

SISBI/UFU



1000212896

MON
681.51
C266u
TES/MEM

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

**O USO DE CONTROLADORES FUZZY ADAPTATIVOS,
IMPLEMENTADOS EM MICROCONTROLADORES, NO CONTROLE DE
VIBRAÇÕES DE SISTEMAS MECÂNICOS**

Dissertação apresentada
à Universidade Federal de Uberlândia por:

ISRAEL JORGE CÁRDENAS NUÑEZ

Como parte dos requisitos para obtenção do título de Mestre em
Engenharia Mecânica

Banca Examinadora:

Prof. Dr. José Francisco Ribeiro	(UFU) – Orientador
Prof. Dr. Valder Steffen Júnior	(UFU)
Prof. Dr. Celso K. Morooka	(UNICAMP)

Uberlândia, 13 de Dezembro de 2001



ALUNO: ISRAEL JORGE CÁRDENAS NUÑEZ

ÚMERO DE MATRÍCULA: 5992607-1

ÁREA DE CONCENTRAÇÃO: Mecânica dos Sólidos e Vibrações

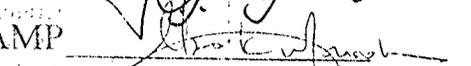
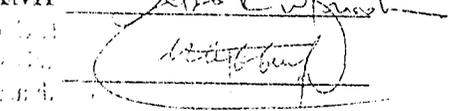
PÓS GRADUAÇÃO EM ENGENHARIA MECÂNICA: NÍVEL MESTRADO

TÍTULO DA DISSERTAÇÃO:

“O Uso de Controladores Fuzzy Adaptativos, Implementados em Microcontroladores, no Controle de Vibrações de Sistemas Mecânicos”

ORIENTADOR: Prof. Dr. José Francisco Ribeiro

A Dissertação foi APROVADA em reunião pública, realizada no Anfiteatro do Bloco E do Campus Santa Mônica, em 13 de dezembro de 2001, às 14:00 horas, com a seguinte Banca Examinadora:

NOME	ASSINATURA
José Francisco Ribeiro, Prof. Dr.	UFU 
Celso Kazuyuki Morooka, Prof. Dr.	UNCAMP 
Valder Steffen Jr, Prof. Dr.	UFU 

Uberlândia, 13 de dezembro de 2001.

DEDICATÓRIA

*A Jorge y Angela Nuñez
A Nicolás Javier Santos Cárdenas*

AGRADECIMENTOS

A Deus.

À família Nuñez: pais, tios, irmãs, primos, primas e sobrinhos pelo apoio.

Ao Professor José Francisco Ribeiro pelo exemplo de trabalho e pela orientação.

A todos meus professores (as) da graduação e pós graduação da Universidade Federal de Uberlândia pelo ensino.

Aos amigos Tito, Gustavo, Rafael e Patrick pelo trabalho em equipe.

A CAPES – Coordenação de Aperfeiçoamento de pessoas de Nível Superior, pelo apoio financeiro oferecido.

O USO DE CONTROLADORES *FUZZY* ADAPTATIVOS, IMPLEMENTADOS EM MICROCONTROLADORES, NO CONTROLE DE VIBRAÇÕES DE SISTEMAS MECÂNICOS

SUMÁRIO

Capítulo 1	INTRODUÇÃO	1
Capítulo 2	LÓGICA <i>FUZZY</i>	4
2.1	Introdução	4
2.2	Conjuntos <i>Fuzzy</i>	4
2.3	Operações sobre Conjuntos <i>Fuzzy</i>	6
2.4	Proposições e Operadores Lógicos <i>Fuzzy</i>	8
2.5	Métodos de Inferência da Base de Regras <i>Fuzzy</i>	9
2.6	Controle a Partir da Lógica <i>Fuzzy</i>	11
2.6.1	Fuzzificação das Entradas	12
2.6.2	Inferência da Base de Regras	13
2.6.3	Defuzzificação das saídas	17
Capítulo 3	CONTROLE ADAPTATIVO	20
3.1	Introdução	20
3.2	Controle <i>Fuzzy</i> Adaptativo	20
3.3	Elementos dos Controladores Adaptativos	22
3.3.1	Monitor de Processo	22
3.3.1.1	Monitoramento das Variáveis Paramétricas	22
3.3.1.2	Monitoramento da Medida de Desempenho	22
3.3.2	Mecanismo de Adaptação	23
3.3.2.1	Mecanismos de Adaptação Auto Sintonizados	23

3.3.2.2	Mecanismos de Adaptação Auto Organizados	25
3.4	Proposta de Adaptação do Universo do Discurso a partir do Fator de Escala	26
3.4.1	Inicialização e Memória do Fator de Escala	29
Capítulo 4	MICROCONTROLADORES	30
4.1	Introdução	30
4.2	Características Gerais	30
4.3	Generalidades da Arquitetura dos Microcontroladores PIC	31
4.4	O Microcontrolador PIC 16F877	32
4.4.1	Arquitetura Interna	34
4.4.2	Organização da Memória do Programa	36
4.4.3	Memória de Dados	37
4.4.4	Pinos de Entradas e Saídas E/S	39
4.4.5	Configuração do Oscilador	39
4.4.6	Os Ciclos do Temporizador “Clock”	40
4.4.7	Características Especiais	41
4.5	Programação dos Microcontroladores	43
Capítulo 5	RESULTADOS EXPERIMENTAIS	44
5.1	Introdução	44
5.2	Controle <i>Fuzzy</i> Adaptativo de uma Mesa Vibratória por Meio de um PC	44
5.2.1	Projeto do Controlador <i>Fuzzy</i> Principal	45
5.2.2	Projeto do Controlador <i>Fuzzy</i> Adaptativo	47
5.2.3	Resultados Experimentais e Discussões	49
5.2.3.1	Sensibilidade do Controlador <i>Fuzzy</i> Principal ao Universo do Discurso	50
5.2.3.2	Sensibilidade do Mecanismo de Adaptação à Variações do Universo do Discurso do Mecanismo <i>Fuzzy</i> de Adaptação	52
5.2.3.3	Comparação entre o Controlador <i>Fuzzy</i> com e sem Mecanismo <i>Fuzzy</i> de Adaptação	54

5.2.3.4	Influência do Valor Inicial do Fator de Escala Kv	54
5.3	Controle <i>Fuzzy</i> Adaptativo de uma Viga Flexível por meio de um Microcontrolador	57
5.3.1	Descrição da eletrônica de Controle do Microcontrolador	58
5.3.2	Análise em Freqüência da Resposta da Viga	60
5.3.3	Projeto do Controlador <i>Fuzzy</i> principal	61
5.3.4	Projeto do Controlador <i>Fuzzy</i> Adaptativo	62
5.3.5	Implementação do Controlador <i>Fuzzy</i> no Microcontrolador	62
5.3.6	Resultados Experimentais e Discussões	63
5.3.6.1	Desempenho do controlador <i>Fuzzy</i> com e sem mecanismo <i>Fuzzy</i> de adaptação	64
5.3.6.2	Sensibilidade do Mecanismo de Adaptação à Variações do Universo do Discurso do Mecanismo <i>Fuzzy</i> de Adaptação	66
5.3.6.3	Sensibilidade do controlador <i>Fuzzy</i> Adaptativo à variações do Kv inicial	67
Capítulo 6	CONCLUSÕES E FUTUROS DESDOBRAMENTOS	71
Referências Bibliográficas		73
APÊNDICES		
Apêndice I	– Programa em linguagem “C” para o controle da Mesa Vibratória	77
Apêndice II	– Programa em linguagem “C” para o controle da Viga Engastada	82
Apêndice III	– Componentes do Sistema Mecânico da Mesa Vibratória	89
Apêndice IV	– Componentes do Sistema Mecânico da Viga Engastada	90

SIMBOLOGIA

A, B	- Conjuntos <i>fuzzy</i>
C	- Complemento de um conjunto <i>fuzzy</i>
N	- Variável lingüística negativo
Z	- Variável lingüística zero
P	- Variável lingüística positivo
A	- Variável lingüística aumenta
M	- Variável lingüística mantém
D	- Variável lingüística diminui
G	- Variável lingüística grande
R	- Regra <i>fuzzy</i>
S	- Norma S ou conorma T
T	- Norma t
U	- Universo do discurso
u_A	- Defuzificação pelo método do centro da área
u_{CG}	- Defuzificação pelo método do centro da gravidade
u_{MM}	- Defuzificação pelo cálculo da média do máximo

u_{SUGENO}	- Defuzificação para regras do tipo Sugeno
α	- Valor de pertinência
μ	- Função de pertinência
\emptyset	- Função de ativação
Φ	- Função de implicação <i>fuzzy</i>
FS	- Fator de escala
W_n	- Frequência natural
K	- Rigidez global
Kv	- Fator de escala

Cárdenas, N. I. J., 2001, "O uso de Controladores *Fuzzy* Adaptativos, Implementados em Microcontroladores, no Controle de Vibrações de Sistemas Mecânicos", Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia, MG.

RESUMO

Este trabalho trata do controle ativo de vibrações em sistemas dinâmicos utilizando a tecnologia *fuzzy* de controle implementado em microcontroladores. Esta é uma das áreas na engenharia mecânica em que o avanço tecnológico representado pelo desenvolvimento de novos materiais, novos sensores e atuadores, novas tecnologias de controle, proporcionam, a todo instante, novas possibilidades.

Na área de controle, por exemplo, a formulação heurística e o curto ciclo de desenvolvimento tem propiciado à tecnologia *fuzzy* de controle avanços significativos. Apesar de dispensarem modelos matemáticos precisos dos sistemas envolvidos, o projeto de um controlador *fuzzy* clássico exige o conhecimento a priori de um especialista sobre o comportamento do sistema que se deseja controlar. Para tentar contornar esta exigência surgiram nos últimos anos os controladores *fuzzy* adaptativos. Uma revisão sobre tais controladores é aqui apresentada, assim como é proposto um novo *mecanismo de adaptação* que modifica os ganhos dos sinais de entrada do controlador *fuzzy* tradicional. Este mecanismo é de natureza *fuzzy*, opera *on line* e em paralelo com o controlador *fuzzy*.

Uma outra razão para o sucesso experimentado pelos controladores *fuzzy* se deve à facilidade de implementação desta tecnologia em microcontroladores de baixo custo. Uma breve revisão sobre microcontroladores é também aqui apresentada.

Tendo em vista avaliar experimentalmente a metodologia de controle proposta e na perspectiva de acumular experiências no uso de materiais piezoelétricos como atuadores/sensores, assim como no emprego de microcontroladores, este trabalho apresenta e discute dois experimentos de controle de vibrações: o primeiro envolve uma mesa vibratória acionada por atuadores eletromagnéticos e o segundo uma viga flexível instrumentada por acelerômetros e comandada por atuadores piezoelétricos.

O trabalho termina apresentando uma discussão dos resultados experimentais obtidos, faz uma avaliação da metodologia de controle proposta e sugere alguns desdobramentos para o futuro.

Palavras chave: Lógica Fuzzy, Controle Adaptativo, Controle Ativo de Vibrações, Microcontroladores.

Cárdenas, N. I. J., 2001, "The use of Adaptatives Fuzzy Controllers Implemented in Microcontrollers for the Vibrations Control of Mechanical Systems", Master's Dissertation, Federal University of Uberlândia, Uberlândia, MG.

ABSTRACT

This work deals with the active control of vibrations in dynamic systems. This area of mechanical engineering is one in which the technological progress represented by the development of new materials, new sensors and actuators and new control technologies open interesting and challenging possibilities.

In the control area, the heuristic formulation and the short development cycle has been promoting significant progress to fuzzy control technology. These controllers do not need mathematical models of the involved systems to operate. A classic fuzzy controller project demands a priori, the knowledge of an expert about the system behavior. In the attempt to overcome this demand, in the last years, adaptive fuzzy controllers were developed. A review of such controllers as well as a new adaptation mechanism that modifies the fuzzy controller gains are presented in this work. The nature of this mechanism is fuzzy and it operates on line and in parallel whit the fuzzy controller.

Another reason for the experienced success of fuzzy controllers is the easiness of implementation of this technology in microcontrollers of low cost. A brief review about microcontrollers is presented.

In order to evaluate experimentally the proposed control methodology and in the perspective of accumulating experience in the use of piezoelectric materials as actuators/sensors, as well as the microcontrollers employment, this work presents and discusses two applications of vibration control: the first one involves a vibratory system controlled by electromagnetic actuators and, the second is a flexible beam in which accelerometers are used as motion sensors and electromagnetic actuators are used for control purposes.

Finally, a discussion of the obtained experimental results is presented and the control methodology proposed is evaluated. Some future developments using the techniques presented are promoted to conclude the present contribution.

Key words: Fuzzy Logic, Adaptative Control, Vibration Control, Microcontrollers.

Capítulo I

Introdução

Um dos fenômenos mais importantes no estudo da dinâmica estrutural de sistemas mecânicos é o problema do controle das vibrações indesejáveis que tais sistemas eventualmente apresentam quando sujeitos a forças de excitações internas ou externas. Estas excitações podem provocar desde um simples desconforto sonoro até fadigas estruturais comprometedoras à integridade do sistema.

A evolução vertiginosa da informática e da eletrônica nos últimos anos tem possibilitado o aparecimento de novas tecnologias na área de controle de vibrações. De maneira geral, as técnicas de controle são divididas em dois grupos: técnicas de controle passivo e técnicas de controle ativo. Os aparatos especialmente projetados para absorver vibrações de máquinas e estruturas são, de forma genérica, denominados absorvedores dinâmicos de vibrações (ADVs). Os primeiros ADVs surgiram em 1911 (Frahm) e desde então inúmeros trabalhos com diferentes configurações de ADVs são encontrados na literatura.

Uma síntese bastante abrangente da teoria e prática dos ADVs é apresentada em (Koronev e Reznikov, 1993), (Klasztorny, 1995) e (Ram e Elhay, 1996). Luft (1979) utilizou os ADVs para atenuação de vibrações em edifícios, Nessler *et al.* (1997) usou para atenuar vibrações em máquinas ferramentas, Jones *et al.* (1981) empregou os ADVs para atenuação de vibrações em passarelas e Teodoro (1994) os utilizou para atenuação de vibrações em cabos de transmissão de energia.

Classicamente, os ADVs são projetados para minimizar vibrações geradas por uma excitação harmônica e sua eficiência se limita a um determinado intervalo de frequência das forças excitadoras. Este intervalo é definido pela escolha adequada dos parâmetros de um ADV – inércia, rigidez e amortecimento – (Cunha Jr., 1999). Duas estratégias principais têm sido exploradas para contornar esta limitação: a primeira consiste em determinar um conjunto de parâmetros invariante que garanta amplitudes mínimas de vibração em uma banda de frequência a mais larga possível.

A segunda estratégia consiste em conceber absorvedores adaptativos, cujos parâmetros podem ser contínua e automaticamente alterados para manter a sintonização. Na construção de ADVs adaptativos, os chamados materiais inteligentes têm sido utilizados com muito

sucesso, incluindo materiais piezoelétricos, materiais com memória de forma, etc. Um amplo levantamento sobre o emprego de ADVs adaptativos é feito por Sun *et al.* (1995), Marques (2000) e uma configuração controlada hidraulicamente é avaliada por Hrovat *et al.* (1983).

