada canal. As amplitudes dos sinais de entrada são de até +-10 Volts.

As saídas analógicas tem 12 bits de precisão com faixas programáveis de forma unipolar bipolar e deslocados, limitados entre +-10 Volts.

Ainda possui 2 temporizadores programáveis de 16 e 32 bits que podem gerar uma saída de frequência programável em Hz e MHz.

Funcionando como suporte de hardware do computador onde esta instalado o programa FuzSsd.C, esta placa tem a função neste projeto de adquirir os dados de erro e velocidade do erro, provenientes do processo da máquina síncrona(placa DAX) e colocar na sua saída (canal 1) a ação de controle para ser utilizada pelo processo da máquina síncrona, ação será esta que será lida pela placa ADA 10 conectada no processo.

5.5 Conexão das Placas

Na figura 5.1 é apresentado um esquemático da conexão dos cabos entre as placas. Também é indicado as referências e limites de tensões em cada ligação.

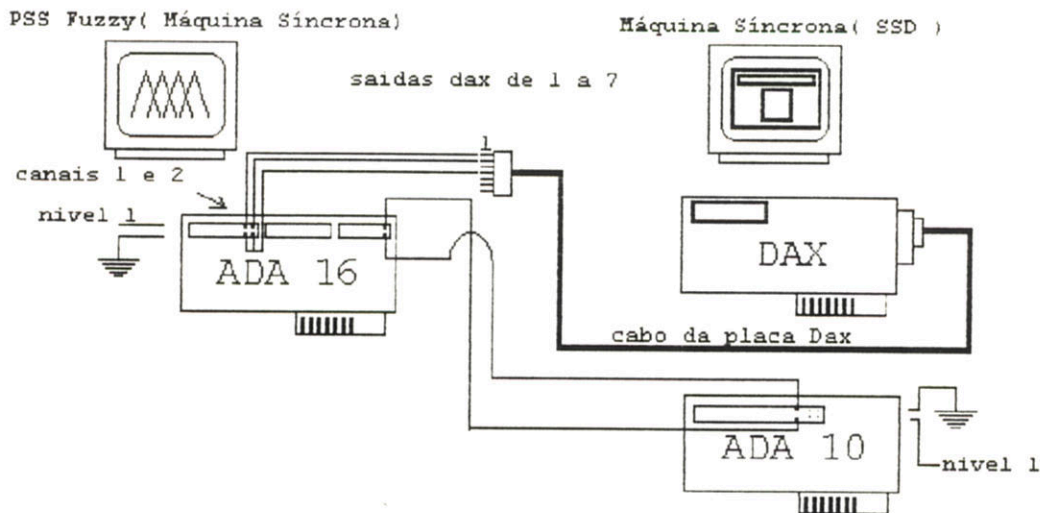


figura 5.1) Esquema de conexões das placas

5.6 Conclusão

Neste capítulo foi apresentado as características referentes a conexão do processo de geração de potência(máquina síncrona) e o controlador fuzzy, do ponto de vista das placas.

Foi apresentado os elementos relevantes em cada placa assim como as características pecualires de cada placa.

Esta conexão poderia ser melhorada se fosse utilizado apenas duas placas de mesmo tipo, pois a programação e configuração da malha de controle seria mais facilitada, entretanto não foi possível esta alternativa, resultando na configuração apresentada a qual foi utilizada neste projeto.

Capítulo VI

O Software FuzSsd

6.1 Introdução

O desenvolvimento de controladores fuzzy tem se intensificado nos últimos anos, por isso a atualização e a criação de linhas de pesquisa dentro deste campo se torna um fato inevitável dentro das empresas de controle.

Esta questão foi estudada dentro da REIVAX Automação e Controle, detentora de uma fatia do mercado de sistemas de potência, onde foi decidido iniciar uma pesquisa de controladores fuzzy já que a mesma esta bem avançada, dado os artigos do IEEE.

Dentre as muitas implementações de controladores fuzzy apresentadas nestes artigos, há um número bastante considerável de aplicações em estabilizadores e potência para máquinas síncrona, que é o objeto de estudo deste projeto final de projeto.

O programa FuzSsd vem como uma ferramenta de auxílio aos estudos, avaliação da capacidade e aspectos dinâmicos do controle fuzzy aplicado como PSS em máquina síncrona. Tendo ainda como característica a possibilidade de aplicação em outros processos, dado que a interface com o meio é de simples alteração.

6.2 Especificações do Programa

Como especificações podemos apresentar:

*Reusabilidade;

*Possibilidade de acesso a todas as operações fuzzy estudadas nos capítulos anteriores;

*Criação de uma Base de Regras;

*Possibilidade de conexão com outro micro através da placa ADA 16 para posterior simulação;

*Fácil interface com o usuário;

*Fácil ajustes do controlador;

*Ser utilizado como ferramenta de análise e estudos de comportamento de controladores baseados em lógica fuzzy.

6.3 Funcionamento do FuzSsd

Implementado em Microsoft C o FuzSsd é um programa estruturado e dividido em três arquivos, onde foram implementadas todas as funções fuzzy de antecedentes, conseqüentes, fuzzificação, defuzzificação e base de regras de acordo com a teoria estudada[Cox 94] e nos artigos do IEEE[Bimal 95][Hasan 94].

Possui duas entradas de controle e uma saída calculada por lógica fuzzy.

Tem a denominação FuzSsd por ser um controlador Fuzzy atuando num processo definido no software Ssd

O FuzSsd tem seu funcionamento baseado em menus , que através de opções facilitam o ajuste e a execução do controlador.

Internamente este controlador possui uma estrutura definida como SFF(Sub-Função Fuzzy) a qual possui os parâmetros que formam uma sub-função fuzzy.

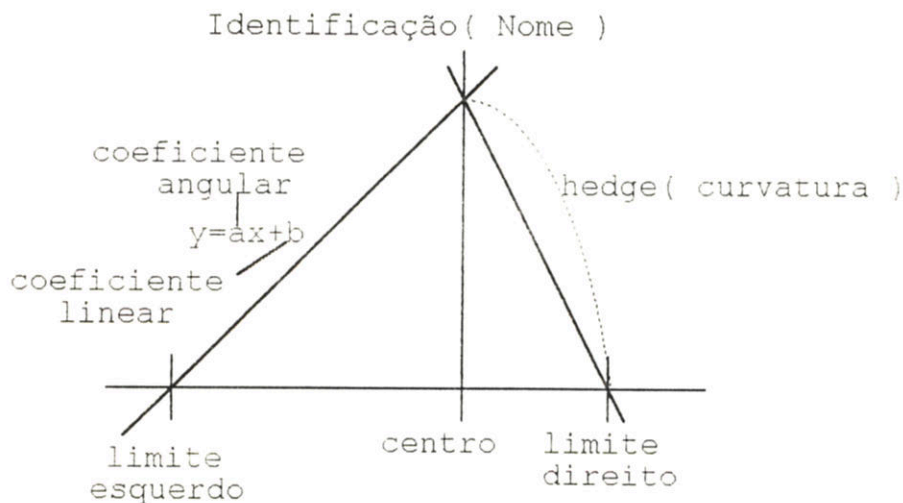


figura 6.1) Parâmetros de uma sub-função fuzzy

Pela figura 6.1, a estrutura SFF tem os seguintes parâmetros

*Limite esquerdo;

*Limite Direito;

- *Ponto Central;
- *Coeficiente angular da função ascendente;
- *Coeficiente linear da função ascendente;
- *Coeficiente angular da função decrescente;
- *Coeficiente linear da função decrescente;
- *Hedge;
- *Identificador.

Para cada variável do controlador, no caso erro, velocidade do erro e ação de controle, tem-se um vetor e 7 sub-funções fuzzy, identificadas respectivamente como :

NB => Negativa Grande

NM => Negativa Médio

PS => Negativa Pequena

Ze => Zero

PS => Positiva Pequena

PM => Positiva Média

PB => Positiva Grande

Os parâmetros destas sub-funções podem ser alterados pelo usuário, de acordo com regras a serem apresentadas mais adiante.

Pela figura 6.2 temos a descrição da estrutura geral do programa.

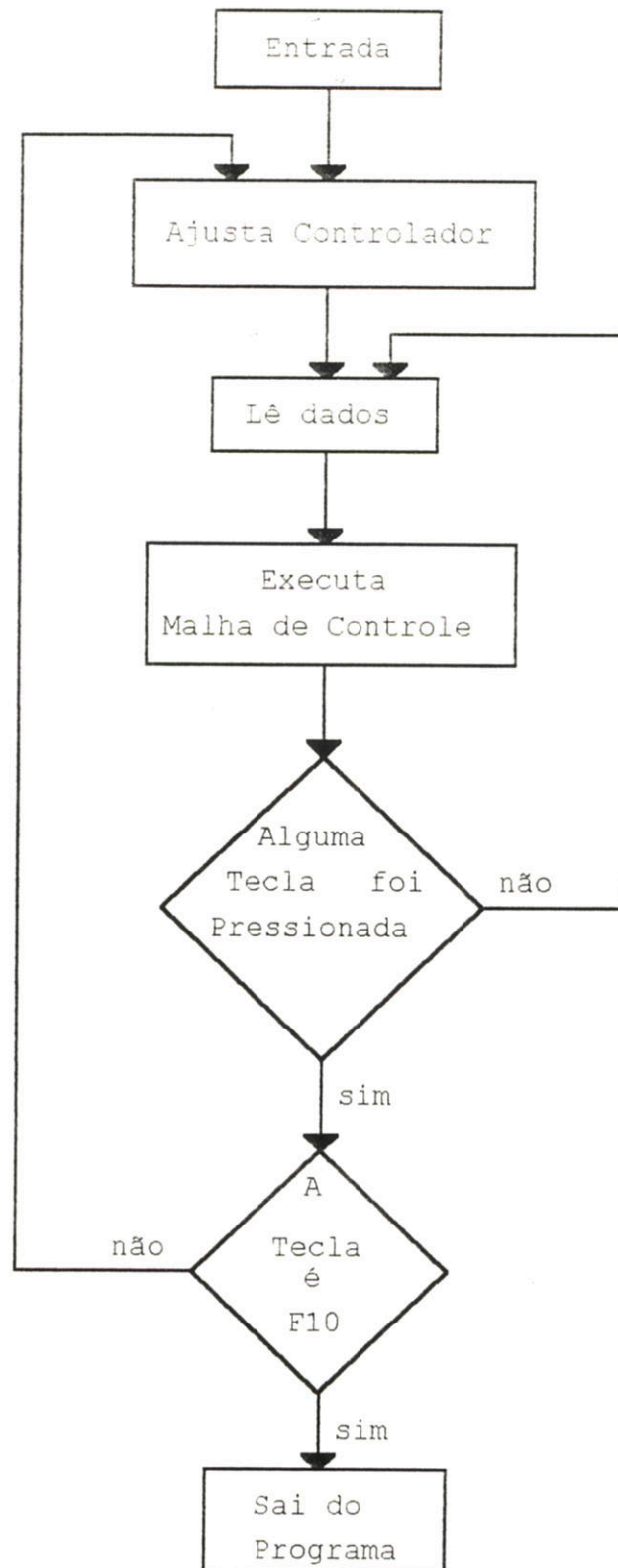


figura 6.2) Estrutura geral do programa FuzSsd

O FuzSsd foi construído de forma tal que a parte referente a comunicação de dados entre placas fosse independente do cálculo de ação de controle fuzzy.