Uma das dificuldades no projeto deste tipo de controlador decorre da necessidade do conhecimento prévio do modelo matemático da planta a ser controlada. Em alguns casos esta informação é muito complexa ou de difícil obtenção. Apesar disto, informações verbais, ou lingüísticas, provenientes da experiência do operador sobre o processo, bem como sobre os critérios de desempenho desejados, são quase sempre conhecidas. Estas informações, embora imprecisas, revelam aspectos importantes sobre o comportamento do sistema. As incertezas ou dificuldades na definição dos modelos físicos e a disponibilidade de um "conhecimento intuitivo" sobre o comportamento do sistema sugerem o uso de controladores *fuzzy* neste tipo de problema. Nos últimos anos, este tipo de informação tem sido aplicada com sucesso no controle de sistemas utilizando-se a teoria dos conjuntos *fuzzy* (Nebuloso). É neste contexto que o presente trabalho se insere.

A utilização dos conceitos da lógica *fuzzy* na área de controle tem como base os trabalhos de Lofti A. Zadeh (1965). Os controladores *fuzzy* se aplicam em áreas onde decisões objetivas precisam ser tomadas a partir de um conjunto de informações muitas vezes vagas, imprecisas e não adequadamente quantificadas. As razões para tanto interesse se concentram em algumas características básicas desta tecnologia, quais sejam: (i) é natural e intuitiva na sua formulação, pois tenta imitar o comportamento consciente ou a estratégia de controle de um operador humano; (ii) dispensa o conhecimento detalhado dos modelos dos elementos do processo a ser controlado (planta, sensores, atuadores, etc.); (iii) se aplica a sistemas lineares e não lineares; (iv) é de implementação fácil e de baixo custo e (v) apresenta boas características de robustez a incertezas ou variações paramétricas.

Em algumas situações, no entanto, o especialista não dispõe de conhecimento intuitivo suficiente que lhe permita assegurar ou prever o comportamento do sistema com confiança. Algumas modificações no processo podem estar fora do campo de experiência do especialista. Isto propiciou o aparecimento dos controladores *fuzzy* adaptativos. Tais controladores têm a capacidade de modificar o seu comportamento em resposta a mudanças nas dinâmicas do processo e/ou perturbações. Uma introdução ao controle adaptativo pode ser encontrada em Ånström (1989). De maneira geral, os controladores adaptativos possuem duas componentes a mais que os controladores convencionais. O primeiro denominado *monitor de processo* que detecta a mudança nas características do processo e o segundo componente é o *mecanismo de adaptação* propriamente dito (Driankov, 1995). Yester *et al.* (1992) empregaram a lógica *fuzzy* adaptativa num sistema de suspensão automotiva que incorporava elementos ativos, Teixeira e Ribeiro (2001) usam a lógica *fuzzy* no controle de vibrações de um sistema massa-

mola a partir de atuadores eletromagnéticos, Yamashita (1998) faz uso da controle *fuzzy* adaptativo no controle de um reator de fluxo de gás, etc.

Em paralelo ao desenvolvimento das teorias de controle *fuzzy*, a última década experimentou um avanço vertiginoso na área da eletrônica, particularmente no que toca ao desenvolvimento da informática. Evoluíram significativamente o *hardware* e *software* dos microcontroladores, que são pequenos *chips* com elevada capacidade de processamento e alto grau de integração, que reúne numa única pastilha memórias, conversores A/D, D/A, timers, etc. (Garcia, 1998).

Diante deste cenário o presente trabalho se propõe a investigar o uso de controladores fuzzy adaptativos, implementado em microcontroladores e voltados para o problema de controle de vibrações de sistemas mecânicos.

Este trabalho está assim organizado: o capítulo II discute os fundamentos da lógica e do controle *fuzzy*, o capítulo III apresenta a teoria de controle *fuzzy* adaptativo e a metodologia de adaptação proposta, o capítulo IV discute os conceitos fundamentais, e a estrutura de um microcontrolador, o capítulo V apresenta os resultados experimentais obtidos ao se utilizar a estratégia proposta no controle de uma mesa vibratória (atuador eletromagnético) e de uma viga engastada (atuador piezoelétrico) e, finalmente, o capítulo VI discorre sobre as conclusões e os desdobramentos do presente trabalho.

Capítulo II

Lógica Fuzzy

2.1 Introdução

Este capítulo apresenta os conceitos fundamentais da lógica *fuzzy* e como tais conceitos podem ser usados no projeto de controladores *fuzzy*. O conceito de conjunto *fuzzy* foi introduzido, em 1965, por Lotfi A. Zadeh (Universidade da Califórnia, *Berkeley*).

Em meados da década de 60, Zadeh observou que os recursos tecnológicos disponíveis eram incapazes de automatizar as atividades de natureza industrial, biológica ou química, que compreendessem situações ambíguas, não passíveis de processamento, através da lógica computacional booleana. Procurando solucionar esse problema o Prof. Zadeh publicou em 1965 um artigo resumindo os conceitos dos conjuntos *fuzzy* (*Fuzzy Sets*), revolucionando o assunto com a definição dos sistemas *fuzzy*.

Em 1974, o Prof. Mamdani, do *Queen Mary College*, Universidade de Londres, após inúmeras tentativas frustradas em controlar uma máquina a vapor com tipos distintos de controladores, incluindo o PID, somente conseguiu fazê-lo através da aplicação do raciocínio *fuzzy*. Esse sucesso serviu de alavanca para muitas outras aplicações, como por exemplo o controle *fuzzy* de operação de um forno de cimento em 1980. Vieram em seguida várias outras aplicações destacando-se, por exemplo, os controladores *fuzzy* de plantas nucleares, refinarias, processos biológicos e químicos, trocadores de calor, máquinas diesel, tratamento de água e sistemas de operação automática de trens.

2.2 Conjuntos *fuzzy*

Ocorrem situações *fuzzy* (“*nebulosas*”) quando as informações sobre um processo não são muito claras ou são passíveis de diferentes interpretações. Definições como caro, barato, alto, baixo, quente, etc. são situações *fuzzy*. Ao se estabelecer, por exemplo, o seguinte padrão:

Uma roupa que custa 50 reais ou mais é considerada cara.

Uma roupa que custa 20 reais ou menos é considerada barata.

Ao fazermos a seguinte pergunta: Esta roupa é cara? A resposta deve ser: Se a roupa custar 50 reais ou mais é cara, se custar 20 reais ou menos é barata. Mas se a roupa tiver um preço entre 21 e 49 reais ela não pode ser considerada nem cara, nem barata, mas pode-se estabelecer um grau de pertinência quanto a ser cara ou barata.

A representação desta classe de informações pode ser tratada a partir da teoria dos conjuntos *fuzzy*. Na teoria clássica de conjuntos a pertinência de um elemento a um conjunto é claramente descrita, ou o elemento pertence a um conjunto ou não pertence. Na teoria dos conjuntos *fuzzy* a pertinência de um elemento a um determinado conjunto é parcial, possuindo um certo grau (possibilidade) de pertinência.

Assim um conceito fundamental na teoria dos conjuntos *fuzzy* é o *grau de pertinência*. O grau de pertinência de um determinado objeto a um conjunto *fuzzy* é um número real definido no intervalo $[0,1]$ que representa quão *verdadeiro* é a afirmação de que um certo elemento pertence a um dado conjunto. Um conjunto *fuzzy* "A" pode ser caracterizado por sua função característica, também denominada função de pertinência e denotada por $\mu_A(x)$, assim definida:

$$\mu_A(X) : U \rightarrow [0,1] \quad (2.1)$$

onde U é denominado "*universo do discurso*" e definido como o espaço ou domínio das variáveis deste conjunto. Formalmente o conjunto *fuzzy* "A" pode ser descrito como um conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)), x \in U\} \quad (2.2)$$

A Figura 2.1 apresenta uma função de pertinência para um conjunto nebuloso denominado "*temperatura baixa*".

Nesta função, temperaturas entre 0 e 10°C possuem valores de pertinência $\mu(x)=1.0$, significando que estas temperaturas tem total compatibilidade com o conceito de temperatura baixa. Temperaturas maiores que 10°C e menores que 200°C apresentam valores de pertinência decrescentes, representando um decaimento com o conceito de temperatura baixa. Temperaturas acima de 200°C possuem graus de pertinência $\mu(x)=0.0$, isto é, são valores considerados incompatíveis com o conceito temperatura baixa.

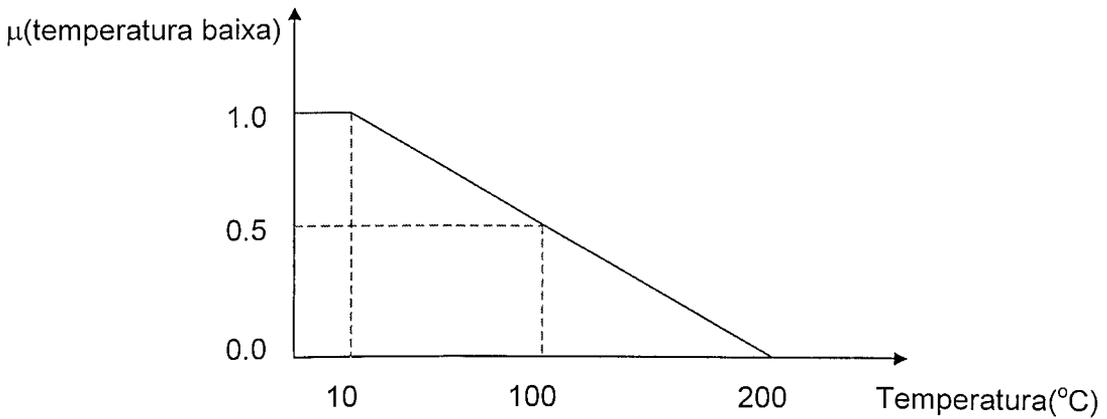


Figura 2.1 - Função de pertinência para o conjunto “Temperatura - baixa”

2.3 Operações sobre Conjuntos Fuzzy

Do mesmo modo que a teoria clássica de conjuntos permite efetuar operações de complemento, interseção e união, assim também as permite a teoria dos conjuntos *fuzzy*. A união e interseção de conjuntos *fuzzy* são definidas a partir dos valores de pertinência das funções características dos conjuntos. Zadeh (1965) propõe as seguintes definições para a união e para interseção:

$$\mu_A \cup \mu_B = \max [\mu_A(x) ; \mu_B(x)] \tag{2.3}$$

$$\mu_A \cap \mu_B = \min [\mu_A(x) ; \mu_B(x)] \tag{2.4}$$

Dado os conjuntos *fuzzy* “A” e “B” diz-se que “B” é o complemento de “A” se:

$$\mu_{\bar{a}}(x) = 1 - \mu_a(x) \tag{2.5}$$

As operações que podem ser aplicadas aos conjuntos *fuzzy* não se limitam ao máximo e mínimo, mas formam uma família de normas, denominadas normas triangulares (norma-T) e co-normas triangulares (conorma-T ou norma-S).

A norma-T é uma função definida de \mathfrak{R}^2 em \mathfrak{R}^1 , dentro dos domínios: $[0, 1] \times [0, 1] \rightarrow [0, 1]$, tal que $\forall a, b, c \in [0, 1]$, se verificam as seguintes propriedades:

$$1) T(a,0) = 0; \quad T(a,1) = a \tag{2.6}$$

$$2) T(a,b) = T(b,a) \tag{2.7}$$

$$3) T[T(a,b), c] = T[Ta, T(b,c)] \tag{2.8}$$

$$4) \text{ Se } a \leq b \text{ então } T(a, c) \leq T(b, c) \tag{2.9}$$

Uma co-norma triangular ou norma-S é uma função definida de \mathfrak{R}^2 para \mathfrak{R}^1 , dentro dos domínios: $[0, 1] \times [0, 1] \rightarrow [0, 1]$, tal que $\forall a, b, c \in [0, 1]$, se verificam as propriedades:

$$1) S(a,0) = a; \quad S(a,1) = 1 \tag{2.10}$$

$$2) S(a,b) = S(b,a) \tag{2.11}$$

$$3) S[S(a,b), c] = S[Sa, S(b,c)] \tag{2.12}$$

$$4) \text{ Se } a \leq b \text{ então } S(a, c) \leq S(b, c) \tag{2.13}$$

Vários autores propuseram diferentes definições para a interseção e a conjunção de conjuntos *fuzzy*, todos obedecendo às normas T e S acima definidas (Tabela 2.1).

Tabela 2.1 - Operações de união (norma-T) e de interseção (norma-S) de dois conjuntos fuzzy

Interseção T(a, b)	Conjunção S(a, b)	Condição	Referência
$T = \min(a,b)$	$S = \max(a,b)$		Zadeh (1965)
$T = a \cdot b$	$S = a + b - a \cdot b$		Bandler e Kohout (1980)
$T = \max(a + b - 1, 0)$	$S = \min(a + b, 1)$		Giles, Lukasiewicz (1976)
$T = \frac{a \cdot b}{\gamma + (1 - \gamma)(a + b - ab)}$	$S = \frac{a + b - (2 - \gamma) a \cdot b}{1 - (1 - \gamma) ab}$	$\gamma > 0$	Hamacher (1978)
$T = \frac{a \cdot b}{\max(a,b,\alpha)}$	$S = \frac{a + b - ab - \min(a,b,1-\alpha)}{\max(1-a,1-b,\alpha)}$	$\alpha \in [0,1]$	Dubois e Prade (1986)
$T = \max(1 - \sqrt[p]{(1-a)^p + (1-b)^p}, 0)$	$S = \min(\sqrt[p]{a^p + b^p}, 1)$	$p \geq 1$	Yager (1980b)
$T = \max\left(\frac{a + b - 1 + \lambda ab}{1 + \lambda}, 0\right)$	$S = \max(a + b + \lambda ab, 1)$	$\lambda \geq 1$	Weber (1993)

Para ilustrar os operadores descritos, as tabela 2.2 e 2.3 mostram o uso das normas de interseção e conjunção com os operadores de Zadeh.

Tabela 2.2 - Norma T de Zadeh

		a				
		0	.25	.50	.75	1
b	0	0	0	0	0	0
	.25	0	.25	.25	.25	.25
	.50	0	.25	.50	.50	.50
	.75	0	.25	.50	.75	.75
	1	0	.25	.50	.75	1

Tabela 2.3 - Norma S de Zadeh

		a				
		0	.25	.50	.75	1
b	0	0	.25	.50	.75	1
	.25	.25	.25	.50	.75	1
	.50	.50	.50	.50	.75	1
	.75	.75	.75	.75	.75	1
	1	1	1	1	1	1

2.4 Proposições e Operadores Lógicos Fuzzy

Outro conceito importante da lógica *fuzzy* é a proposição *fuzzy*. Seja uma frase da forma “X é A” onde “X” é o nome de uma variável lingüística e “A” um subconjunto *fuzzy* definido no universo de discurso “U”. Por exemplo:

$$\text{“a temperatura é baixa”} \tag{2.14}$$

onde “baixa” representa um rótulo lingüístico definido por um subconjunto *fuzzy* no universo do discurso da variável lingüística “temperatura”.

Tal como na lógica clássica, as proposições *fuzzy* podem ser combinadas pelo uso de operadores lógicos “e” e “ou”, como por exemplo:

$$\text{“a temperatura é baixa e a umidade é alta”} \tag{2.15}$$

Para combinar as proposições “e” e “ou”, a lógica *fuzzy* faz uso das normas-T e das normas-S, respectivamente. Como já foi mostrado na tabela 2.1 existem várias maneiras de aplicar estas normas. Os operadores mais utilizados na lógica *fuzzy* são apresentados na tabela 2.4.

Tabela 2.4 - Operadores comumente utilizados para os conectivos “e” e “ou” na lógica *fuzzy*

“e”	“ou”	Referência
$\min(a, b)$	$\max(a, b)$	Zadeh (1965)
$\max(a+b-1, 0)$	$\min(a+b, 1)$	Giles (1976)
$a \cdot b$	$a+b-a \cdot b$	Bandler (1980)

A figura 2.2 ilustra o uso destes operadores para as funções A e B.

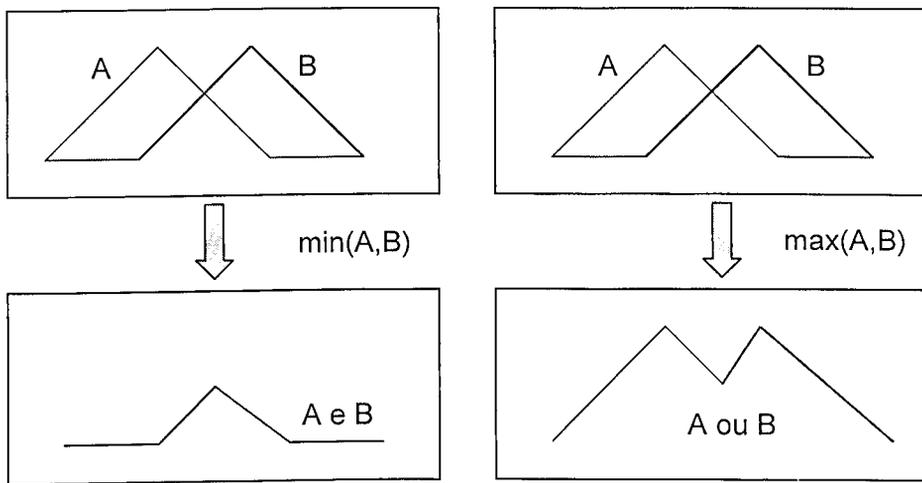


Figura 2.2 - Uso dos operadores “e” e “ou”

2.5 Métodos de inferência de regras *fuzzy*

Nas descrições matemáticas elementares, uma entrada “x” pode ser inferida em uma função $y=f(x)$ para a obtenção de “y”. Similarmente em descrições lingüísticas *fuzzy* uma entrada pode ser inferida para a obtenção da saída. Este mecanismo de inferência está assentado num conjunto de condições lingüísticas do tipo “se...então” que configura uma base de regras. O processo de avaliação de uma descrição lingüística, contida em uma base de regras, é chamado de inferência composicional *fuzzy*.

Normalmente, a inferência composicional da base de regras *fuzzy* consiste em, num primeiro instante, avaliar as regras através de uma relação chamada de implicação de regra, e depois agregar todas elas.

Cada regra é representada por uma afirmação “se - então”, onde o antecedente e o conseqüente consiste de proposições combinados pelos operadores lógicos “e” e “ou”. Considerando a seguinte regra:

$$\text{Se } x_1 \text{ é } A_1 \text{ e } x_2 \text{ é } A_2 \text{ então } y \text{ é } B \quad (2.16)$$

onde “ A_1 ”, “ A_2 ” e “B” são conjuntos *fuzzy*. A seguinte relação *fuzzy* pode ser construída para representar essa regra:

$$R = \Phi (T(A_1, A_2), B) \tag{2.17}$$

onde T é uma norma-T e Φ é uma função de implicação *fuzzy*. Assim como a norma-T modela o conectivo “e”, a implicação Φ modela a sentença “se - então”.

As regras *fuzzy* podem também ser representadas por meio das funções de pertinência envolvidas:

$$\mu_R(x_1, x_2, y_1) = \Phi \{T[\mu_{A_1}(x_1), \mu_{A_2}(x_2)]; \mu_B(y_1)\} \tag{2.18}$$

As funções de implicação *fuzzy* são normalmente denotadas por $\Phi(a,b)$ quando $a,b \in [0,1]$. Existem várias maneiras de representar as implicações *fuzzy*. A tabela 2.7 apresenta as implicações freqüentemente encontradas em aplicações de engenharia, particularmente no projeto de controladores *fuzzy*.

Tabela 2.5 - Implicações *fuzzy* entre a e b

$\Phi(a, b)$	Referências
$\min [1-a; \min(a,b)]$	Zadeh (1973)
$\min [1-a+b; 1]$	Giles (1976)
$\max [1-a; b]$	Kleene (1980)
$1-a+ab$	Reichenbach (1999)
b, se $a > b$ ou: 1, caso contrário	Godel (1999)
1, se $a = 0$ ou: $\min(ba^{-1}; 1)$, caso contrário	Goguem (1969)
1, se $a < b$ ou: $\min(1-a; b)$, caso contrário	Wu (1986)
b^a	Yager (1980a)
$\text{Min}\{ \max(1-a; b); \max[a, 1-b; \min(b, 1-a)]\}$	Willmott (1980)
$\min (a, b)$	Mamdani (1974)
$a*b$	Larsen (1980)

Uma vez definidos os elementos essenciais da lógica *fuzzy* vejamos como tais elementos podem ser utilizados no projeto de controladores *fuzzy*.

2.6 Controle a partir da Lógica Fuzzy

A lógica *fuzzy* pode ser utilizada para a implementação de controladores *fuzzy* aplicados aos mais variados tipos de processos (Lee, 1980). Os controladores *fuzzy* apresentam, quando comparados aos controladores convencionais, algumas vantagens, como por exemplo: o controlador é formulado a partir de um conhecimento intuitivo do sistema e utiliza descritores lingüísticos no seu projeto; dispensa a formulação matemática do modelo do sistema a ser controlado; aplica-se a sistemas lineares e não lineares; é de implementação fácil e de baixo custo e tem um ciclo de projeto e desenvolvimento mais rápido.

Das vantagens acima, uma das que mais contribui para a sua aceitação é a característica da facilidade de implementação associada ao ciclo de desenvolvimento relativamente curto. A figura 2.3 ilustra uma seqüência de projeto para o desenvolvimento de um controlador, usando a técnica convencional e a lógica *fuzzy*.

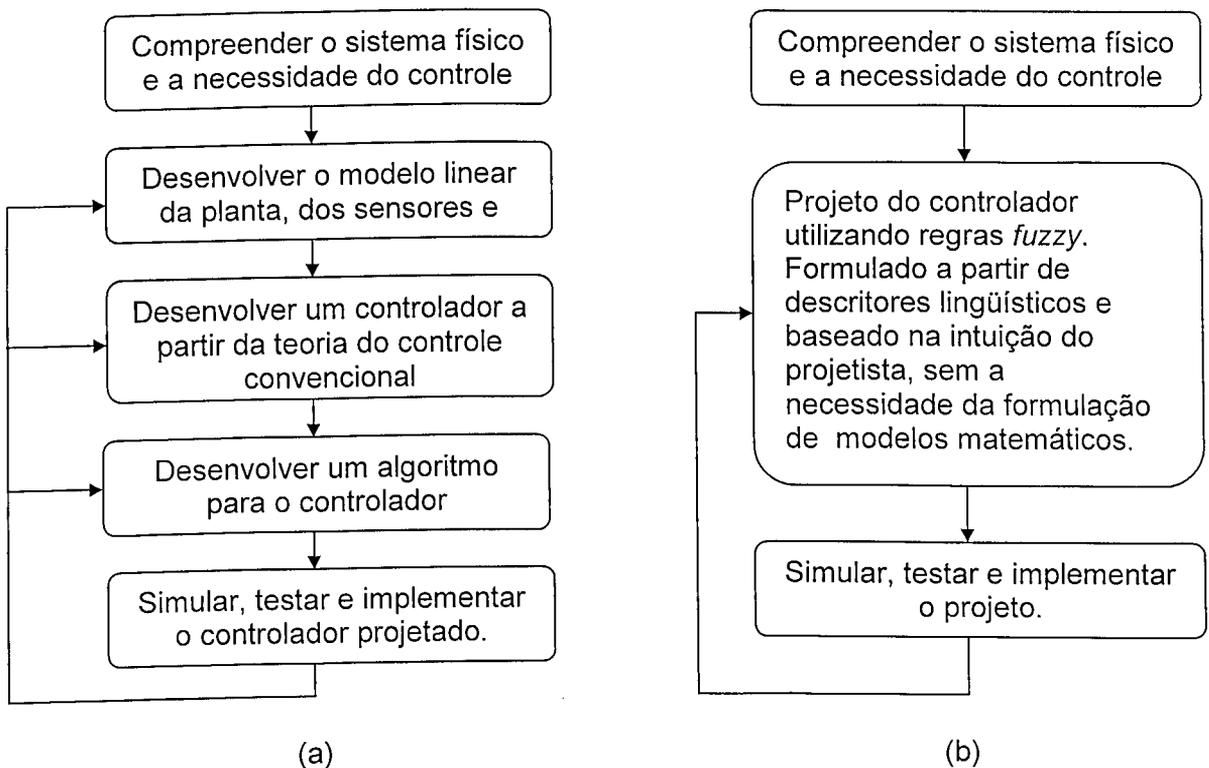


Figura 2.3 - Etapas de um projeto usando controle convencional (a) e controle *fuzzy* (b)

Como, as entradas e saídas de controladores são normalmente valores numéricos há necessidade de transformar tais variáveis em variáveis *fuzzy* numéricas. Estas transformações

são chamadas de *fuzzificação* e *defuzzificação*. Este é um dos primeiros passos na construção de um controlador *fuzzy*.

A metodologia de projeto de um controlador *fuzzy* envolve também a construção de uma base de regras, que utiliza operadores lógicos e funções de pertinência. O controlador *fuzzy* essencialmente mapeia, por meio de um processo de inferência, variáveis de entrada em variáveis de saída.

O projeto de controladores, utilizando a lógica *fuzzy*, obedece normalmente as seguintes etapas:

Etapa 1: caracterização do intervalo de valores que as variáveis de entrada e as variáveis de saída podem assumir (definição do universo do discurso);

Etapa 2: definição de um conjunto de funções (as funções de pertinência) que mapeiam as variáveis de entrada e de saída do universo do discurso;

Etapa 3: definição de um conjunto de regras, usando operadores lógicos e descritores lingüísticos, que buscam estabelecer uma relação entre os valores de entrada e os valores de saída;

Etapa 4: uma vez obtidas as regras, passa-se à fase de tradução matemática da linguagem simbólica construída. Isto é realizado através da utilização de operadores lógicos definidos pela teoria dos conjuntos *fuzzy*. Para tanto são necessárias três sub-etapas distintas: a *fuzzificação* das entradas, a aplicação de um mecanismo composicional de inferência e a *defuzzificação* das saídas.

A figura 2.4 mostra os componentes básicos de um controlador *fuzzy*.

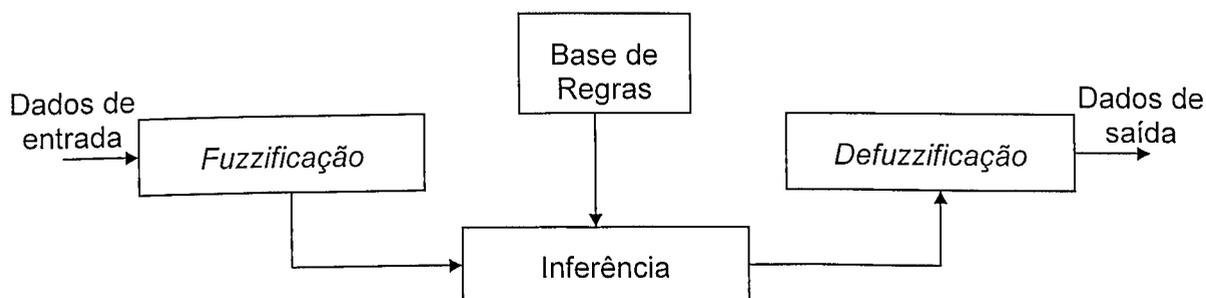


Figura 2.4 - Controlador *Fuzzy* com seus componentes básicos

Estes componentes serão descritos detalhadamente para um melhor entendimento.

2.6.1 Fuzzificação das entradas

A etapa de fuzzificação de um controlador transforma os valores reais das variáveis de entrada, medidos ou estimados, em graus de pertinência de um determinado conjunto *fuzzy*.

Para exemplificar esta etapa, considere um problema de duas entradas erro e Δ erro, como mostrado na figura 2.5. Também considere a seguinte regra:

$$r_1: \text{ Se erro é N e } \Delta\text{erro é P Então ...} \quad (2.19)$$

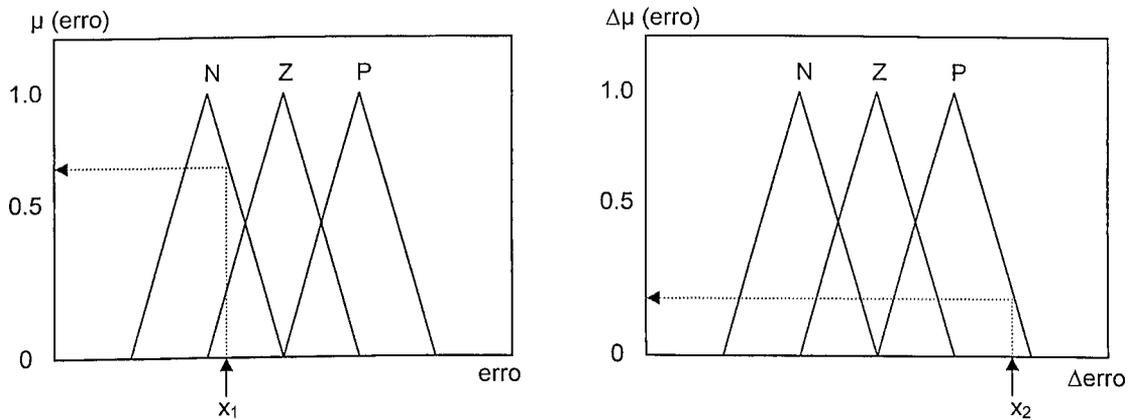


Figura 2.5 - Representação gráfica da fuzzificação para as entradas x_1 e x_2

De acordo com a figura 2.5, tem-se:

$$r_1: \text{ Se erro} = x_1 \text{ é N e } \Delta\text{erro} = x_2 \text{ é P Então ...} \quad (2.20)$$

$$r_1: \text{ Se } \mu_N(x_1) = 0.6 \text{ e } \mu_P(x_2) = 0.2 \text{ Então ...} \quad (2.21)$$

Assim a etapa de fuzzificação das entradas é o cálculo dos antecedentes das regras, através da conversão de valores numéricos em valores de pertinência dos conjuntos enunciados nas regras.