Isto foi possível por uma característica peculiar do controlador fuzzy, para cada par de entradas existe somente uma saída. Esta característica em controladores como o PID não se observa, pois a saída do mesmo é determinada por equações de segunda ordem, impondo um cálculo da ação de controle em função de valores a dois períodos de amostragem atrás.

6.3.1 Utilizando o FuzSsd

A chamada no ambiente DOS é feita pelo por <c:\FUZZY3> e logo é apresentada uma tela como segue na figura 6.3 .

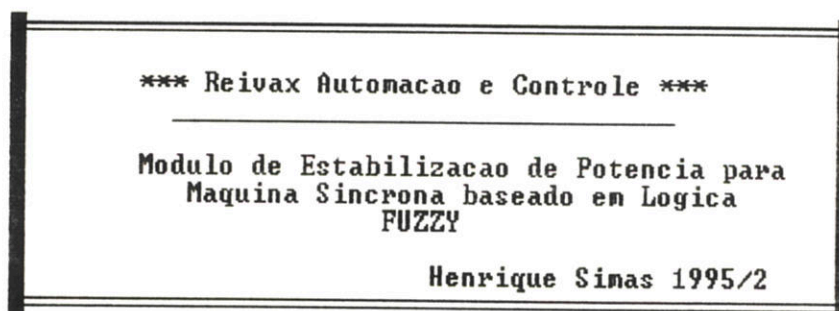


figura 6.3) Tela inicial do FuzSsd

Após teclar qualquer tecla é apresentado o menu principal, veja figura 6.4, onde se pode navegar pelo controlador e ajusta-lo.

Através das teclas de setas pode-se escolher qual item do controlador fuzzy deseja-se alterar. Quando o cursor estiver indicando a opção então deve-se teclar <ENTER>.

Também as teclas F1, F2, F3, F4, F5, e F10 são sensibilizadas com o intuito de se melhorar a interface com o usuário. A explicação mais detalhada sobre estas teclas será dada mais adiante

6.3.1.1 Menu Principal

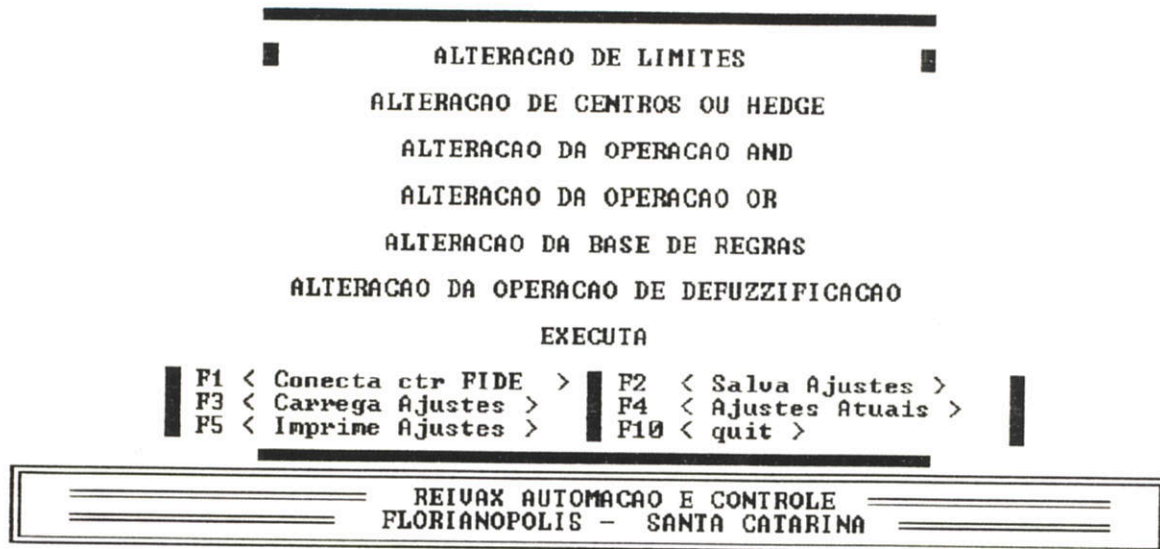


figura 6.4) Tela do menu principal

6.3.1.1.1 Alteração de Limites

Ao se entrar na opção de alteração de limites, aparecerá de imediato o menu da figura 6.5, o qual será utilizado para se escolher qual das variáveis terão seus limites das sub-funções alterados.

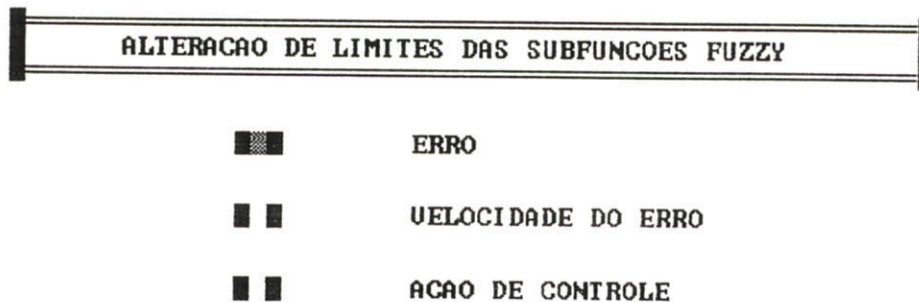


figura 6.5) Tela do menu de alteração de limites

Ao se entrar numa opção do menu de limite uma tela como a da figura 6.6 é apresentada.

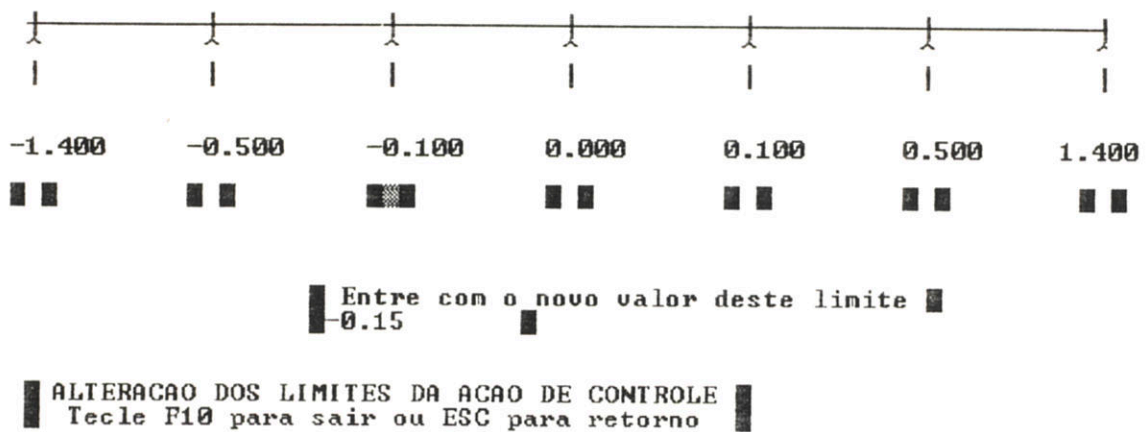
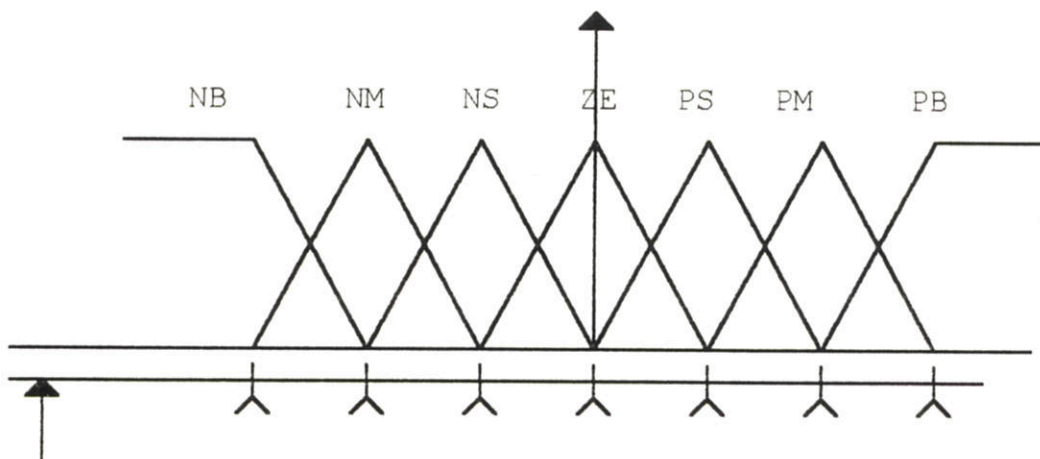


figura 6.6) Tela de alteração de limites

Esta tela tem a característica de alterar um vetor de limites para cada variável do controlador, na figura 6.6 a escolha de alteração foi da variável Ação de Controle. Estes limites, que podem ser alterados são mostrados na figura 6.7 abaixo.



reta que aparece na tela de alteração de limites

figura 6.7) Relação entre os pontos dos limites apresentados no programa e o correspondente de um conjunto de sub-funções fuzzy

Pode-se observar que esta configuração de limites impõe uma única sobreposição entre duas sub-funções.

Esta restrição na disposição dos limites foi implementada para que na fuzzificação, encontre-se apenas duas sub-funções para cada medição de uma variável.

A ação de controle não pode ser ilimitada como descrito na figura 6.7, para limita-la são definidos dois limites auxiliares nos extremos, calculados de forma que o limite mais próximo seja médio, como descrito na figura 6.8.

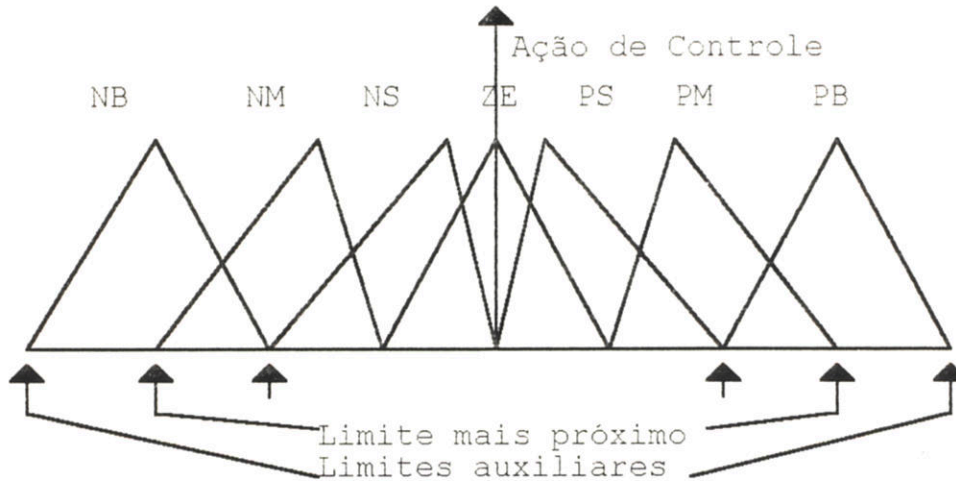


figura 6.8) Definição dos limites auxiliares da variável da ação de controle

Para alterar um limite basta colocar o cursor sobre o limite desejado e teclar ENTER. Será escrito então "Entre com o novo valor deste limite" e então o mesmo é digitado. Para sair deste menu e retornar ao principal é teclado ESC, para sair do programa e retornar ao DOS é teclado F10.

É bom advertir para se tomar cuidado em ajustar os limites em ordem do menor para o maior, da esquerda para direita respectivamente. A troca da ordem pode acarretar problemas de overflow e cálculo de saídas de controle absurdas.

6.3.1.1.2 Alteração de Centros e Hedges

Esta opção do menu principal é responsável pela alteração dos centros ou hedges das sub-funções fuzzy das variáveis do controlador.

Ao se escolher alterar os centros ou hedges o menu da figura 6.9 é apresentado. Neste menu é escolhido qual das variáveis do processo vai se alterar os centros ou hedges de suas respectivas sub-funções fuzzy.

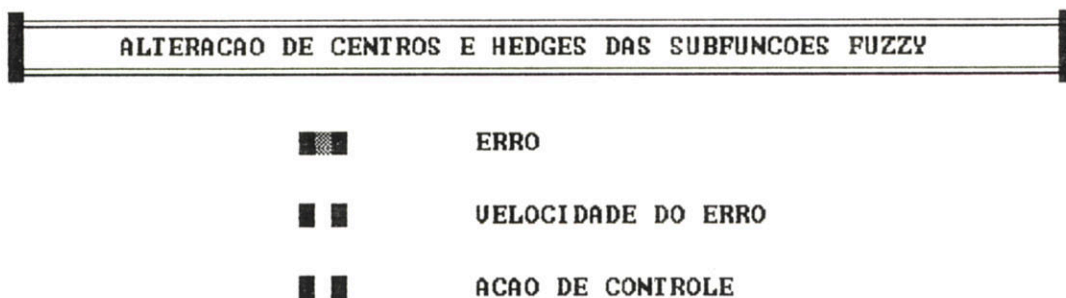


figura 6.9) Tela de alteração dos centros ou hedges

Pode-se sair deste menu teclando ESC e retornando ao menu principal, ou F10 para retornar ao DOS.

Escolhendo-se uma variável para executar a alteração, é apresentada na tela uma sub-função fuzzy, a começar pela primeira função da esquerda, com os limites e centro vigentes, veja figura 6.10. Com as teclas de setas escolhe-se qual das sub-funções se vai executar a alteração. Escolhendo-se, determina-se qual a alteração será executada, se de centro ou hedge. Na alteração de centro é escrito "Entre com o novo valor do centro" e o valor deverá ser escrito no espaço determinado.

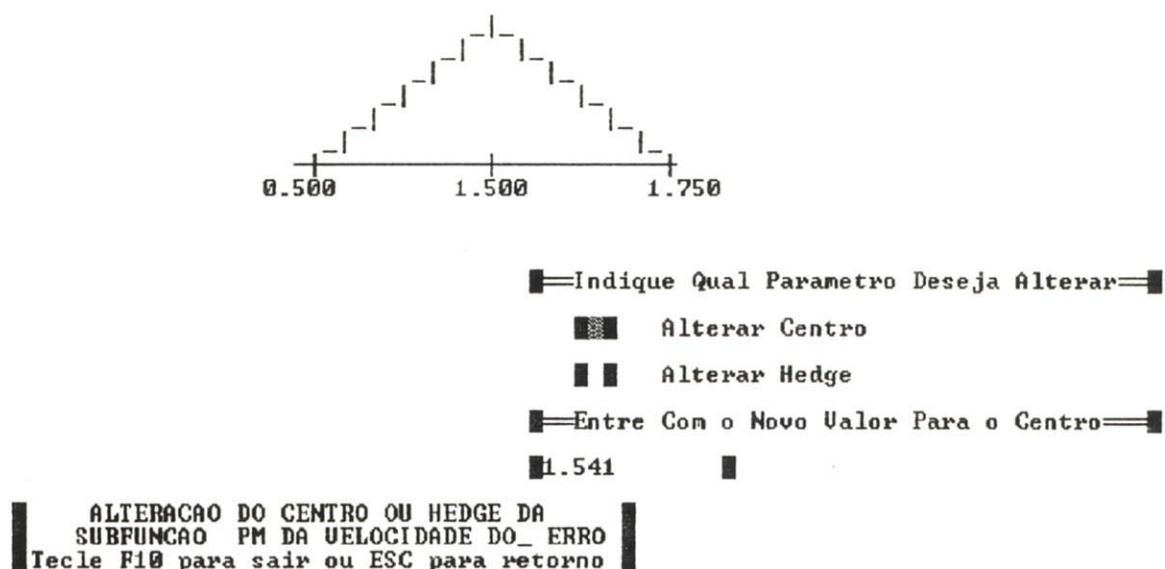


figura 6.10) Tela de alteração de limites ou hedges

No exemplo da figura 6.10 a variável escolhida foi a velocidade do erro e vai se alterar o centro da sub-função PM(positiva média).

Na alteração de Hedge um menu como o da figura 6.11 é apresentado. Neste menu é escolhido o hedge da sub-função de acordo com o que já foi apresentado no capítulo de controladores fuzzy. Para se sair basta teclar <ENTER> para aceitar a alternativa indicada pelo cursor, ESC para retornar e manter a alternativa anterior ou F10 para retornar ao DOS.

ALTERACAO DE TIPO DE HEDGE DA SUBFUNCAO FUZZY

- ■ Extremamente Fechada
- ■ Muito Fechada
- ■ Parcialmente Fechada
- ■ Pouco Fechada
- Sem Alteracao
- ■ Pouco Aberto
- ■ Parcialmente Aberto
- ■ Muito Aberto
- ■ Extremamente Aberto

figura 6.11) Tela de alteração do hedge de uma sub-função

6.3.1.1.3 Alteração da Operação AND

A operação fuzzy AND pode ser alterada por qualquer uma das apresentadas no capítulo de controladores fuzzy anterior. Escolhendo esta opção no menu principal aparece na tela um menu como o da figura 6.12, para a alteração da operação de antecedentes.

Neste menu estão especificadas as operações de antecedentes disponíveis, para selecioná-las basta correr o cursor através das teclas de setas para cima ou para baixo e escolher teclando < ENTER >, o retorno para o menu principal é automático. Se não desejar alterar a operação AND terá que ser teclado < ESC >, o retorno é automático mantendo-se a operação anterior a entrada neste menu. Para sair do programa e retornar ao DOS basta teclar F10.

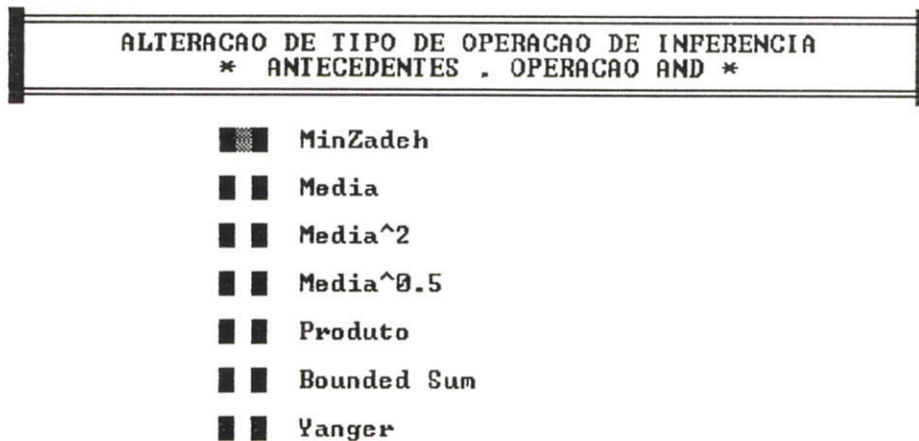


figura 6.12) Menu de alteração da operação AND

6.3.1.1.4 Alteração da Operação OR

Análogo a operação no menu de alteração de operação AND, a operação fuzzy OR é alterada para qualquer uma das apresentadas no capítulo de controladores fuzzy. Escolhendo esta opção "Alteração da Operação OR" no menu principal aparece na tela um menu como na figura 6.13 .

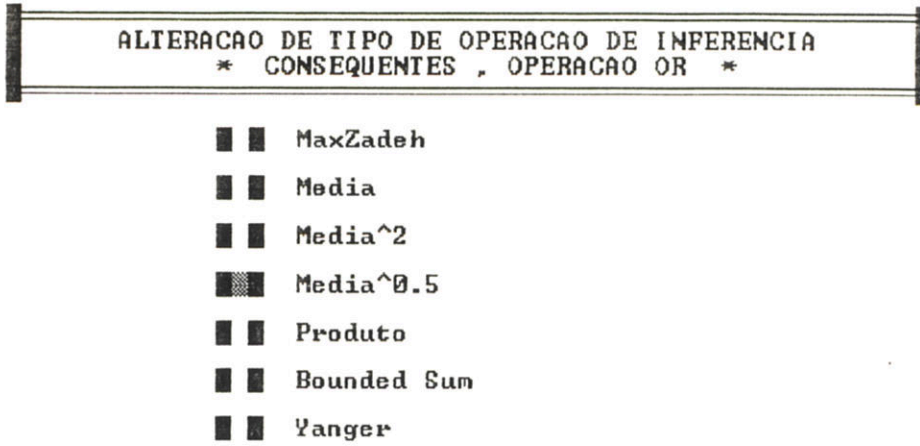


figura 6.13)Menu de alteração da operação OR

Aqui neste menu são apresentadas as alternativas disponíveis para a operação de alteração de consequentes(OR), para seleciona-las basta correr o cursor através das teclas de setas para cima ou para baixo e escolher teclando < ENTER >, o retorno para o menu principal é automático. Se não desejar alterar a operação OR terá que ser teclado < ESC >, o retorno é automático mantendo-se a operação anterior a entrada neste menu. Para sair do programa e retornar ao DOS basta teclar F10.

6.3.1.1.5 Alteração da Base de Regras

Para se alterar a base de regras do controlador fuzzy a opção a selecionada no menu principal é "Alteração de Base de Regras". Selecionada aparece na tela uma matriz que representa a base de regras como apresentado abaixo na figura 6.14

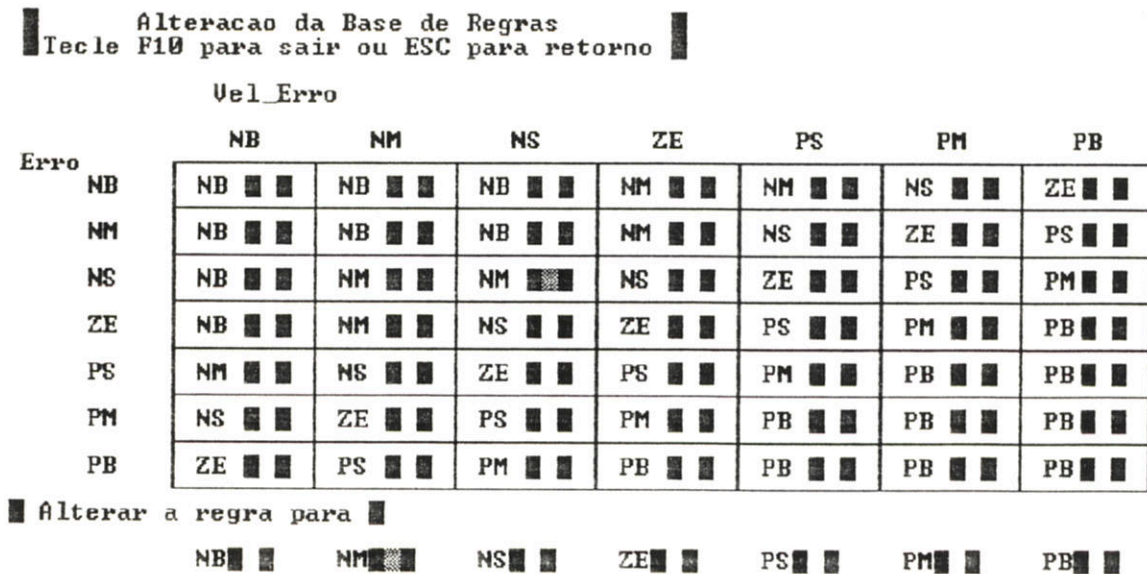


figura 6.14)Base de regras do FuzSsd

Esta base de regras é composta de uma matriz 7X7 definida pelos conceitos apresentados no capítulo de controladores fuzzy, ou seja:

```

IF
    Erro < . >
AND
    Velocidade do Erro < . >
THEN
    Ação de Controle < . >
  
```

Para alterar a base de regras, basta levar o cursor até a posição em que se encontra a sub-função da saída que se deseja alterar, para isto se utiliza as teclas de setas.