2.6.2 Inferência da base de regras

Os mecanismos mais comuns de inferência aplicados na construção de controladores fuzzy são:

(a) Método composicional de inferência max-min: Este método consiste em escolher o operador *min* (interseção) para as implicações fuzzy e o operador *max* (união) para a agregação do conjunto de regras. Vejamos estes conceitos por meio de um exemplo. Considere um controlador que possua duas entradas e uma saída, e a seguinte base de regras:

$$r_1: \text{ Se erro} = x_1 \text{ é N e } \Delta\text{erro} = x_2 \text{ é P Então a saída é } u_c = P \quad (2.22)$$

$$r_2: \text{ Se erro} = x_1 \text{ é N e } \Delta\text{erro} = x_2 \text{ é Z Então a saída é } u_c = Z \quad (2.23)$$

Faz-se inicialmente a fuzzificação do sistema a partir do antecedente de cada regra. Supondo que para o conectivo “e” utilize-se a formulação proposta por Zadeh (veja tabela 2.4)

e que para a implicação adote-se a proposta de Mamdani (tabela 2.5), tem-se, de acordo com a figura 2.6:

$$r_1 : \text{Se } \mu_N(\text{erro}) = 0.6 \text{ e } \mu_P(\Delta\text{erro}) = 0.2 \text{ Então a saída } u_c \text{ é } \min[0.2; \mu_P(u_c)] \quad (2.24)$$

$$r_2 : \text{Se } \mu_N(\text{erro}) = 0.6 \text{ e } \mu_Z(\Delta\text{erro}) = 0.5 \text{ Então a saída } u_c \text{ é } \min[0.5; \mu_Z(u_c)] \quad (2.25)$$

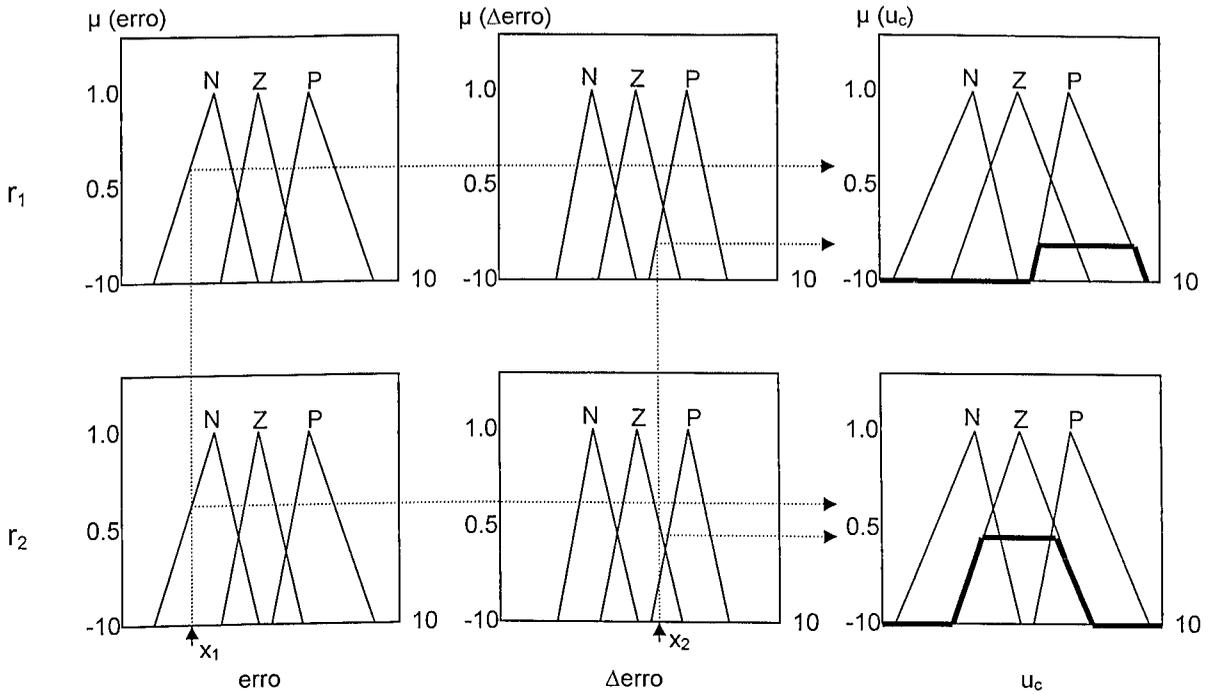


Figura 2.6 - Cálculo dos antecedentes e dos conseqüentes a partir da inferência *max-min*

A agregação dos conseqüentes é feito observando o “máximo” das inferências de cada uma das regras, como mostra a figura 2.7.

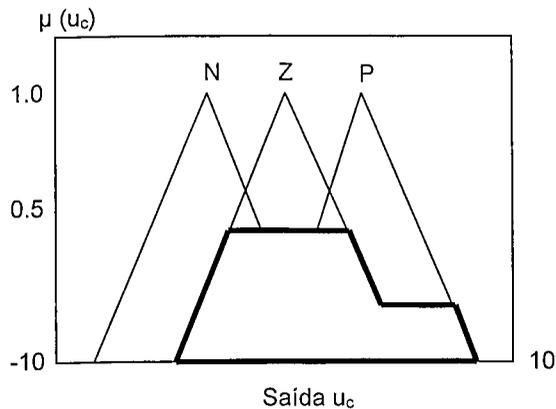


Figura 2.7 - Agregação dos conseqüentes a partir da inferência *max-min*

(b) Método composicional de inferência max-prod: O método max-prod foi proposto por Larsen (1980). Este método consiste em escolher o operador prod (Larsen) para as implicações fuzzy e o operador max (união) para a agregação.

Suponha novamente que um controlador fuzzy possua duas entradas e uma saída, e para obter a lei de controle é necessário a seguinte base de regras:

$$r_1 : \text{Se erro} = x_1 \text{ é N e } \Delta\text{erro} = x_2 \text{ é P Então a saída é } u_c = P \quad (2.26)$$

$$r_2 : \text{Se erro} = x_1 \text{ é N e } \Delta\text{erro} = x_2 \text{ é Z Então a saída é } u_c = Z \quad (2.27)$$

Inicialmente é feito a fuzzificação do antecedente de cada regra. Como as implicações são do tipo Larsen, é utilizado o operador “e” como o produto dos graus de pertinência do antecedente, resultando num número “crisp”, e o produto deste número “crisp” pelo conjunto fuzzy da consequente é o resultado da regra. A inferência destas duas regras é obtida de acordo com a figura 2.8:

$$r_1 : \text{Se } \mu_N(\text{erro}) = 0.6 \text{ e } \mu_P(\Delta\text{erro}) = 0.2 \text{ Então a saída } u_c \text{ é } 0.12 \cdot \mu_P(u_c) \quad (2.28)$$

$$r_2 : \text{Se } \mu_N(\text{erro}) = 0.6 \text{ e } \mu_Z(\Delta\text{erro}) = 0.5 \text{ Então a saída } u_c \text{ é } 0.30 \cdot \mu_Z(u_c) \quad (2.29)$$

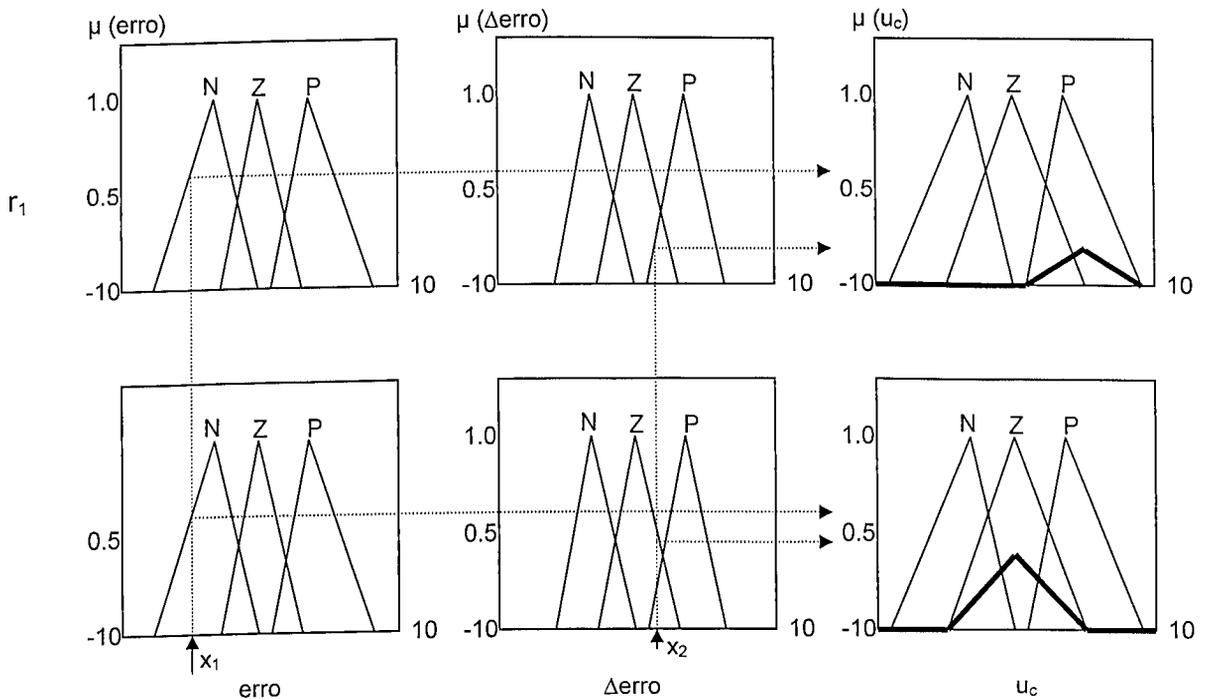


Figura 2.8 - Cálculo dos antecedentes e conseqüentes a partir da inferência max-prod

A figura 2.9 ilustra a agregação de todos os conseqüentes, ou seja, a união dos subconjuntos obtidos em cada regra pelo método de inferência composicional *max-prod*.

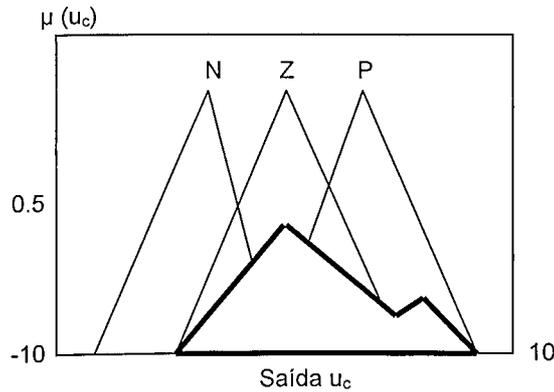


Figura 2.9 - Agregação dos conseqüentes a partir da inferência *max-prod*

(c) Método composicional de inferência Sugeno: Este método foi proposto por Sugeno (1985), as regras deste tipo de inferência tem a seguinte forma geral:

$$r_1: \text{ Se } x \text{ é } A \text{ e } y \text{ é } B \text{ Então } z = p \cdot x + q \cdot y + r \quad (2.30)$$

onde A e B são conjuntos *fuzzy* do antecedente e p, q e r são valores constantes.

Suponha que um controlador *fuzzy* possua duas entradas, uma saída e a seguinte base de regras:

$$R_1: \text{ Se } \text{erro} = x_1 \text{ é } N \text{ e } \Delta\text{erro} = x_2 \text{ é } P \text{ Então a saída é } u_c = P \quad (2.31)$$

$$R_2: \text{ Se } \text{erro} = x_1 \text{ é } N \text{ e } \Delta\text{erro} = x_2 \text{ é } Z \text{ Então a saída é } u_c = Z \quad (2.32)$$

Faz-se inicialmente a fuzzificação do sistema a partir do antecedente de cada regra. Para o conectivo “e” utilize-se uma das formulações propostas na tabela 2.4, neste exemplo é adotado a formulação do tipo Larsen. Para a implicação do antecedente com o conseqüente adote-se a proposta por Sugeno (equação 2.30). Assim, de acordo com a figura 2.10, tem-se:

$$r_1: \text{ Se } \mu_N(\text{erro}) = 0.6 \text{ e } \mu_P(\Delta\text{erro}) = 0.2 \text{ Então a saída } u_c \text{ é } (0.12 * 5) \quad (2.33)$$

$$r_2: \text{ Se } \mu_N(\text{erro}) = 0.6 \text{ e } \mu_Z(\Delta\text{erro}) = 0.5 \text{ Então a saída } u_c \text{ é } (0.30 * 0) \quad (2.34)$$

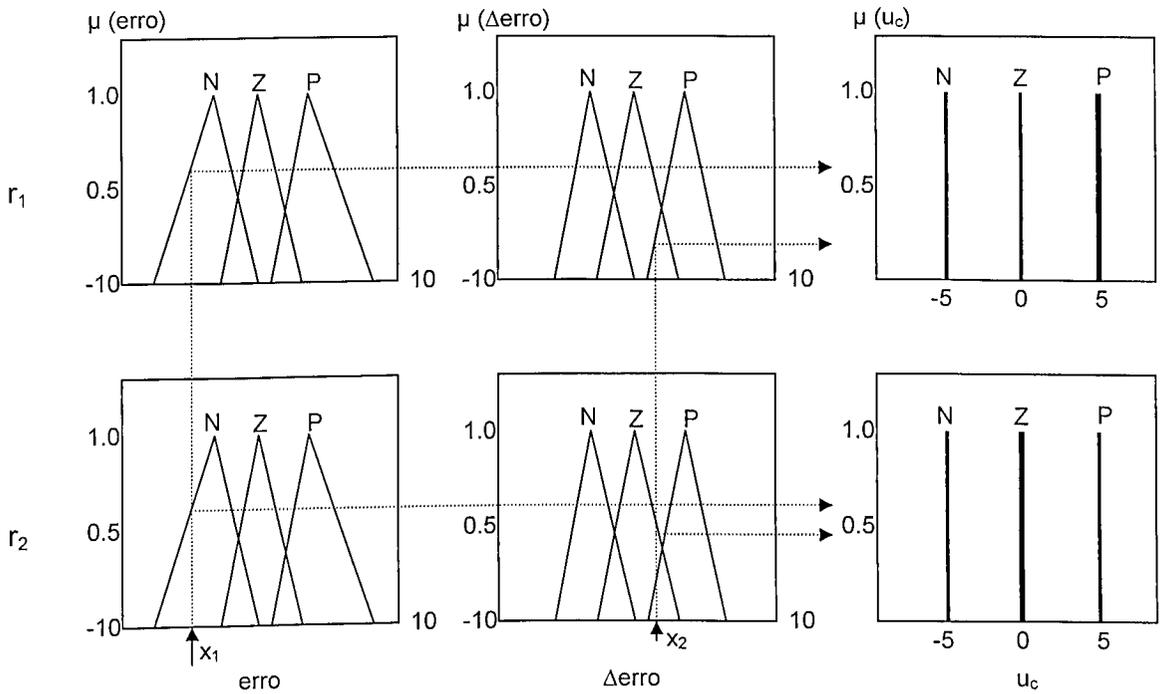


Figura 2.10 - Cálculo dos antecedentes e consequentes a partir da inferência Sugeno

Quando a base de regras é do tipo Sugeno, não é necessário a agregação dos conjuntos resultantes das implicações de cada regra.

2.6.3 Defuzzificação das saídas

Depois de feita a agregação, resta ao controlador *fuzzy* fazer a defuzzificação das saídas, isto é, transformar o resultado da inferência difusa em uma saída numérica. Muitos autores (Terano *et al.*, 1992; Pedrycz, 1992; Yager *et al.*, 1993) tem sugerido diferentes métodos para o cálculo da defuzzificação, a tabela 2.6 mostra alguns destes métodos.