No ponto onde será alterada a base de regras, é teclado < ENTER >, e um menu abaixo da base de regras aparece, como descrito na figura 6.14. Este menu tem o nome das funções fuzzy da saída, basta colocar o cursor sobre a desejada para selecioná-la e alterar a base de regras.

Em todas as opções de escolha teclando-se < ESC > retorna-se ao menu principal, entretanto se uma posição da base de regras foi alterada, esta não retornará a configuração anterior. A tecla F10 termina o programa e retorna ao DOS.

6.3.1.1.6 Alteração da operação de Defuzzificação

A alteração da operação de defuzzificação consiste em setar um dos tipos disponíveis num menu como descrito na figura 6.15, de acordo com os tipos apresentados no capítulo de controladores fuzzy.

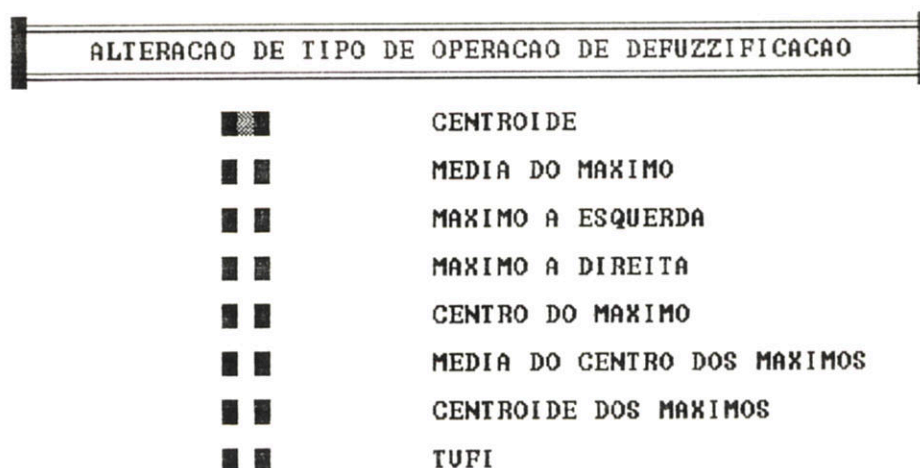


figura 6.15) Menu de alteração de operação de defuzzificação

Da mesma forma dos menus anteriores para se alterar a operação de defuzzificação, basta correr o cursor dentre as alternativas pelas teclas de setas e depois te-

clar < ENTER >. Caso não se deseje alterar basta teclar < ESC > o retorno é automático para o menu principal sem a alteração da operação de defuzzificação.

6.3.1.1.7 Executa

Quando todos os parâmetros do controlador fuzzy estiverem devidamente ajustados a aplicação, juntamente com os cabos de conexão entre as placas ADA 10 e 16 e DAX e acordo com o esquema descrito no capítulo "Placas de Aquisição", parte-se para a execução do programa.

Quando se vai executar, um menu de escolha do tipo de acompanhamento dos dados durante a execução é apresentado, veja figura 6.16 .

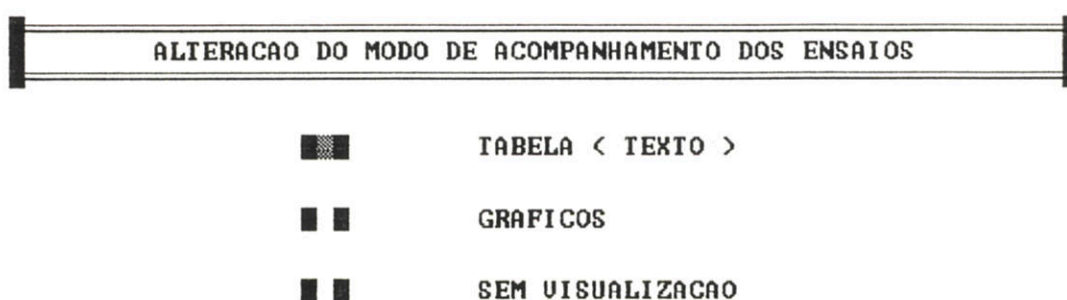


figura 6.16) Menu de escolha do modo de acompanhamento da execução

Pelo menu da figura 6.16 pode-se escolher entre uma visualização em modo:

*TEXTO, onde são apresentados através de uma tabela os dados de leitura do erro e ação de controle calculada, figura 6.17 ;

*GRÁFICOS, onde são apresentados os gráficos de acompanhamento das variáveis envolvidas no processo, no caso erro, velocidade do erro e ação de controle, figura 6.19;

*SEM VISUALIZAÇÃO, sem visualização onde não é apresentado nenhum dado ficando a tela totalmente limpa.

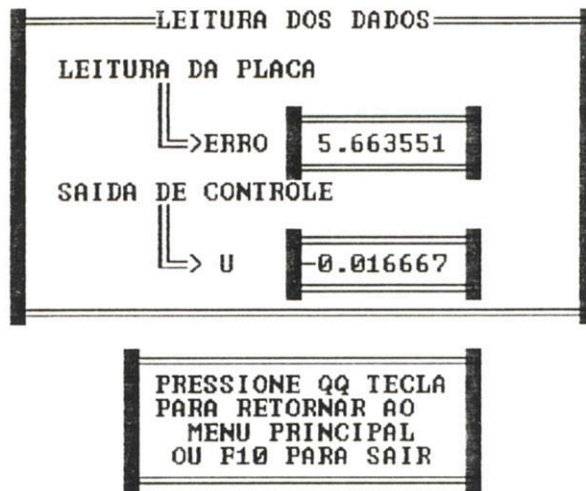


figura 6.17) Tela de acompanhamento da execução no modo texto

Na alternativa de visualização por gráficos, tem-se a alternativa de alteração dos limites dos gráficos, figura 6.18 . Escolhendo-se uma variável para alterar os limites estes últimos deverão ser rescritos. Se não desejar alterar os limites simplesmente tecla-se < ESC >.

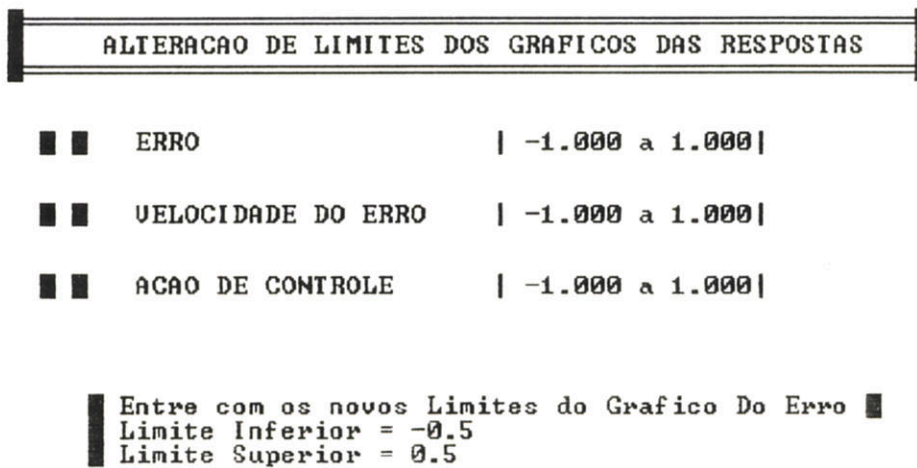


figura 6.18) Tela de alteração dos limites dos gráficos

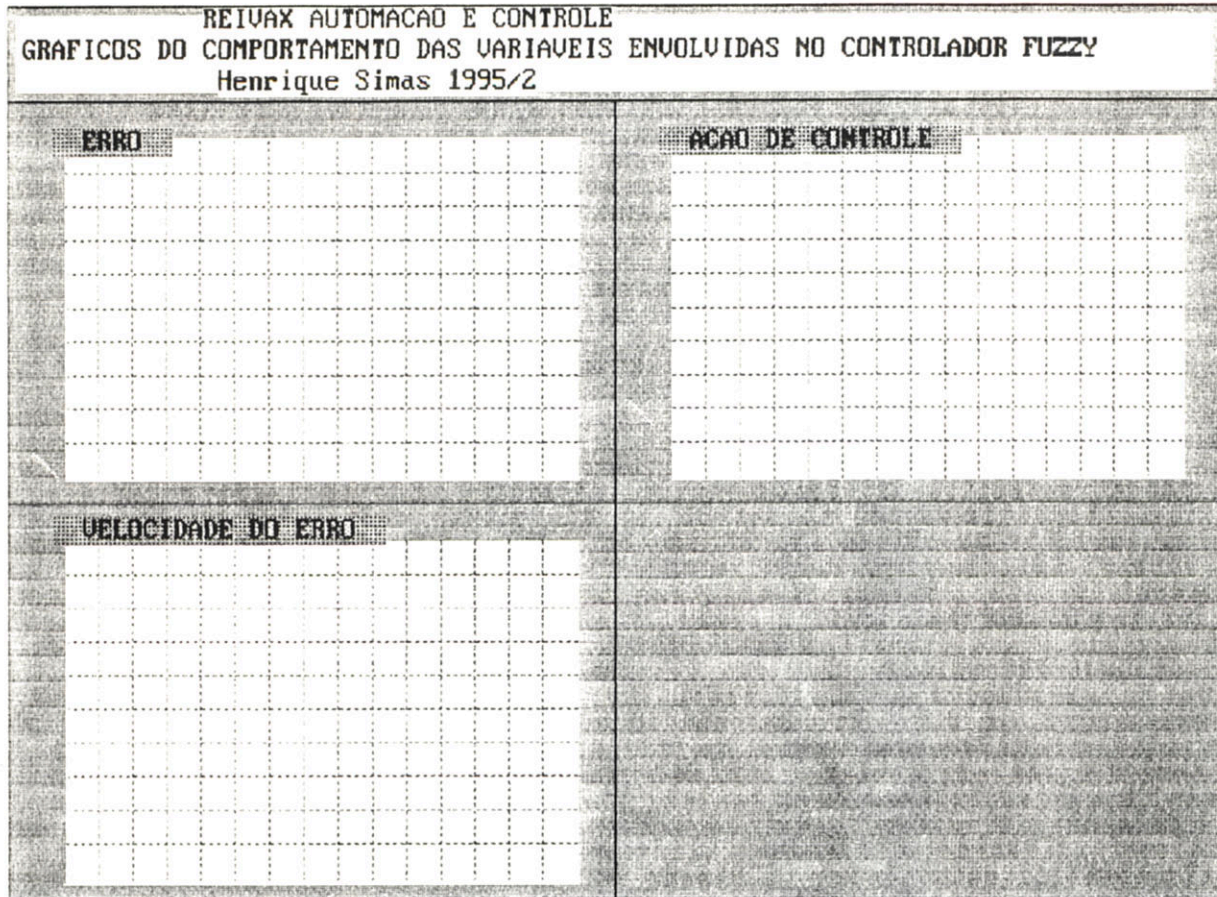


figura 6.19) Tela dos gráficos de acompanhamento

6.3.1.2 Teclas de funções Especiais

Para o auxílio do ajuste do controlador, e proteção dos dados foram programadas teclas de funções especiais F .

As teclas são as seguintes:

6.3.1.2.1 Tecla F1

Responsável por ativação e desativação do controlador fuzzy definido pelo software FIDE, deixando todo o controlador fuzzy do FuzSsd inativo.

6.3.1.2.2 Tecla F2

Quando se deseja salvar um ajuste feito no controlador fuzzy, tecla-se F2, e um quadro como descrito na figura 6.20 aparece esperando pelo nome do arquivo.

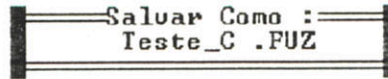


figura 6.20)Quadro de entrada do nome do arquivo de ajustes do controlador

Basta escrever o nome do arquivo sem a extensão. O arquivo salvo tem extensão ".Fuz".

6.3.1.2.3 Tecla F3

Esta tecla é utilizada para se chamar um arquivo de ajustes do controlador salvo. Veja a figura 6.21 .

Aparecerá na tela um menu como o da figura 32. Quando o número de arquivos é maior do que o número de linhas da tela basta teclar <PAGE DOWN> ou <PAGE UP> para se encontrar a página desejada, e através das teclas de setas selecionar o arquivo a ser carregado.



figura 6.21)Menu de chamada de ajustes em arquivos .FUZ

Para sair deste menu sem carregar nenhum arquivo, basta teclar <ESC>, para sair do programa é teclado F10.

6.3.1.2.4 Tecla F4

Esta tecla mostra na tela uma janela com o arquivo atual de ajustes, figura 6.22 .

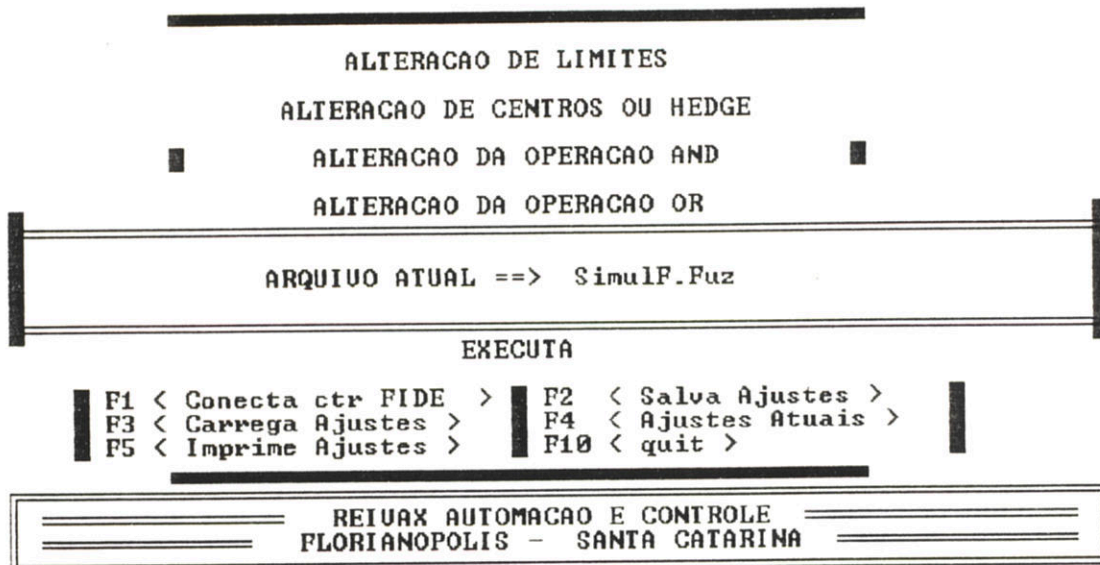


figura 6.22)Quadro que mostra o arquivo de ajustes corrente

Se nenhum arquivo tiver sido carregado então os ajustes atuais serão o default ajustado no programa fonte.

Para sair basta teclar qualquer tecla.

6.3.1.2.5 Tecla F5

Esta tecla imprime um relatório com todos os ajustes atuais do controlador fuzzy, tipos de operações vigentes e base de regras.

Este relatório pode ser visto no anexo 1.

6.3.1.2.6 Tecla F10

Como já visto nos itens anteriores, a tecla F10 tem a função de sair do programa e retornar ao DOS.

6.4 Conclusão

O programa FuzSsd tem vantagens como reusabilidade e fácil interface com o usuário, facilitando assim a aplicação do mesmo em outros processos e uma melhor compreensão dos conceitos da lógica fuzzy.

Este capítulo serviu também para apresentar na forma de um manual todas as potencialidades do FuzSsd, de formas a que se possa fazer uma avaliação sobre este controlador.

As perspectivas de melhoras deste programa são muito atraentes, pois vão de reestruturação do mesmo para um ambiente melhor que não fosse o DOS a aplicação em outros processos. Isto porque mesmo que não se reutilize as funções, os algoritmos estão bem a mão.

Capítulo VII

Simulações

7.1 Introdução

Neste capítulo é apresentada uma avaliação final da performance e características dinâmicas da resposta do controlador fuzzy, conectado como PSS no processo da máquina síncrona.

As simulações são descritas por gráficos coletados no software SSD.

O diagrama de blocos para as simulações foi construído do acordo com a figura 1.4 no capítulo 1.

7.2 Simulações da Variação de Referência

A variação de referência consiste em fazer variar a tensão de referência saída da máquina síncrona de uma percentagem, no caso 5% de uma tensão nominal.

A resposta sem um estabilizador de potência é muito oscilatória e pouco amortecida.

Abaixo, na figura 7.1 temos o resultado da comparação entre o PSS com 2 avanço-atrasos linear, o controlador fuzzy descrito em [Simas 95] e o controlador do fuzzy implementado no FuzSsd.

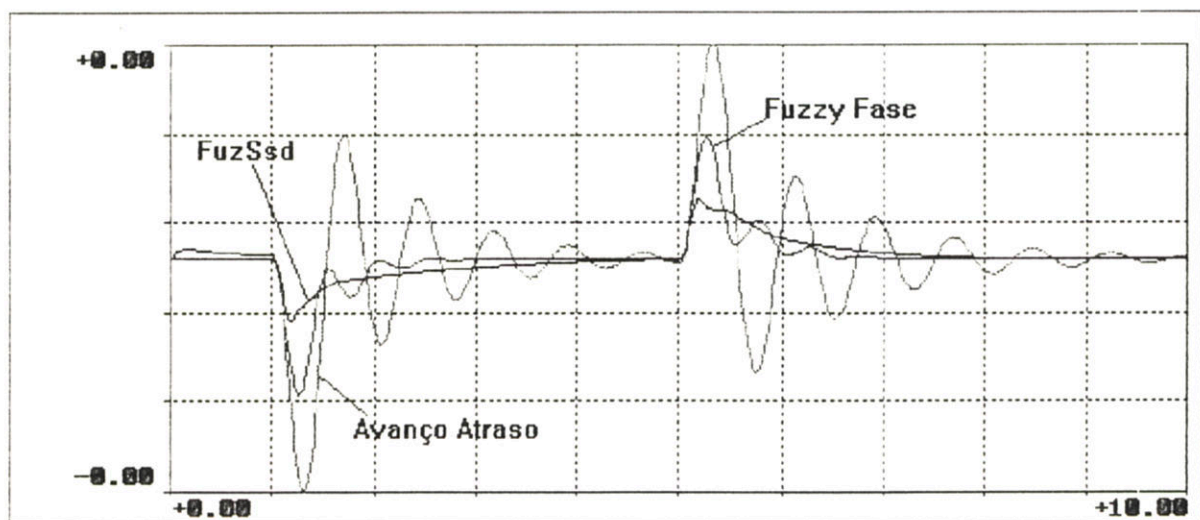


figura 7.1)Simulação da variação de referência

Podemos observar pela figura 7.1 que o controlador fuzzy do FuzSsd teve melhor desempenho dinâmico, comparado com os outros controladores utilizados na simulação. O controlador linear com 2 avanço-atrasos foi o de pior desempenho.

Outra simulação também foi feita para comparar-se a ação de controle destes controladores. A figura 7.2 descreve o gráfico comparativo.

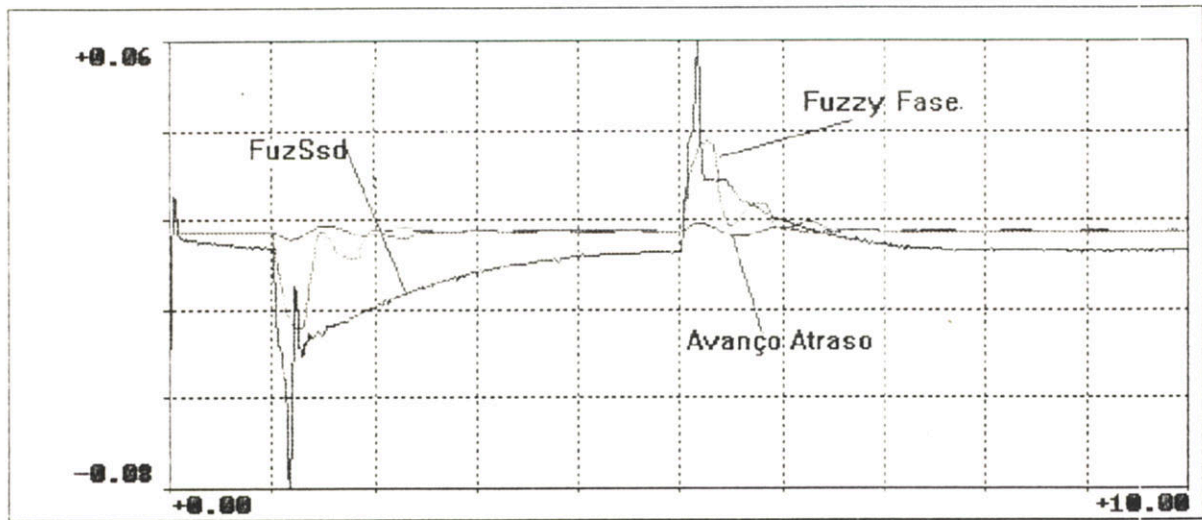


figura 7.2)Comparação das ações de controle de cada controlador para a variação de referência

Apesar do controlador Fuzzy do FuzSsd ter apresentado o melhor desempenho, da sua resposta da ação de controle não se pode falar o mesmo, pois a mesma é limitada na máquina síncrona entre ± 0.1 , e note que na simulação acima esta quase se chegando neste limite, portanto para uma variação mais alta da referência ou dos parâmetros da máquina síncrona, podem ocorrer problemas de saturação dos atuadores, o que não se deseja a princípio.