Tabela 2.6 - Alguns métodos de defuzzificação

Função	Método
$u_{CG} = \frac{\sum_{i=1}^N u_i \cdot \mu_u(u_i)}{\sum_{i=1}^N \mu_u(u_i)}$	Centro de Gravidade (CG)

$u_{CA} = \frac{\sum_{i=1}^N u_i \cdot \sum_{k=1}^n \mu_{Bk}(u_i)}{\sum_{i=1}^N \sum_{k=1}^n \mu_{Bk}(u_i)}$	<p>Centro da Área (CA)</p>
$u_{MM} = \sum_{i=1}^M \frac{u_M}{M}$	<p>Cálculo da média do máximo (MM)</p>
$u_S = \frac{\sum_{i=1}^k u_i \cdot \mu_{u_i}}{\sum_{i=1}^k \mu_{u_i}}$	<p>Sugeno</p>

Para exemplificar esta etapa, será descrito o método de defuzzificação do centro de gravidade.

a) Método do Centro de gravidade (CG): Este método consiste no cálculo do centróide de figuras planas. Este cálculo é feito na região resultante da agregação de todas as regras, como mostrado nas figuras 2.7 e 2.9. O centro de gravidade é definido como:

$$u_{CG} = \frac{\sum_{i=1}^N u_i \cdot \mu_u(u_i)}{\sum_{i=1}^N \mu_u(u_i)} \tag{2.35}$$

onde o somatório é tomado sobre os valores discretos do universo do discurso de “u_i” amostrados em N pontos, μ_u(u_i) é o valor de pertinência para um dado valor de controle “u_i”. A figura 2.10 representa graficamente a defuzzificação pelo método do centro de gravidade.

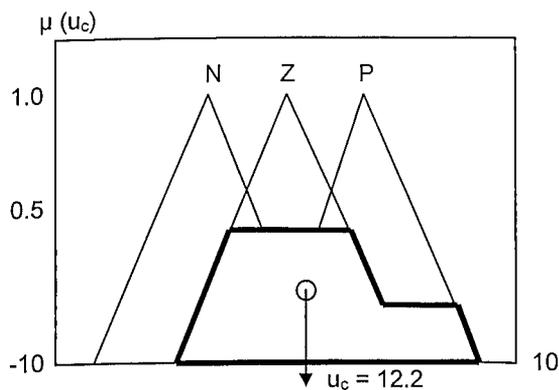


Figura 2.10 - Defuzzificação pelo método do centro de gravidade

Em algumas aplicações *fuzzy* a definição da base de regras não é um tarefa trivial e imediata. Influenciam também no desempenho do controlador *fuzzy* a escolha das funções de pertinência e do mecanismo de defuzzificação.

No próximo capítulo são descritos alguns mecanismos de adaptação dos controladores *fuzzy* que procuram ajustar “*on-line*” ou “*off-line*” alguns de seus parâmetros como a base de regras, as funções de pertinência, os fatores de escala, etc.

Capítulo III

Controle Adaptativo

3.1 Introdução

Um dos maiores desafios no projeto de controladores está na busca de soluções que apresentem simultaneamente boas características de robustez e excelência no desempenho. Os controladores *fuzzy*, a despeito de dispensarem o modelamento matemático clássico dos sistemas físicos, dependem do conhecimento a priori que um especialista possui a cerca do processo que se quer controlar. Em algumas situações é muito difícil, ao especialista, construir uma estratégia de controle confiável a partir dos enunciados lingüísticos que configuram a base de regras do controlador *fuzzy*. Algumas modificações no processo podem, inclusive, estarem fora do campo de experiência do especialista. Estas questões suscitaram nos últimos tempos o aparecimento dos controladores *fuzzy* adaptativos. São controladores que revelam alguma capacidade de modificar o seu comportamento em resposta à mudanças nas dinâmicas do processo e/ou perturbações. Este capítulo tem por objetivo apresentar os conceitos básicos dos controladores *fuzzy* adaptativos e também propor uma nova estratégia para a adaptação destes controladores.

3.2 Controle *fuzzy* adaptativo

Muitos dos processos do mundo real que requerem controle automático são de natureza não linear, com parâmetros dependentes das condições de operação. A necessidade de identificar estas mudanças e de se adaptar a novas condições acrescenta aos controladores *fuzzy* adaptativos dois componentes extras, que não estão presentes nos controladores *fuzzy* convencionais. O primeiro componente é o “monitor de processos” que detecta as mudanças nas características do processo. Este monitoramento pode ser feito de duas maneiras: (i) medindo o desempenho do sistema sob controle ou (ii) estimando continuamente alguns parâmetros do modelo do processo.

O segundo componente é responsável pelo “mecanismo de adaptação”. Este mecanismo pode modificar os parâmetros do controlador, para melhorar o desempenho com base nas

saídas do monitor do processo. A Figura 3.1 mostra o esquema de um controlador *fuzzy* adaptativo típico com todos seus componentes.

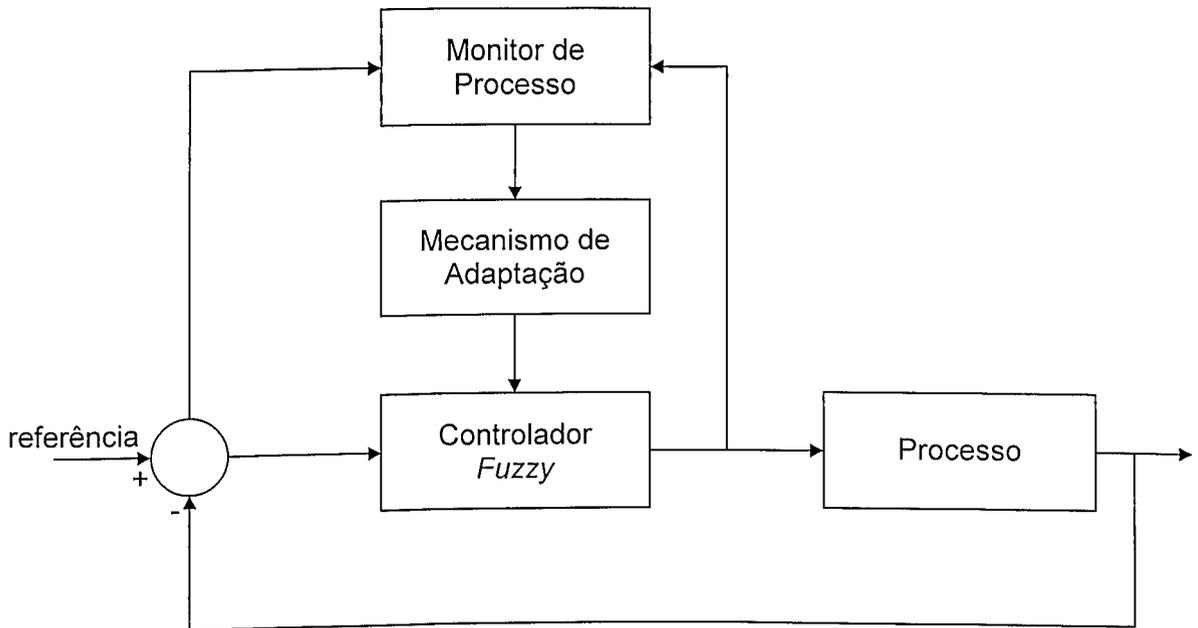


Figura 3.1 - Componentes de um controlador *Fuzzy* Adaptativo

Os controladores adaptativos podem ser classificados como adaptativos de desempenho ou adaptativos de parâmetro, dependendo do tipo de monitoramento utilizado. Ambos os tipos de monitoramento serão apresentados em detalhes mais adiante. Os parâmetros normalmente adaptados nos controladores *fuzzy* são:

- o fator de escala das variáveis de entrada e/ou saída;
- a “forma” dos conjuntos *fuzzy* e
- as regras “se – então”.

Na literatura os controladores *fuzzy* adaptativos que modificam a definição do conjunto *fuzzy* ou o fator de escala são chamados *controladores auto-sintonizados*, enquanto que os controladores adaptativos que modificam as regras são denominados de *controladores auto-organizados* (*Self-Organizing Controller - SOC*).

3.3 Elementos dos controladores adaptativos

Os controladores adaptativos, como descrito anteriormente, possuem dois componentes importantes, o monitor de processo e o mecanismo de adaptação.

3.3.1 Monitor de processo

O monitor de processo procura detectar mudanças importantes no comportamento da planta que apontem para a necessidade de uma reconfiguração nos parâmetros do controlador. Dois tipos de monitoramento se destacam: o primeiro observa alterações em variáveis paramétricas e o segundo monitora o desempenho do sistema controlado.

3.3.1.1 Monitoramento das variáveis paramétricas

Mudanças no processo podem ser detectadas *on line*, observando variações no tempo de parâmetros de um modelo inicialmente “ajustado” e tido como referência para a descrição do comportamento do sistema sob controle. Um modelo típico é:

$$y_{t+1} = \sum_{i=1}^n a_i(t)y_{t-i} + \sum_{i=1}^m b_i(t)u_{t-i} \quad (3.1)$$

onde y_t é a saída do processo, u_t é a entrada, $a_i(t)$, $b_i(t)$, m e n são inteiros naturais. Este processo demanda um esforço computacional apreciável uma vez que os parâmetros $a_i(t)$ e $b_i(t)$ são estimados o tempo todo e quaisquer modificações nos seus valores podem implicar em redefinições do controlador.

3.3.1.2 Monitoramento da medida de desempenho

Este tipo de monitoramento procura observar o desempenho do processo, monitorando algumas grandezas típicas, tais como:

overshoot;

tempo de subida;

tempo de acomodação;

freqüência de oscilação do transiente;

integral do erro quadrático;

integral do valor absoluto do erro;

margem de ganho e fase

etc.

A escolha de uma ou outra destas grandezas depende do tipo de resposta de controle que o projetista do sistema deseja alcançar.

3.3.2 Mecanismos de adaptação

O mecanismo de adaptação é especialmente projetado para alterar os parâmetros do controlador *fuzzy*. Os mecanismos de adaptação para os controladores *fuzzy* são classificados de acordo aos parâmetros ajustados:

3.3.2.1 Mecanismos de adaptação auto sintonizados

a) Escalonamento dos ganhos do fator de escala:

O fator de escala é um mecanismo de adaptação onde os valores da entrada e/ou saída do controlador *fuzzy* são multiplicados por um ganho. A Figura 3.2 ilustra a utilização deste mecanismo, onde a definição dos conjuntos é feita sobre um universo de discurso normalizado, isto é, num intervalo de -1.0 a 1.0.

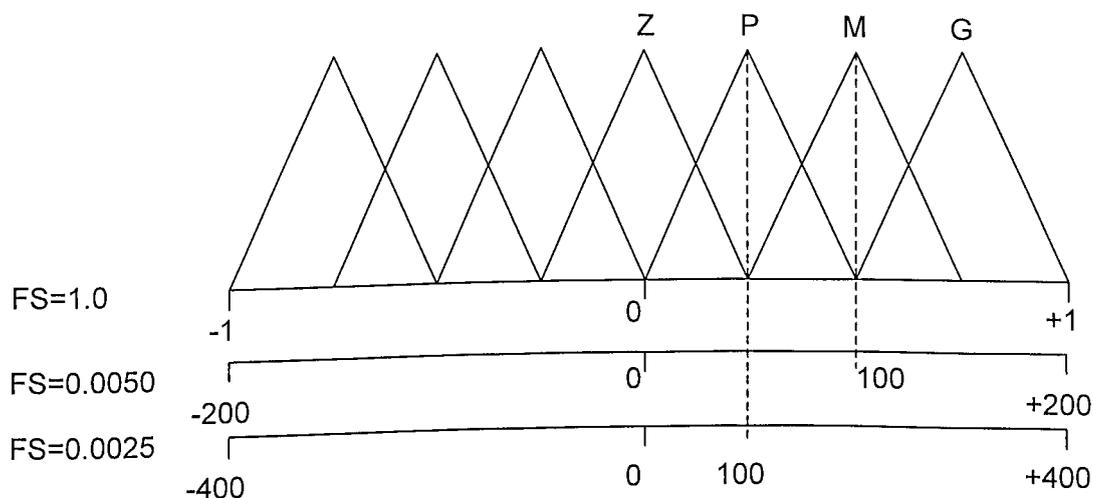


Figura 3.2 - Efeito de alterar o fator de escala

Supondo uma variável de entrada que varie em termos absolutos no intervalo de -200 a 200, quando se multiplica esta variável pelo fator de escala $FS=0.0050$, a mesma passa a ser mapeada no universo normalizado -1.0 a 1.0. Neste caso, o valor de entrada de 100 é classificado como "(M)édio". Para Alterar a classificação deste valor, basta multiplicar o valor da entrada por um novo fator escala, por exemplo, 0.0025, assim, o novo universo do discurso

terá um intervalo de -400 a 400 , e o valor de 100 será classificado como "(P)equeno" para este novo universo do discurso. Alterar, portanto o fator de escala significa modificar a sensibilidade do controlador.

Um exemplo deste mecanismo é dado por Yamashita *et al.* (1988), no controle da temperatura de um reator químico.

b) Ajustes dos conjuntos fuzzy:

Um outro mecanismo similar de sintonização é alterar a forma dos conjuntos *fuzzy* definindo o significado dos valores lingüísticos. Os conjuntos *fuzzy* não são escolhidos de forma arbitrária, mas procuram refletir um determinado significado lingüístico. Para modificar a forma dos conjuntos *fuzzy*, pequenas modificações podem ser feitas sem por em perigo o significado fundamental lingüístico. Um exemplo onde os conjuntos *fuzzy* são alterados para aumentar a sensibilidade do controlador para valores pequenos de entrada é mostrado na Figura 3.3.

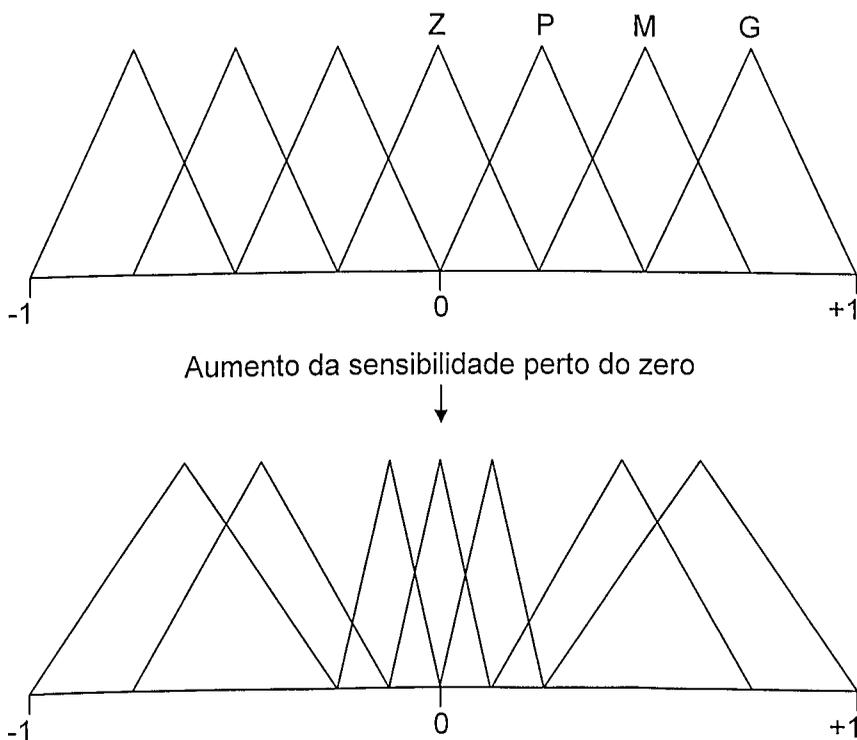


Figura 3.3 - Adaptação dos conjuntos *fuzzy* definidos

Trabalhos recentes tem se utilizado de técnicas de otimização matemática para alterar os conjuntos *fuzzy* definidos, de modo que a saída destes controladores se aproxime de uma saída de referência. Teixeira (2001), Maeda *et al.* (1991) e Nomura *et al.* (1991) propõem um

mecanismo “*off-line*” de ajuste destes parâmetros baseado em otimização genética enquanto Bartoloni *et al.* (1982) propõe um mecanismo para ajuste “*on-line*” baseado no monitoramento do erro médio quadrático.

3.3.2.2 Mecanismos de adaptação auto organizados

Reorganização do conjunto de regras (SOC):

Vários esquemas para controladores auto-organizados que modificam a base de regras têm sido desenvolvidos. Uma abordagem clássica foi proposta por Mamdani e seu grupo de pesquisa (Mamdani, 1975; Mamdani, 1976; Procyk, 1979). Uma descrição detalhada de controladores auto-organizados (SOC) é apresentada por Procyk (1979). A idéia deste tipo de controlador é identificar quais regras são responsáveis pelo desempenho pobre do controlador e assim substituí-las por regras melhores.

A modificação de regras envolve dois mecanismos: um para determinar qual regra deve ser alterada e outro de como alterar esta regra. Semelhante ao controlador de Bartolini *et al.* (1982) o controlador auto-organizado também é um controlador adaptativo de desempenho. A estrutura deste controlador é similar ao controlador mostrado na figura 3.1, porém este inclui no seu formalismo um modelo para o processo, como mostrado na figura 3.4, que retrata uma abordagem proposta por Graham e Newell (1986, 1988, 1989).

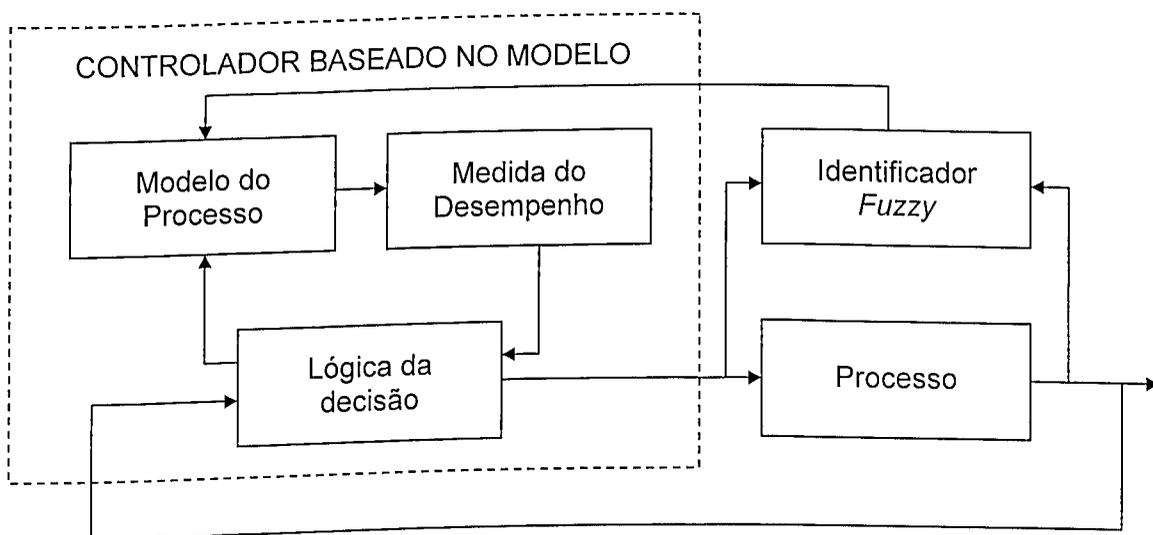


Figura 3.4 - Controlador *fuzzy* adaptativo baseado em modelo

3.4 Proposta de adaptação do universo do discurso a partir do fator de escala

O desempenho de um controlador *fuzzy*, como já mencionado, depende em grande parte da magnitude dos sinais de entrada e saída manipulados pelo controlador. Uma maneira de melhorar o desempenho do controlador, proporcionando-lhe certa adaptabilidade, pode ser construída a partir do monitoramento da performance do processo controlado associado ao ajuste “*on line*” dos fatores de escala dos sinais de entrada e saída do controlador. Propõe-se, neste trabalho, um novo mecanismo de adaptação baseado nestas idéias

O mecanismo de adaptação proposto constitui um “*controlador fuzzy secundário*” para os fatores de escala de um controlador *fuzzy* primário. Projeta-se o controlador *fuzzy* primário de maneira usual mas observando a normalização no intervalo [-1; 1] do universo do discurso das variáveis cujos ganhos se pretende ajustar. A base de regras deste controlador e as funções de pertinência dos sinais envolvidos devem atender ao conhecimento heurístico do projetista.

Para ilustrar o procedimento, seja um controlador *fuzzy* clássico dotado de duas entradas (erro e variação do erro) e uma saída. Suponha que se deseje melhorar o desempenho, modificando o fator de escala da variação do erro, por exemplo. Assim:

$$\Delta e(k) = \frac{e(k) - e(k-1)}{\Delta t} \cdot K_v(k) \quad (3.2)$$

onde $e(k)$ é o sinal do erro no instante k , $K_v(k)$ é o fator de escala da variação do erro e Δt é o tempo de amostragem.

Propõe-se neste trabalho que o fator de escala $K_v(k)$ obedeça a seguinte expressão:

$$K_v(k) = K_v(k-1) \cdot K_{K_v} \quad (3.3)$$

onde K_{K_v} é a saída do controlador *fuzzy* do fator de escala. Observe-se que quando K_{K_v} é maior que a unidade o ganho aumenta, tornando o sistema mais sensível o que pode levar a ações de controle mais elevadas.

O diagrama de blocos do controlador *fuzzy* com o mecanismo de adaptação do universo do discurso é mostrado na figura 3.5.

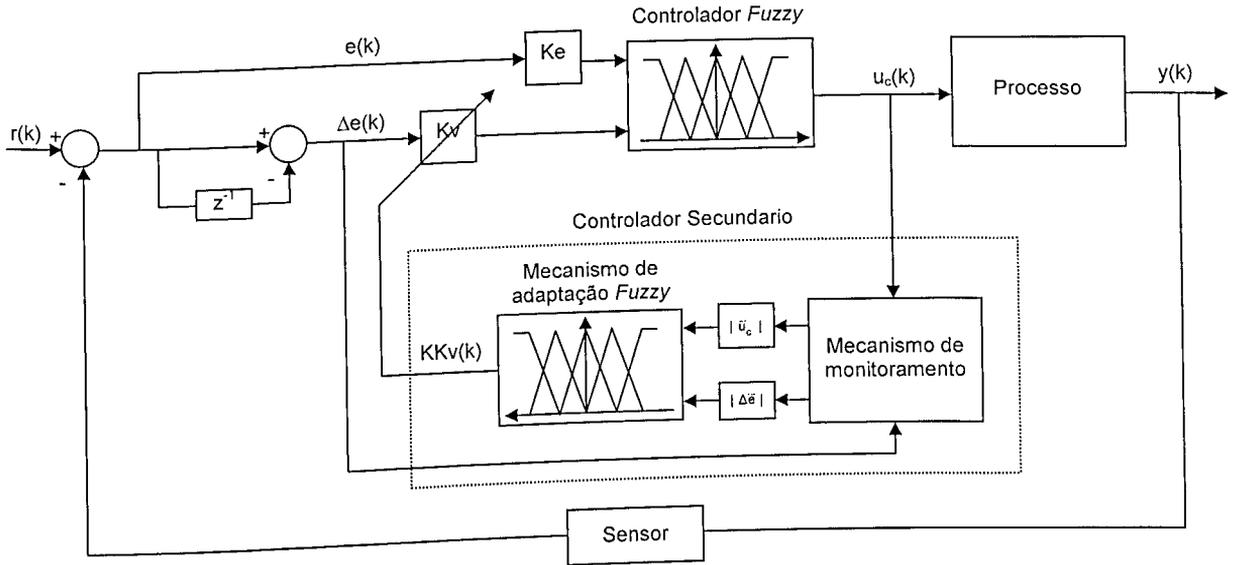


Figura 3.5 - O controlador *fuzzy* adaptativo

Assim como nos controladores *fuzzy* adaptativos clássicos, nesta abordagem tem-se também presente o mecanismo de monitoramento e o mecanismo de adaptação.

Mecanismo de monitoramento: O controlador adaptativo é monitorado a partir da média absoluta da derivada do erro $|\Delta\bar{e}|$ e da média da saída do controle $|\bar{u}_c|$ observando os n -últimos instantes de amostragem. Estes valores são calculados da seguinte forma:

$$|\Delta\bar{e}| = \frac{|\Delta e(k)| + \dots + |\Delta e(k-n)|}{n} \tag{3.4}$$

$$|\bar{u}_c| = \frac{|u_c(k)| + \dots + |u_c(k-n)|}{n} \tag{3.5}$$

Mecanismo de adaptação: O mecanismo de adaptação é de natureza *fuzzy* e possui duas entradas provenientes do mecanismo de monitoramento: a média absoluta da derivada do erro $|\Delta\bar{e}|$ e a média absoluta do controle $|\bar{u}_c|$.

De forma similar ao controlador *fuzzy* principal, são definidas para cada uma destas entradas funções de pertinência. Supondo três funções (Pequeno, Médio e Grande) do tipo

triangular igualmente espaçadas e definidas dentro de um universo de discurso apropriado¹. As saídas são funções de pertinência “crisp” que representam a ação desejada para o fator de escala, que seriam: (A)umentar K_v ($KK_v=1.2$); (M)anter K_v ($KK_v=1$) ou (D)iminuir K_v ($KK_v=0.8$), conforme mostrado na figura 3.6.

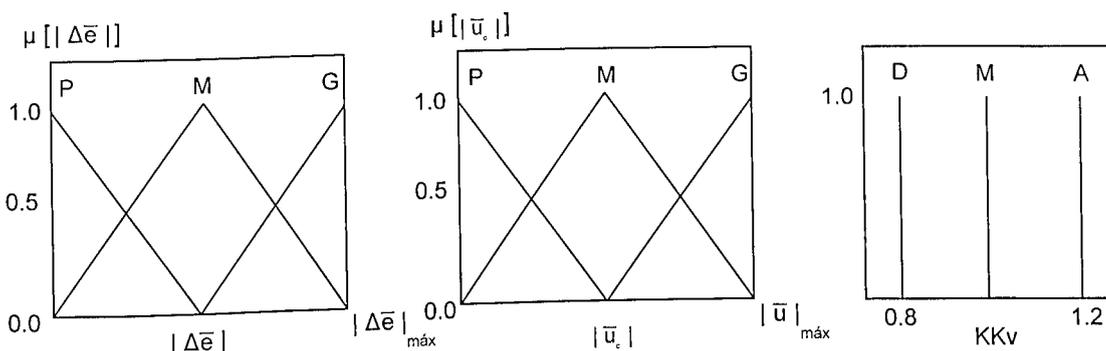


Figura 3.6 - Funções de pertinência e intervalos fuzzy das variáveis de entrada do mecanismo de adaptação

Pode-se, heurísticamente, inferir que:

- a) se $|\Delta\bar{e}|$ é grande e $|\bar{u}_c|$ é pequeno então é preciso aumentar o valor do controle. Isto pode ser obtido aumentando a sensibilidade à variação do erro do controlador *fuzzy* principal, ou seja aumentando o valor de K_v .
- b) se $|\Delta\bar{e}|$ é grande e $|\bar{u}_c|$ é grande então é preciso diminuir o valor do controle, pois o sistema pode estar divergindo. É preciso, portanto diminuir a sensibilidade à variação do erro do controlador *fuzzy* principal, ou seja é necessário diminuir o valor de K_v .
- c) se $|\Delta\bar{e}|$ é pequeno e $|\bar{u}_c|$ é pequeno então K_v deve ser mantido.

Estas e outras afirmações constituem a base de regras, construídas de forma intuitiva, retratadas na tabela 3.1.

Tabela 3.1 - Base de regras do mecanismo de adaptação fuzzy

		$ \Delta\bar{e} $		
		P	M	G
$ \bar{u}_c $	P	M	D	A
	M	M	M	D
	G	A	A	D

¹ No capítulo seguinte será discutida a robustez deste procedimento frente a incertezas nas definições dos limites destes universos do discurso

3.4.1 Inicialização e memória do fator de escala

O procedimento de adaptação proposto requer a inicialização do fator de escala K_v , conforme a expressão 3.2. Em se admitindo uma condição inicial de grandes perturbações, a evolução de K_v ao longo do processo deve permitir o crescimento gradativo dos esforços de controle. Assim, K_v deve começar pequeno (baixa sensibilidade à entrada) o que conduz à esforços iniciais de controle também pequenos. O controlador adaptativo do fator de escala, observando a condição inicial de elevados erros e de pequeno controle atua de forma a aumentar a sensibilidade do controlador *fuzzy* principal aumentando o fator de escala K_v por meio da saída KK_v .

Quando o sistema converge tem-se uma a uma situação de baixos sinais de entrada e baixos sinais de controle e um valor de K_v praticamente constante ($KK_v \approx 1$). Caso haja nesta situação uma perturbação que provoque o aumento do erro o algoritmo de adaptação, tal como proposto, aumentaria o valor de K_v (via KK_v) o que poderia levar à divergência do controlador. Para resolver esta questão propõe-se que, uma vez detectado um condição de equilíbrio do processo (caracterizado por KK_v próximo da unidade durante um certo tempo) deve-se diminuir o valor do fator de escala ao patamar da inicialização. Se isto for feito, o controle continua pequeno, mas o sistema continua “aberto” à novas perturbações. Os seguintes passos precisam ser tomados para garantir o funcionamento adequado do algoritmo.

- Inicializa K_v
(um valor pequeno garante baixos sinais de controle, $K_v=1/1000$ por exemplo)
- Verifica se o sistema atingiu o regime permanente.
(se durante as últimas N amostras o valor de KK_v permanece na faixa de $1 \pm \delta\%$, $N=20$ e $\delta=1$, por exemplo)
- Se o regime permanente foi atingido, reinicializa-se o valor de K_v .

Esta formulação heurística para o controlador adaptativo do fator de escala será objeto de avaliação e testes experimentais no capítulo V.

Capítulo IV

Os Microcontroladores

4.1 Introdução

No começo dos anos 70 a indústria eletrônica lançou no mercado mundial o microprocessador, componente que viria revolucionar o campo da eletrônica digital e analógica. Desde então numerosos sistemas de controle e de instrumentação industrial foram projetados em torno deste componente que se impôs não somente pelo preço como também pela versatilidade.

Os microprocessadores funcionam a partir de duas unidades básicas: uma de processamento e outra de controle de dados. São dotados de componentes externos auxiliares como memórias RAM (*Random Access Memory*), memórias EPROM (*Electric Program Read Only*), periféricos de entradas e saídas, temporizadores, etc. O avanço da eletrônica permitiu que os fabricantes deste tipo de microcircuitos, integrassem esses componentes auxiliares num só *chip*, estes novos microcircuitos foram denominados **microcontroladores**.

Neste capítulo, mostraremos de maneira geral, os conceitos básicos sobre microcontroladores, como exemplo e aplicação, será considerado o microcontrolador da Microchip® “PIC16F877”.

4.2 Características Gerais

Na atualidade diversos fabricantes vem desenvolvendo modelos de microcontroladores com características particulares baseados, porém, num mesmo conceito tecnológico, o que permite do projetista eleger o modelo que melhor se ajuste a suas necessidades, tanto utilitárias como econômicas.