7.3 Simulações da Variação da Reatância Externa

Para esta simulação, não há a variação de referência, e um bloco auxiliar faz a variação da reatância externa da máquina síncrona simulando uma variação de carga do sistema de distribuição.

Nas simulações esta variação foi de 0.4 .

Na figura 7.3 esta descrita a variação de velocidade comparando-se os mesmos três controladores descritos no item anterior deste capítulo.

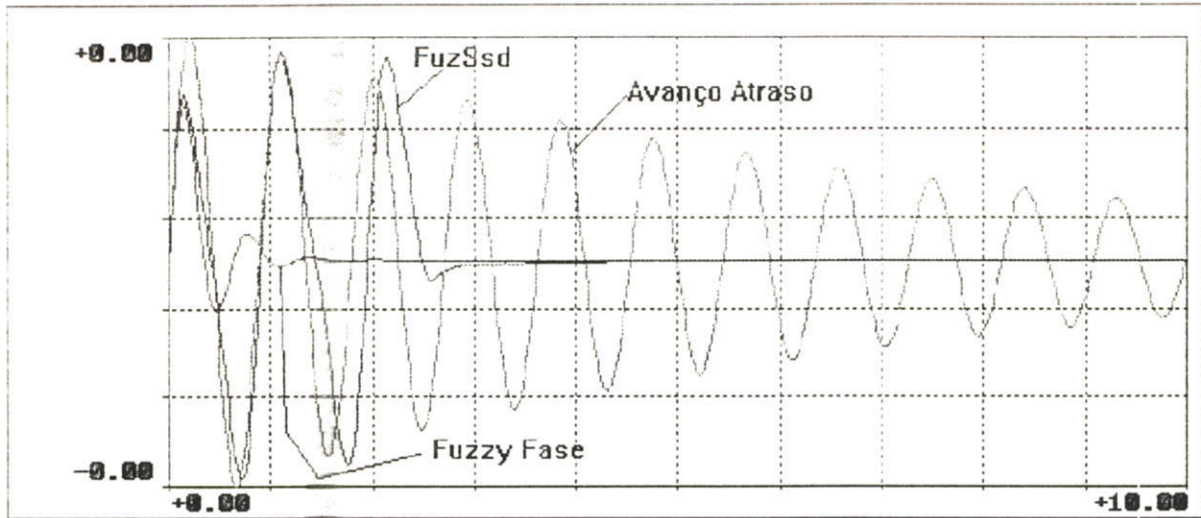


figura 7.3) Simulação da variação de reatância externa

Como observações podemos destacar a dinâmica resultante do controlador fuzzy do FuzSsd, bem diferenciada das demais, caracterizando a forte não-linearidade deste controlador.

Nesta figura 7.3 o controlador Fuzzy baseado no plano de fase [Simas 95] obteve um melhor desempenho, entretanto a resposta proveniente do FuzSsd não deixou a desejar.

Na figura 7.4 esta descrita a ação de controle empregada por cada controlador nesta simulação.

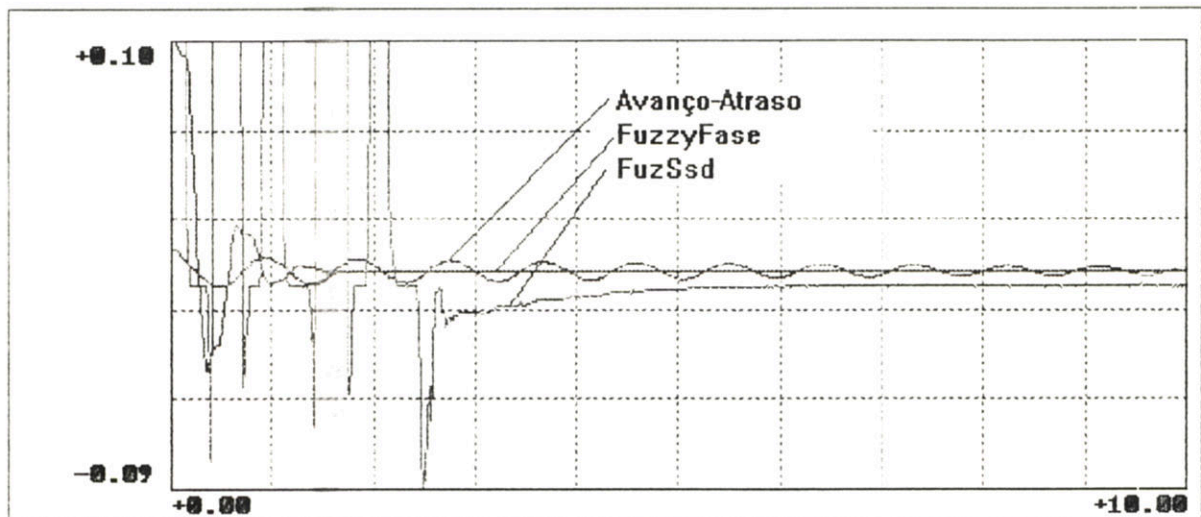


figura 7.4) Comparação de ação de controle para a variação da reatância externa.

Nesta figura 7.4 a ação de controle do FuzSsd não foi muito satisfatória dada que a mesma foi muito oscilatória. Se a base onde se instalar um PSS com este controlador suportar estas oscilações, o mesmo poderá ser aceito com boa performance.

Na figura 7.5 esta descrito o gráfico de comparação entre os escorregamento resultante da máquina síncrona sob a ação de controle de cada um dos controladores descritos acima.

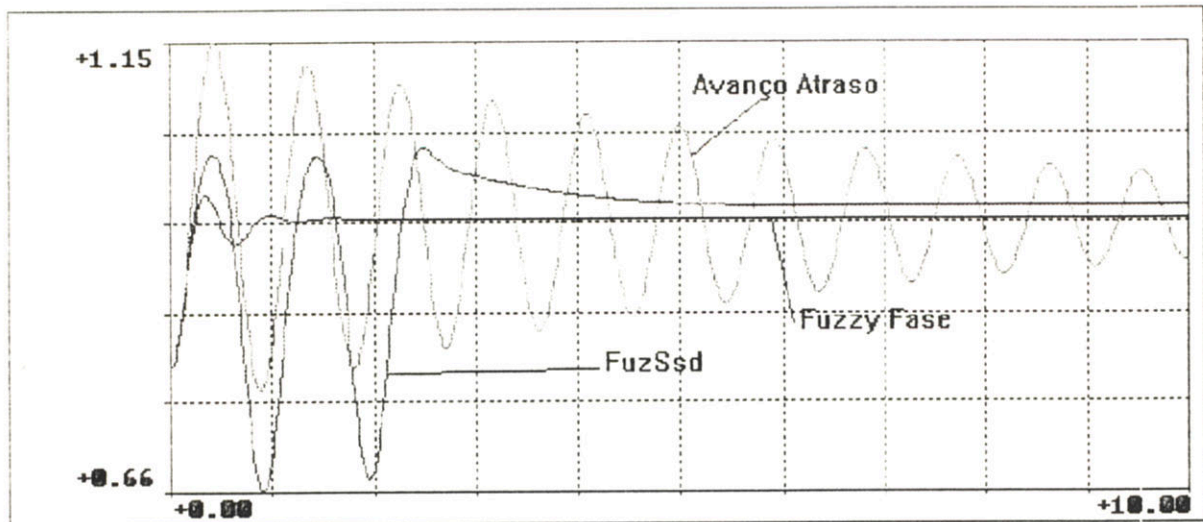


figura 7.5) Comparação do escorregamento da máquina síncrona de acordo com os controladores conectados.

Este escorregamento é resultante da variação da reatância externa e influência na potência fornecida pela máquina síncrona. O objetivo é amortecê-lo o máximo.

Pela figura 7.5 vemos que o controlador FuzSsd tem uma melhor performance e produz um melhor amortecimento em relação ao controlador com 2 avanço-atrasos. Mas o controlador fuzzy do plano de fase apresentou melhores características.

7.4 Comparações entre Ajustes de Defuzzificação

Para finalizar foi feito um estudo de comparação para se demonstrar como pode variar a performance do controlador fuzzy em função dos ajustes e escolha das operações fuzzy.

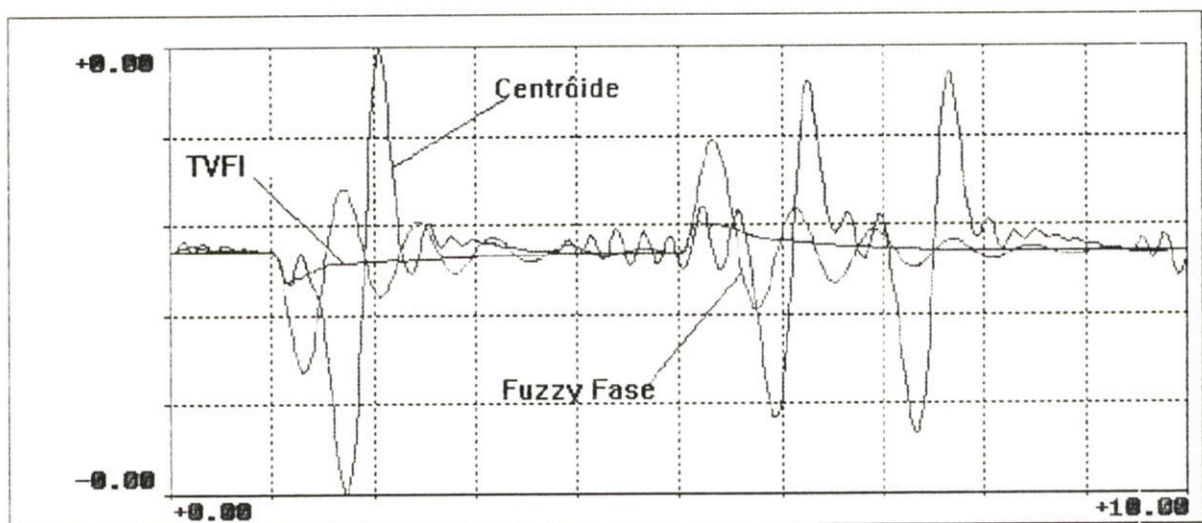


figura 7.6) Comparação entre técnicas de defuzzificação

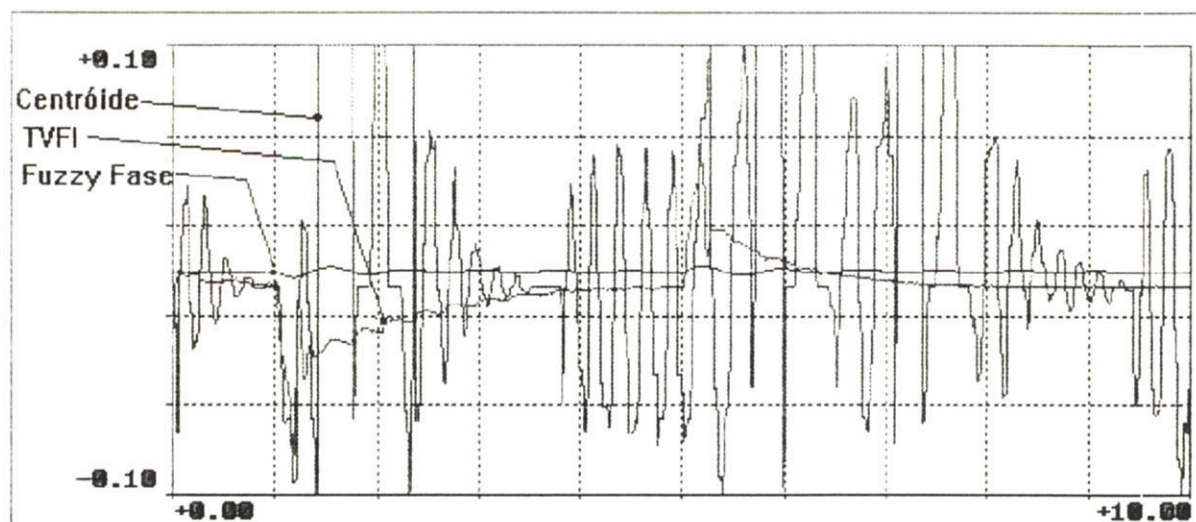


figura 7.7) Comparação de ações de controle de acordo com ajustes no controlador.