Geralmente os microcontroladores apresentam as seguintes características básicas:

- uma CPU (*Central Processor Unit* ou Unidade de Processamento Central) que tem por finalidade interpretar as instruções do programa;
- memória de dados RAM (*Random Access Memory* ou Memória de Acesso Aleatório) utilizada para memorizar as variáveis utilizadas pelo programa;

- memória de programa PROM (*Programmable Read Only Memory* ou Memória Programável Somente para Leitura) que retém de maneira permanente as instruções do programa;
- um circuito de *clock* incorporado no próprio microchip;
- uma série de linhas de entradas/saídas para controlar dispositivos externos ou receber pulsos de sensores, chaves, etc.
- baixo consumo (*power down*);
- um conjunto de instruções que facilita a sua programação e confere maior rapidez na execução do programa;
- dispõe da possibilidade de proteção do programa;
- possui temporizadores programáveis e
- trabalham com uma o mais interrupções com diferentes níveis de prioridade.

Poderíamos seguir enumerando características dos microcontroladores, no entanto, há um elemento importante que fez tão popular a estes elementos: um conjunto de ferramentas de programação de alto nível e implementação, que foram disponibilizadas para o usuário.

4.3 Generalidades da arquitetura dos microcontroladores PIC

Os microcontroladores PIC são fabricados pela *Microchip® Technology* (Microchip, 2000) que tem um repertório amplo de microcontroladores com diferentes características e preços. Todos eles baseados na mesma arquitetura e com um repertório de instruções de programação bem semelhantes.

A maioria dos microprocessadores comuns e vários microcontroladores existentes no mercado tem sua estrutura interna de memória de dados e programa baseados na arquitetura de Von Neumann Figura 4.1. Esta arquitetura prevê um único *bus* (via) de comunicação entre memórias e CPU, nela todos os dados tratados pela CPU passam por um via única, assim, enquanto a CPU lê um dado ou instrução da memória, as vias internas não podem ser usadas para outra finalidade (Silva, 1997).



Figura 4.1 - Estrutura interna de *Von Neumann* do microprocessador

Os microcontroladores PIC's utilizam uma arquitetura diferente conhecida como Harvard, que prevê várias vias de comunicação entre CPU e periféricos, permitindo a realização de varias operações simultaneamente, o que implica num aumento considerável na velocidade de execução e permite ainda que a memória de dados e de programa tenham tamanhos diferentes.

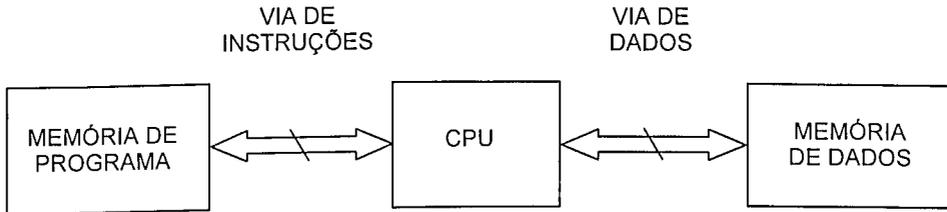


Figura 4.2- Estrutura interna *Harvard* de microcontroladores PIC's

O alto rendimento dos microcontroladores podetambém ser atribuído ao seu conjunto de instruções características do processador RISC (*Reduced Instruction Set Computer* ou computador de conjunto de instruções reduzido). Este processador aplica a técnica de segmentado “*pipe-line*” que permite ao processador realizar simultaneamente a tarefa de execução e procura em uma mesma instrução, assim, pode-se executar cada instrução num ciclo de máquina, que equívale a “quatro ciclos de um relógio”.

4.4 O Microcontrolador “PIC 16F877”

O PIC 16F877 é um microcontrolador de baixo custo e alto desempenho. Possui periféricos especiais como entradas analógicas/digitais (A/D), saída por modulação de largura de pulso (PWM) e principalmente memória relativamente grande (8K). Na figura 4.3 pode-se observar a configuração esquemática do PIC 16F877:

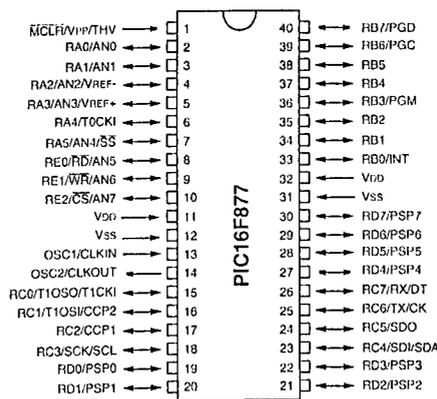


Figura 4.3 - Pinagem do PIC 16F877

Tabela 4.1 - Descrição dos pinos do "PIC 16F877"

Pino	Denominação	Descrição
1	MCLR/V _{PP}	Reset externo. Por este pino também se aplica tensão V _{PP} usada para a gravação do programa.
2, 3, 4	RA0/AN0-A2/AN2	E/S digital da Porta A, o entradas analógicas.
5	RA3/AN3/V _{REF}	E/S digital, analógica o entrada externa de V _{REF} .
6	RA4/T0CKI	E/S digital o entrada ao <i>clock</i> externo para o TMR0.
7	RA5/AN4/SS	E/S digital, analógica o seleção da porta serie síncrono.
8	RE0/RD/AN5	E/S digital da Porta E. Sinal de escritura da Porta paralela escravo. Entrada analógica.
9	RE1WR/AN6	E/S digital do Porto E. Sinal de escritura do Porto paralelo escravo. Entrada analógica.
10	RE2/CS/AN7	E/S digital. Sinal de ativação do Porto paralelo escravo. Entrada analógica.
11, 32	V _{DD}	Entrada do positivo da alimentação (2.5-6.0 V).
12, 31	V _{SS}	Terra.
13	OSC1/CLKIN	Entrada ao cristal de quartzo ou <i>clock</i> externo.
14	OSC2/CLKOUT	Saída do cristal. Em modo R-C por este pino sai ¼ F _{OSC1} .
15	RC0/T1OSO/TL1	E/S digital. Conexão do oscilador externo para o temporizador TMR1 o entrada do <i>clock</i> para o TMR1.
16	RC1/T1OSI/CCP2	E/S digital. Conexão do oscilador externo para TMR1 o saída do modulo 2 de captura/comparação.
17	RC2/CCP1	E/S digital da Porta C. Saída do modulo 1 de captura/comparação/PWM.
18	RC3/SCK/SCL	E/S digital. E/S de <i>clock</i> para a Porta serie síncrono (SSP) dos módulos SPI a I ² C.
19, 20, 21, 22, 27, 28, 29, 30.	RD0/PSP0-RD7/PSP7	E/S digitais da Porta D. Esta Porta pode trabalhar como Porta paralela escravo para interconexão com o bus de dados de 8 bits de outro microprocessador.
23	RC4/SDI/SDA	E/S digital. Entrada de dados serie no modo SPI. E/S de dados serie no modo I ² C.
24	RC5/SD0	E/S digital da Porta C. Saída de dados serie no modo SPI.
25	RC6/TX/CK	E/S digital. Transmissão serie assíncrona. Entrada do <i>clock</i> para comunicação serie síncrona.
26	RC7/RX/DT	E/S digital. Recepção serie assíncrona. Dados na comunicação serie síncrona.
33, 34, 35, 36, 37, 38, 39, 40.	RB0/INT-RB7	E/S digitais da Porta B. RB0/INT pode atuar como entrada de interrupção externa. RB4-RB7 podem provocar interrupção quando muda o estado.

Este microcontrolador apresenta as seguintes características principais:

Tabela 4.2 - Características do “PIC 16F877”

Frequência máxima de operação (MHz)	20
Memória EPROM (x14 bits)	8K
Memória de dados (bytes)	384
Módulos de Timers	TMR0, TMR1, TMR2
Capture/ Compare/ PWM	2
Numero canais A/D de 10bits	8
Numero de pinos (E/S)	33
Voltagem de operação (Volts)	2.5 – 6.0

4.4.1 Arquitetura interna

Como já mencionado anteriormente, uma das características importantes destes microcontroladores é que eles abandonam a tradicional arquitetura de *Von Neumann* e utilizam a arquitetura *Harvard*, onde a memória do programa é independente da memória de dados, assim o tamanho e a posição das memórias podem ser diferentes.

O espaço alocado à memória do programa do PIC 16F877 é de 8K palavras de 14 bits e espaço alocado à memória de dados é 384 bytes ou 384 registros de 8 bits.

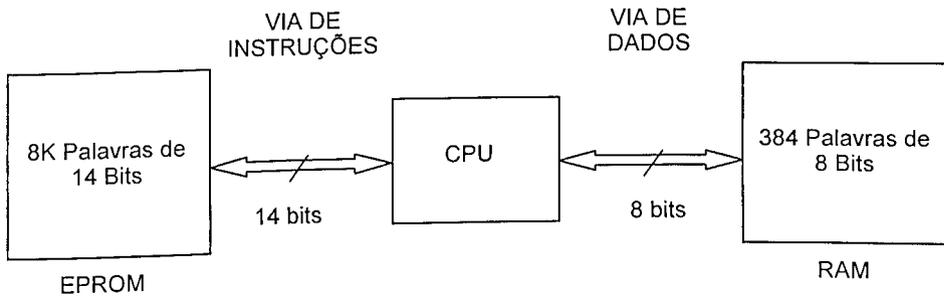


Figura 4.4 – Esquema de conexão da CPU com a memória de dados e programa

A memória do programa sempre é direcionada desde o contador do programa (PC), enquanto que a memória de dados pode ser direcionada diretamente desde o código da instrução ou indiretamente através de um registro denominado FSR (Registro de Seleção de Banco) (Garcia, 1998). A total independência entre a memória do programa e a memória de dados com o CPU permite realizar tarefas simultâneas, elevando a velocidade de trabalho.

Em geral, podemos dizer que os PIC dispõem no seu interior uma série de unidades (Fig. 4.5) interligadas entre si por meio de vias e linhas de controle, que permitem levar a cabo tarefas precisas para o desenvolvimento de programas complexos. Dentre estas unidades internas cabe destacar:

- a ALU (Unidade Aritmética lógica) que realiza os cálculos;
- unidade do oscilador, que marca a velocidade de trabalho do sistema;
- a memória de programa, constituída por uma ROM (pode ser PROM ou EPROM) que guarda as instruções e os dados invariáveis do programa;
- a memória de dados constituída por uma RAM que nos permite guardar os dados do programa a medida que se vão gerando durante a execução;
- os registros de uso geral que utiliza a ALU para suas operações e guardar dados momentaneamente;
- registro de seleção de bancos FSR;
- registro de estado STATUS;
- registro de trabalho W;
- os temporizadores, utilizados para realizar a contagem do programa ou bem para proporcionar um sinal a um determinado tempo e
- os registros que implementam as portas de entradas/saídas que permitem ao PIC se comunicar com o exterior.

programar a memória do programa, o microcontrolador deve ser posto num modo especial, colocando o pino /MCLR em alto nível de tensão.

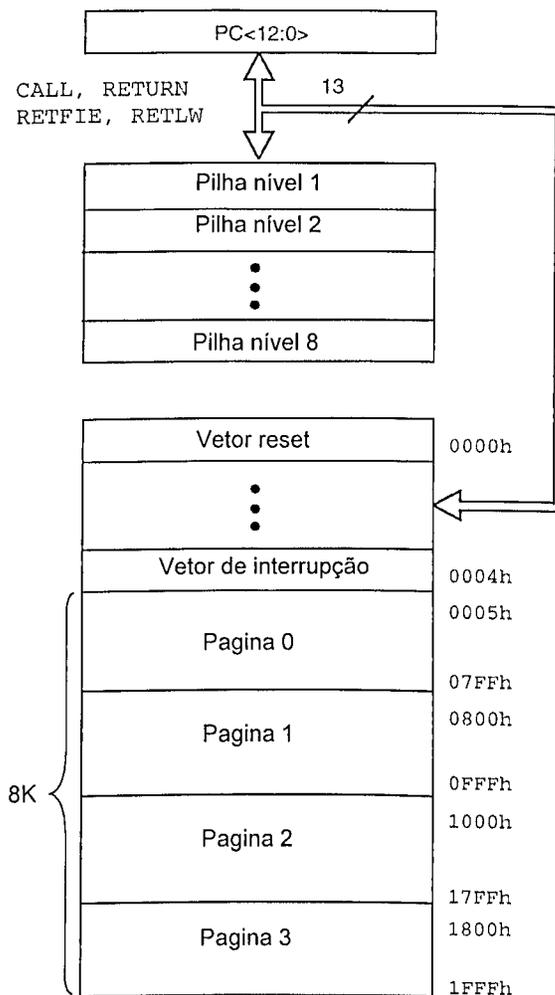


Figura 4.6 - Memória de programa e mapa da pilha do PIC 16F877

4.4.3 Memória de dados

A memória de dados é basicamente a memória RAM composta de uma série de registros ou bytes de 8 bits cada um. A esta se acessa de forma direta ou indireta (Garcia, 1998), mediante o registro de seleção de arquivo *FSR*. No registro de estado *STATUS* existem uns bits que permitem selecionar até 4 bancos na memória de dados, visto na figura 4.7.

Os registros da memória de dados se dividem basicamente em dois grupos, os registros de funções especiais e os registros de propósito geral. São citados alguns dos registros de funções especiais:

- registros TMR0, TMR1, TMR2;

- contador do programa PC;
- registro de estado STATUS;
- registro de seleção de banco FSR;
- registros de E/S, PORTA, PORTB, PORTC e PORTD e
- registros de endereçamento indireto.

Os registros de funções especiais são utilizados para controlar as funções de entrada/saída e programar as configurações necessárias para o funcionamento correto do sistema.

Os registros de propósito geral são utilizados para dirigir dados e informações mediante as instruções do programa a serem executados.

						Endereço do Arquivo	
Indirect addr. ⁽¹⁾	00h	Indirect addr. ⁽¹⁾	80h	Indirect addr. ⁽¹⁾	100h	Indirect addr. ⁽¹⁾	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
Registro de uso geral 96 Bytes		Registro de uso geral 80 Bytes		Registro de uso geral 80 Bytes		Registro de uso geral 80 Bytes	
	7Fh	acessos 70h-7Fh	Eh	acessos 70h-7Fh	16Fh	acessos 70h - 7Fh	1EFh
			F0h		170h		1F0h
			FFh		17Fh		1FFh
Banco 0		Banco 1		Banco 2		Banco 3	

Figura 4.7 - Banco de registros do PIC 16F877

4.4.4 Pinos de Entradas e Saídas E/S

Entende-se como pinos de entradas e saídas (E/S) as vias de comunicação ou os meios de transmissão de dados que têm à sua disposição os microcontroladores, permitindo a comunicação com os periféricos externos. Os microcontroladores PIC da família 16F877 dispõem de 33 pinos de E/S, deixando os outras para aplicações gerais, como receber a tensão de alimentação, conectar o cristal de quartzo (oscilador), etc.

Estes microcontroladores dispõem de quatro portas de comunicação denominados PORTA, PORTB, PORTC e PORTD. As portas E/S são tratados como qualquer outro registrador.

Para levar a cabo o controle do pinos, cada porta tem associado um registrador (TRISA, TRISB, TRISC e TRISD) que realiza o respectivo RESET de todos os bits do registrador. Estes registradores são controlados a partir do *software*. Por exemplo ao se colocar um nível lógico alto (1 lógico) em um determinado bit de um registro TRIS a portas ou o pino ficam configurados como entradas, se o nível lógico é baixo (0 lógico) os pinos são configurados como saídas.