Na figura 7.6 é aplicada uma referência, como item 2 deste capítulo, e uma comparação entre as respostas de variação de velocidade pode ser mostrada, neste caso foi ajustado apenas a operação de defuzzificação (Centróide e TVFI), e comparado ainda com a dinâmica da variação de velocidade do controlador fuzzy baseado no plano de fase.

Na figura 7.7 é apresentado para o mesmo exemplo ação de controle resultante da comparação entre os ajustes do controlador fuzzy do FuzSsd e do controlador fuzzy baseado no plano de fase.

Note como pode-se encontrar diferentes dinâmicas para um mesmo ajuste uma mesma base de regras e apenas alterando uma única operação. Convém a quem projetar um controlador fuzzy, avaliar através de simulações qual melhor ajuste das operações limites e base de regras, e então poderá alcançar a melhor performance do controlador fuzzy.

7.5 Conclusão

Neste capítulo foram apresentadas as simulações feitas para se comparar o desempenho do controlador fuzzy implementado no FuzSsd.

Como conclusão final pode-se levantar que o controlador fuzzy implementado possui características que viabilizam seu estudo e implementação num futuro em um processo real. Apesar das respostas serem saturadas na maioria das situações, isto não seria problema pois, como temos um número bastante expressivo de combinações entre ajustes que se pode fazer no controlador fuzzy, pode-se encontrar um ajuste que venha a amenizar tal problema e ficar dentro de nossas especificações.

Conclusão Geral

O desafio da representação do pensamento humano está apenas começando, os sistemas especialistas e os algoritmos conhecidos atualmente representam apenas a ponta do iceberg. Isto porque sabe-se que a mente humana é tão complexa que ainda não foram apresentados dados conclusivos quanto ao seu funcionamento completo.

A lógica fuzzy, redes neurais, algoritmos genéticos, entre outros servem como base de pesquisa para o avanço da engenharia de controle e automação no sentido de se pesquisar novas ferramentas, as quais possam servir de base para um sistema que imite uma ação humana perante uma situação.

Neste trabalho foi apresentado num contexto didático, a teoria, prática e implementação de um controlador fuzzy, dando maior atenção ao fato da lógica fuzzy ser de difícil entendimento num primeiro contato e portanto dado ênfase especial, em todos os capítulos sobre a mesma, chegando a detalhes mínimos, para que não se tenham dúvidas no momento de se implementar este tipo de controlador. A experiência no estudo e implementação de um controlador fuzzy, tópicos de redes neurais e algoritmos genéticos, traz consigo a visão paradoxal, ou seja, apesar de um controle fuzzy não necessitar, em teoria, do conhecimento do processo onde será implantado, há a necessidade de um embasamento matemático na teoria de controle, pois a esta última serve de base para que o projetista ou o engenheiro de controle tenha meios de avaliar a performance do sistema através de conceitos aprendidos durante os cursos de graduação e pós-graduação a que tem que se submeter.

As respostas computadas nas simulações mostraram a viabilidade de implementação e pesquisa deste tipo de controlador. Tomando-se por base este relatório, que tem um caráter didático científico, pode-se utiliza-lo como referencia no desenvolvimento de aplicações em lógica fuzzy.

O controlador implementado no FuzSsd.C possui algumas vantagens para trabalhos futuros na área de controle:

- *Adaptação e implementação em ambiente for Windows;
- *Adaptação em Classes e objetos - funções implementadas neste programa são modulares;
- *Utilização em trabalhos em outras processo;
- *Criação de debuggers, analisadores e simulações passo-a-passo;
- *Possibilidade de se implementar programas de treinamento de redes neurais.

O FuzSsd.C também pode ser utilizado no aprendizado em laboratório de controladores fuzzy, por ter uma interface amigável e de fácil compreensão, destacando aqui o objetivo principal deste projeto que era o de implementar uma ferramenta de simples para estudos de controladores fuzzy.

A pesquisa da estabilidade dos controladores fuzzy é o grande desafio para os próximos anos, inevitável e imprescindível, é o único caminho para uma maior difusão e aceitação do controle fuzzy na comunidade científica, como uma alternativa sustentável na teoria de controle.

Bibliografia

- [**Anderson 77**] P. M. Anderson e A. A. Fouad, "Power System Control and Stability", Iowa State University Press, 1977.
- [**Apronix 94**] Apronix, "Fide User's Manual", Apronix Incorporated, 1994.
- [**Bimal 94**] K. B. Bimal, "Expert System, Fuzzy Logic and Neural Network, Application in Power Electronics and Motion Control", IEEE, Vol 82, No 8, pp.1303-1323, 1994.
- [**Chiaberge 95**] M. Chiaberge e L. M. Reynery, "Cintia : A Neuro-Fuzzy Real-Time Controller for Low Power Embedded System", IEEE , Vol 15, No 3, pp. 40-47, 1995.
- [**Cox 94**] E. Cox, "The Fuzzy System Handbook : a Practitioner's Guide to Building, Using, and Maintaining Fuzzy System", Academic Press, 1994.
- [**Dubois 80**] D. Dubois e H. Prade, "Fuzzy Sets and Systems: Theory and Practice", Academic Press, 1980.
- [**Galichet 95**] S. Galichet e L. Foulloy, "Fuzzy Controllers : Synthesis and Equivalences", IEEE , Vol. 3, No 2 , pp. 140-148 , 1995.
- [**Hasan 94**] A. R. Hasan, T.S.Martis e A. H. M. S. Ula, "Design and Implementation of a Fuzzy Controller Based Automatic Voltage Regulator for a Synchronous Generator", IEEE, Vol. 9, No 3, pp. 550-557, 1994.
- [**Homaifar 95**] A.Homaifar e E. McCormick, "Simultaneous Design of Membership Function and Rule Sets for Fuzzy Controllers Using Genetic Algorithms", IEEE, Vol. 3, No 2, pp. 129-139, 1995.
- [**Ishibuchi 95**] H. Ishibuchi, K. Nozaki, N. Yamamoto, e H. Tanaka, "Selecting Fuzzy if-Then Rules for Classification Problems Using Genetic Algorithms", IEEE, Vol 3, No 3, pp. 260-270, 1995.
- [**Kaufmann 75**] A. Kaufmann, "Introduction to the Theory of Fuzzy Subsets", Academic Press, 1975.
- [**Klir 81**] G. A. Klir e T. A. Folger, "Fuzzy Sets, Uncertainty and Information", Prattice Hall, 1981.
- [**Pernel 95**] C. Pernel, J.Themlin, e M. Acheroy, "Optimization of Fuzzy Expert system Using Genetic Algorithms and Neural Networks" ,IEEE, Vol 3, No 3, pp. 300-312, 1995.
- [**Sharaf 94**] A. M. Sharaf e T. T. Lie , "ANN Based Pattern Classification of Synchronous Generator Stability and Loss of Excitation", IEEE , Vol 9, No 4, pp. 753-759, 1994.
- [**Simas 95**] H. Simas, "Estudo de Avaliação do Desempenho de Controladores Fuzzy em Máquina Síncrona ", Monografia de Estágio, ECAI, UFSC, Julho de 1995.

[**Tang 94**] Y. Tang e L. Xu, "Fuzzy Logic Application for Intelligent Control of a Variable Speed Drive", IEEE , Vol 9, No 4 , pp. 679-685, 1994.

[**Zhang 95**] Y. Zhang , O. P. Malik e G. P. Chen , "Artificial Neural Network Power System Stabilizers in Multi-Machine Power System Enviroment", IEEE , Vol 10, No 1, pp. 147-155, 1995.

Descrição das funções implementadas no programa FuzSsd

void Tela_Interface(void)

Função responsável pela apresentação dos menus de alteração e ajustes do controlador fuzzy.

void Salva_Arquivos(void)

Função que escreve em arquivo os ajustes vigentes no controlador fuzzy. Salvando o mesmo com a extensão ".Fuz" .

int Mostra_Diretório(void)

Mostra o diretório com os arquivos de ajustes .fuz , retornando um inteiro que indica se um arquivo foi chamado(maior que zero) ou não (zero).

void Mostra_Arquivo_Atual(void)

Mostra num quadro o arquivo atual de ajustes do controlador, escrevendo "default" se nenhum arquivo foi chamado ou salvo.

void Carrega_Arquivo(int)

Carrega ajustes do controlador de um arquivo .fuz no diretório corrente. O inteiro que entra como parâmetro serve para verificar se a função foi chamada na entrada do programa FuzSsd pelo DOS, ou se é proveniente de uma chamada dentro da execução do programa.

int Imprime_Caracter(char)

Imprime um caracter na impressora e retorna um valor que indica se a operação foi feita com sucesso ou não.

int Inicia_Impressora(void)

Inicializa a impressora retornando um valor inteiro que indica se a impressora esta conectada ou não na porta paralela.

int Estado da Impressora(void)

Durante a impressão de uma página esta rotina verifica se há algum problema com a comunicação com a impressora, retornando um inteiro que indica se esta tudo OK ou não.

void Imprime_Linha(char *)

Imprime na impressora uma string indicada na chamada da função.

void Imprime_Float(float)

Nesta função um valor float é convertido em string e impresso dentro de um tamanho especificado.

void Imprime_Ajustes(void)

Usando as rotinas de impressão esta função é responsável pela impressão de um relatório onde são especificados todos os ajustes do controlador fuzzy atuais.

SFF Inicia(float LEsqu, float LDir, float Cen, float hed, int lden)

Nesta é inicializada uma sub-função fuzzy (estrutura SFF). Para isto entra-se com os parâmetros

LEsqu	==>	ponto do limite esquerdo
LDir	==>	ponto do limite direito
Cen	==>	ponto do centro
hed	==>	potência que indica a abertura da sub-função fuzzy(
hedge)		
lden	==>	identificador da sub-função

void Inicia_Placa_ADA16(void)

Antes de fazer o fechamento da malha e iniciar o controlador fuzzy, esta função é chamada automaticamente de modo a fazer a programação da placa ADA 16, para leitura do erro e velocidade do erro e colocar na sua saída a ação de controle calculada.

float Defuzzificação(float *MiE, int *IdentfU, SFF *FunAcao, int l)

Função responsável pelo cálculo da saída do controlador fuzzy através da defuzzificação.

Tem como parâmetros:

MiE => vetor de graus de pertinência das funções encontradas na base de regras.

IdentfU => Vetor identificando quais as sub-funções da saída de controle foram determinadas pela base de regras, respectivamente tal qual o vetor de graus de pertinência associado.

FunAcao = > Vetor com todas as 7 sub-funções fuzzy da saída de controle .

I => número de funções fuzzy da saída encontradas para o cálculo da saída, sempre igual a 4, devido a aspectos construtivos do programa;

float Calcula_Mi(SFF FunFuz,float Valor)

Determinada a sub-função fuzzy(FunFuz) da entrada a que pertence uma determinada medição(Valor), esta função retorna o grau de pertinência da mesma a aquela sub-função fuzzy.

float Intersecção(float a1, float a2)

De acordo com o tipo de operação de intersecção (antecedentes, AND) setada no programa, esta função executa esta operação entre duas entradas, a1 e a2, retornando o resultado da operação.

float Uniao(float a1, float a2)

Executa a operação de união(conseqüentes , OR) setada no programa, entre duas entradas, retornando o resultado da operação.

float AndYanger(float a1, float a2)

Esta função é responsável pelo cálculo da operação intersecção definida por AND Yanger entre duas entradas, retornando o resultado da operação.

float OrYanger(float a1, float a2)

Esta função é responsável pelo cálculo da operação união definida por OR Yanger entre duas entradas, retornando o resultado da operação.

void DesLimites(float vet[])

Função responsável por escrever na tela os limites vigentes de uma variável fuzzy. O vetor vet traz os pontos dos limites da variável fuzzy.

void Altera_Centro(float vetorC[], float vetorL[], float vetorH[], int indica)

Esta função responsável pela alteração dos hedges e centros de uma variável do controlador fuzzy. Os parâmetros de entrada são:

vetorC ==> vetor de centros das sub-funções fuzzy da variável do controlador.

vetorL ==> vetor dos limites das sub-funções fuzzy, utilizado no desenho das mesmas.

vetorH==> vetor com os expoentes que definem o hedge de cada sub-função fuzzy da variável do controlador.

indica => Indica qual variável do controlador está sendo alterado seus centros de suas sub-funções fuzzy.

void DeFuEsq(float Cent, float limDir)

Função responsável pelo desenho da sub-função fuzzy mais a esquerda(NB) de uma variável do controlador na execução da alteração de centros e hedges. Os parâmetros Cent e limDir são o ponto central da função e o limite direito suficientes para a execução da função.

void DeFuDir(float Cent, float limEsq)

Função responsável pelo desenho da sub-função fuzzy mais a direita(PB) de um variável do controlador na execução da alteração de centros e hedges. Os parâmetros Cent e limEsq são o ponto central da função e o limite esquerdo suficientes para a execução da função.

void DeFuTri(float limEsq, float Cent, float limDir)

Função responsável pelo desenho das sub-funções fuzzy que estão entre as mais a esquerda e direita(NB e PB respectivamente).

Esta função é utilizada pela função altera centros e hedges.

Os parâmetros limEsq, Cent e limDir são respectivamente o limite esquerdo, ponto central e limite direito da sub-função fuzzy.

void Altera_Hedge(float vetorH[], int indica)

Esta função tem o objetivo de se alterar o hedge de uma sub-função fuzzy. É utilizada na função de alteração de hedge e centros.

Os parâmetros de entrada são o vetor de hedge da variável(vetorH[]) e qual a sub-função será alterada o seu hedge(indica).

void AlteraBaseDeRegras(void)

Função responsável pela altera a base de regras vigente no controlador fuzzy.

void EscreveRegras(void)

Função responsável pela atualização do quadro da base de regras na tela do computador a cada vez que uma alteração é executada.

void Altera_Defuzzificação(void)

Nesta função é alterada a função fuzzy de defuzzificação por qualquer uma das estudadas no capítulo referente a controladores fuzzy.

void Faz_Quadro_Grafico(void)

Esta função desenha o leiaute da tela de apresentação gráfica juntamente com os gráficos onde serão apresentados a evolução das variáveis do controlador fuzzy.

void Retângulo(int x1, int x2, int x3, int x4, short cor)

Esta função funciona como auxiliar da função Faz_Quadro_Grafico no desenho de retângulos na tela nas coordenadas dadas como parâmetros.

void Atualiza_Quadro_Grafico(void)

No acompanhamento por gráficos, esta função atualiza todos os gráficos com os valores correspondentes a cada período de amostragem.

void Fim_Programa(void)

Função responsável por desativar o programa FuzSsd e retornar ao DOS.

extern adasm(int, int*, int, unsigned int, int*, int*)

Esta função foi projetada e implementada pelos engenheiros da REIVAX e tem o objetivo de fazer a leitura da ADA 16 dos valores de erro e velocidade do erro

extern exec_mqsn(float ent1, float ent2, float *U)

Função gerada pelo software FIDE, a qual engloba todo um controlador fuzzy. Os parâmetros são ent1 e ent2 como entradas do controlador e é setado uma saída U como ação de controle calculada.

Anexo II

Relatório de ajustes do Controlador Fuzz

DADOS DOS AJUSTES DO CONTROLADOR FUZZY

Ajustes salvos no Arquivo => TESTE_B.FUZ

1) LIMITES DAS SUBFUNCOES

ERRO		Lim Esquerdo	Centro	Lim Direito	Hedge
SubFuncao	NB	--	-0.500	-0.300	Extremamente Fechada
SubFuncao	NM	-0.500	-0.300	-0.200	Muito Fechada
SubFuncao	NS	-0.300	-0.200	0.0000	Parcialmente Fechada
SubFuncao	ZE	-0.200	0.0000	0.2000	Pouco Fechada
SubFuncao	PS	0.0000	0.2000	0.3000	Pouco Aberta
SubFuncao	PM	0.2000	0.3000	0.5000	Parcialmente Aberta
SubFuncao	PB	0.3000	0.5000	--	Extremamente Aberta

VEL ERRO		Lim Esquerdo	Centro	Lim Direito	Hedge
SubFuncao	NB	--	-1.750	-1.500	Muito Aberta
SubFuncao	NM	-1.750	-1.500	-0.500	Parcialmente Aberta
SubFuncao	NS	-1.500	-0.500	0.0000	Pouco Aberta
SubFuncao	ZE	-0.500	0.0000	0.5000	Pouco Fechada
SubFuncao	PS	0.0000	0.5000	1.5000	Parcialmente Fechada
SubFuncao	PM	0.5000	1.5000	1.7500	Muito Fechada
SubFuncao	PB	1.5000	1.7500	--	Extremamente Fechada

ACAO CONTROLE		Lim Esquerdo	Centro	Lim Direito	Hedge
SubFuncao	NB	--	-1.400	-0.500	Sem Alteracao
SubFuncao	NM	-1.400	-0.500	-0.100	Sem Alteracao
SubFuncao	NS	-0.500	-0.100	0.0000	Sem Alteracao
SubFuncao	ZE	-0.100	0.0000	0.1000	Sem Alteracao
SubFuncao	PS	0.0000	0.1000	0.5000	Sem Alteracao
SubFuncao	PM	0.1000	0.5000	1.4000	Sem Alteracao
SubFuncao	PB	0.5000	1.4000	--	Sem Alteracao

2) OPERACOES FUZZY

Operacao AND = >Produto
 Operacao OR = >Bounded sum

3) BASE DE REGRAS

Erro	Velocidade do Erro						
	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NM	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NM	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

4) OPERACAO DE DEFUZZIFICACAO

Operacao DEFUZZIFICACAO = >Centroide dos Maximos

Anexo III

Diagrama de Blocos da Máquina Síncrona

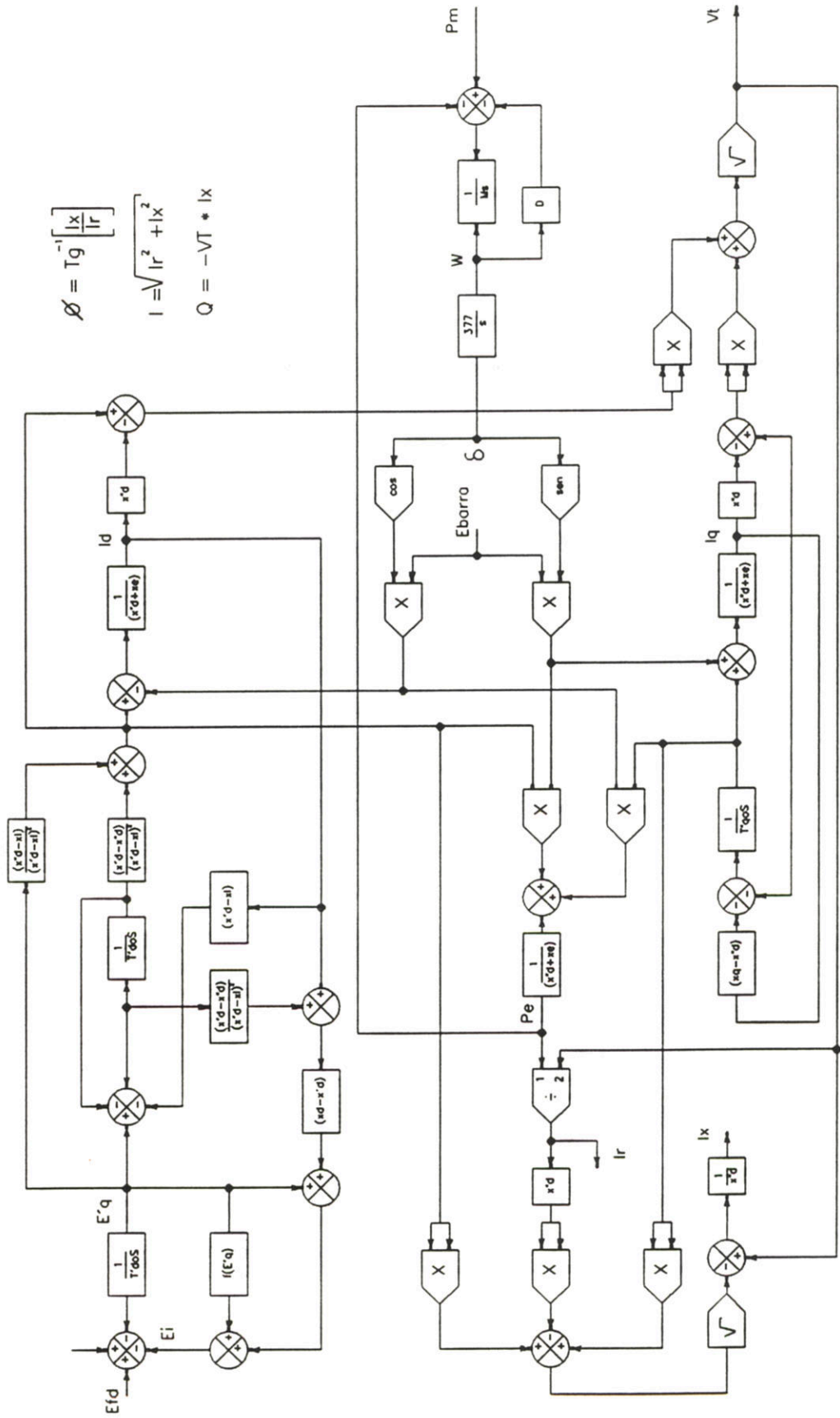


DIAGRAMA DE BLOCOS DA MÁQUINA SÍNCRONA DE PÓLOS SALIENTES LIGADA A BARRA INFINITA