4.4.5 Configuração do oscilador

O PIC 16F877 pode funcionar com quatro modos diferentes de oscilador, escolhidos em função dos elementos externos que serão utilizados. Estes modos de funcionamento são:

- RC: resistor/capacitor, baixo custo;
- LP: "Low Power Crystal" cristal de baixa potência;
- XT: cristal ou ressonador de baixa velocidade e
- HS: cristal ou ressonador de alta velocidade.

Os modos de osciladores mais utilizados são o modo XT e o modo RC. A configuração de montagem é mostrada a seguir:

Modo XT:

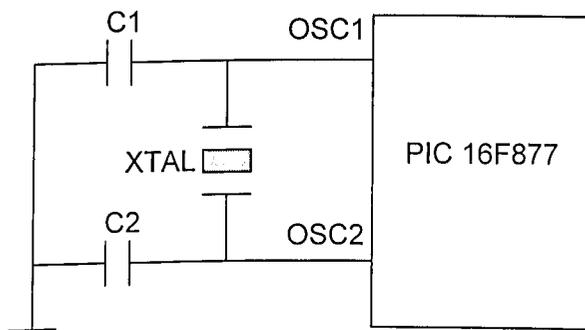


Figura 4.8 - Modo XT do oscilador

Para uma frequência de 20 MHz é recomendado capacitores C1 e C2 de 22pF (Microchip®, 2000).

Modo RC:

Quando a aplicação dada ao microcontrolador não requer precisão de tempos, pode-se utilizar o sistema oscilador RC, representado na figura 4.9:

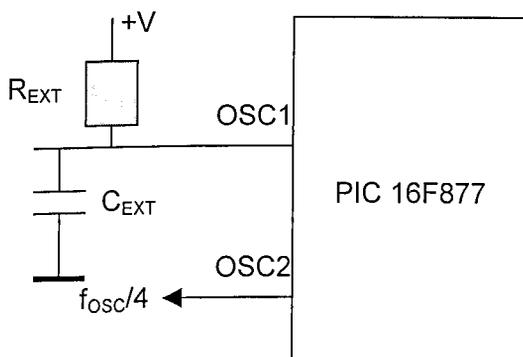


Figura 4.9 - Modo RC do oscilador

Neste tipo de montagens a frequência de oscilação depende dos valores da resistência e capacitor externo (R_{EXT} e C_{EXT}) assim como o valor da tensão de alimentação e da temperatura de funcionamento. É recomendado utilizar para R_{EXT} valores entre 3K e 100k, para o C_{EXT} é recomendado utilizar valores superiores a 20 pF para evitar problemas de ruído e estabilidade (Silva, 1997).

4.4.6 Os Ciclos do temporizador “Clock”

Quando se deseja executar um programa deve-se ter claro que as instruções do programa irão se executar segundo a ordem predeterminada pela velocidade de execução marcada pelo *clock* do sistema. O sinal do *clock* é gerado no pino OSC1 a uma determinada frequência “f” e é dividida internamente no PIC por quatro, gerando quatro novos sinais, figura 4.10, de frequência $f/4$. Estes novos sinais são denominados como Q1, Q2, Q3 e Q4.

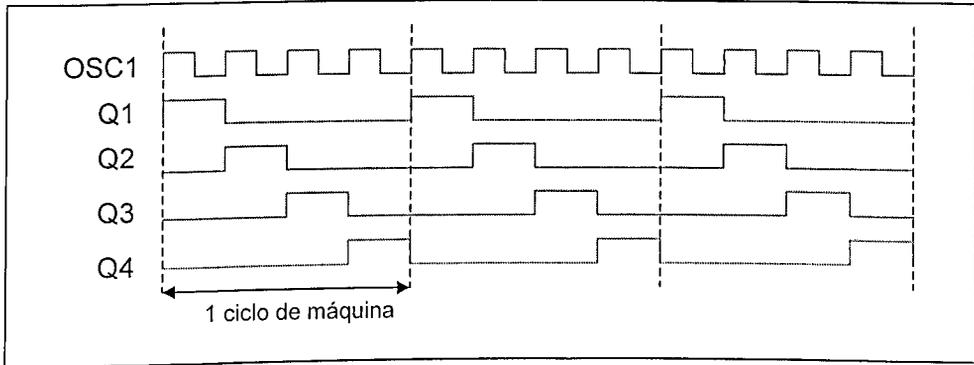


Figura 4.10 - Representação de um ciclo de máquina

O contador de programa (PC) é incrementado internamente a cada Q1 e uma instrução é extraída da memória do programa e colocada no “registro de instruções” durante Q4. Esta instrução é decodificada e executada durante os ciclos Q1 a Q4, realizando efetivamente a instrução em um ciclo. Se uma instrução muda o valor do contador do programa então se requer de dois ciclos para completar a instrução.

4.4.7 Características especiais

Como já mencionado os microcontroladores oferecem amplos recursos como temporizadores “TIMERS”, canais de entradas A/D, canais de saída PWM etc. Nesta etapa serão apresentados alguns destes recursos.

Temporizadores “TIMERS”:

Uma exigência na aplicação no controle é a regulação estrita dos tempos que duram as diversas ações que realiza o sistema. O dispositivo típico destinado a governar estes tempos recebe o nome de temporizador ou “*timer*”, que basicamente consiste num contador ascendente ou descendente que determina o tempo determinado entre o valor que é carregado e o momento em que se produz o seu desdobramento ou passo por zero.

Os microcontroladores desta classe possuem três tipos de TIMERS:

- **TIMER0:** é um módulo que constitui o temporizador principal do sistema. Dispõe de um registro de 8 bits programável que pode ser lido ou escrito, possibilita a divisão da frequência de trabalho, seleção do *clock* externo ou interno e pode trabalhar como contador;
- **TIMER1:** se trata de um temporizador/contador de 16 bits implementado nos registros TMR1H e TMR1L e

- **TIMER2:** este tipo de *timer* é incorporado em poucos modelos de microcontroladores PIC, se trata de um temporizador de 8 bits projetado para se utilizar conjuntamente com o circuito de Modulação de Largura de Pulso (PWM).

Conversor A/D:

Trata-se de um periférico que se encontra integrado no microcontrolador, caracterizado por um conversor analógico/digital de 10 bits com tensão de referência que pode ser interna (V_{DD}) ou externa. O funcionamento do A/D utiliza os registros ADCON0 e ADCON1. O primeiro seleciona o canal a ser convertido (são 8 os canais possíveis), ativa o conversor e contém o sinalizador que avisa o fim da conversão. O segundo estabelece as entradas que são digitais e analógicas, assim como o tipo de tensão de referência usado (interna ou externa).

O tempo que dura a conversão depende da frequência do funcionamento do PIC e do valor dos bits ADCS1 e ADCS0. Por exemplo, se o PIC trabalha a 20 MHz e os bits ADCS1:ADCS0=0, o tempo de conversão é de $T_{ad} = 100ns$ se ADCS1:ADCS0=1, o tempo de conversão é de $T_{ad} = 4\mu s$ (Microchip, 2000).

Modo PWM:

Estes microcontroladores dispõem de dois módulos chamados CCP1 e CCP2 que são controlados com os registros CCP1COM e CCP2COM respectivamente, nos quais se modula o controle do intervalo de tempo do microcontrolador.

O módulo PWM (*Pulse Width Modulation*) ou modulação por largura de pulso é um gerador de onda quadrada que é definida uma frequência fixa na qual se varia o “*Duty Cycle*”, ou seja, a parcela do tempo que a onda fica desligada em baixa (0) ou em alta (1). A figura 4.11 ilustra esquematicamente o PWM.

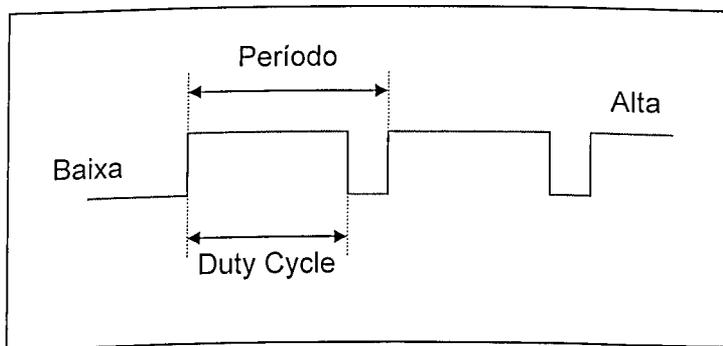


Figura 4.11 - Representação esquemática do PWM

4.5 Programação dos Microcontroladores

A existência de ferramentas de desenvolvimento, baseados em linguagem de alto nível, tem facilitado nos dias atuais, a tarefa de programação e o desenvolvimento de sistemas baseados em microcontroladores. Entre os quais podemos citar o compilador C da PCW[®], o *software* MPLAB[®], o gravador de microcontroladores *PICSTAR-PLUS*[®], entre outros.

O compilador “C” da PCW[®] permite realizar a programação de microcontroladores em linguagem de alto nível “C” e gera o código *assembler*. Este código pode ser usado num emulador da Microchip implementado num PC. O emulador permite o acompanhamento passo a passo do programa gerado, revelando o conteúdo das portas de E/S, dos registradores, timers etc. Uma vez depurado o programa, o mesmo é transferido para o microcontrolador por meio de um *hardware* apropriado, o *PICSTAR-PLUS*[®].

Uma vez apresentado os elementos básicos dos microcontroladores, o capítulo seguinte descreve alguns testes realizados usando tais componentes programados com a lógica *fuzzy*.

Capítulo V

Resultados Experimentais

5.1 Introdução

Este capítulo apresenta a metodologia de controle proposta e os ensaios experimentais efetuados no controle ativo de vibrações de uma mesa vibratória e de uma viga de aço flexível. Enquanto o controle da viga foi implementado inteiramente num microcontrolador, o controle da mesa vibratória fez uso de um PC. Isto porque o tempo de amostragem ($\leq 10\text{ms}$) requerido para o controle da mesa inviabilizou o uso do microcontrolador que se tinha disponível (PIC16F877) para a realização das experiências. Deve ser observado que procurou-se otimizar o *software* em termos de velocidade, operando-se inclusive com números inteiros, mas ainda assim, não foi possível usar o microcontrolador no controle da mesa.

5.2 Controle *fuzzy* adaptativo de uma mesa vibratória por meio de um PC

O sistema a ser controlado é composto por uma plataforma de alumínio montada sobre quatro lâminas metálicas flexíveis, configurando uma mesa vibratória, como mostrado na figura 5.1.

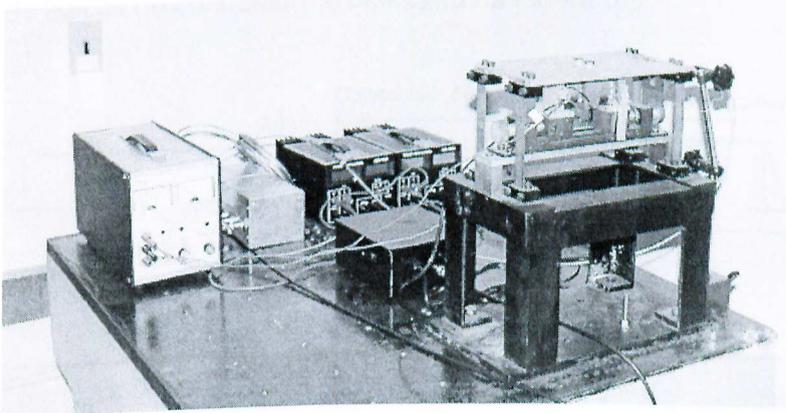


Figura 5.1 - Bancada experimental do sistema mecânico de um grau de liberdade

A mesa vibratória é instrumentada com um sensor de proximidade que gera as informações dos deslocamentos laterais da mesa. Estas informações são utilizadas por um controlador *fuzzy*, implementado em um microcomputador PC. Os sinais de controle provenientes do PC são enviados para um driver de corrente que alimenta um transdutor eletromagnético e transforma o sinal de entrada em forças eletromagnéticas de controle. Estes componentes são mostrados na Figura 5.2.

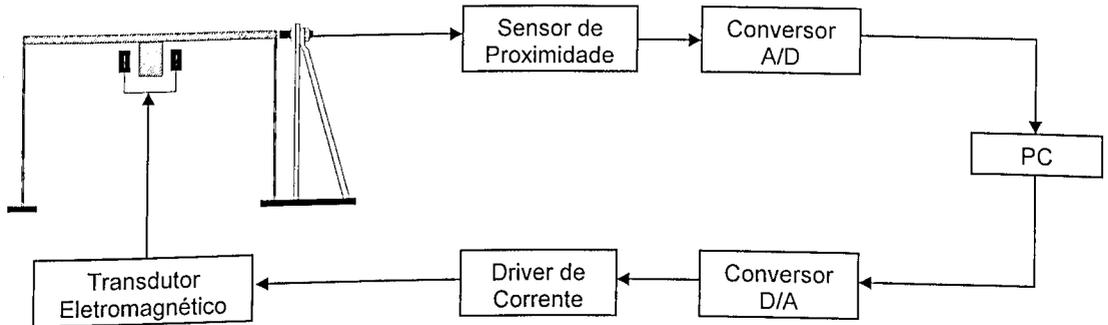


Figura 5.2 – Diagrama esquemático da mesa vibratória e instrumentação de controle

Para realizar o controle deste sistema foi feito inicialmente o projeto do controlador *fuzzy* principal (convencional) e em seguida foi desenvolvido o mecanismo *fuzzy* de adaptação dos ganhos.

5.2.1 Projeto do controlador fuzzy principal

O diagrama de blocos do controlador é mostrado na Figura 5.3.

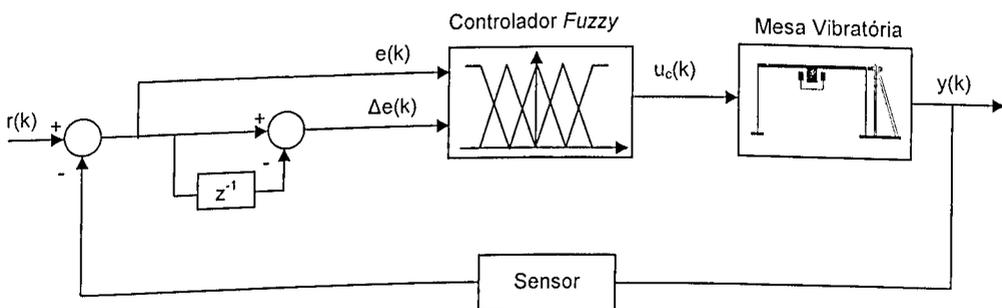


Figura 5.3 - Sistema de controle *fuzzy* do deslocamento lateral da mesa

A definição das variáveis de entrada do controlador *fuzzy* são avaliadas a partir do sinal do erro e da variação do erro, assim definidas:

$$e(k) = y(k) - r(k) \tag{5.1}$$

$$\Delta e(k) = \frac{e(k) - e(k-1)}{\Delta t} \tag{5.2}$$

onde k e Δt referem-se ao instante e ao intervalo de tempo (5ms) utilizados e $y(k)$ é a saída do sensor de proximidade.

Foram definidas para cada uma destas entradas três funções de pertinência [(N)egativo, (Z)ero e (P)ositivo] do tipo triangular e igualmente espaçadas e definidas dentro de um universo do discurso sintonizado de forma heurística, conforme mostrado na figura 5.4. A saída do controlador *fuzzy* é do tipo *Sugeno* de ordem zero, com valores “*crisp*” (N=-1.0, Z=0 e P=1.0) constantes. A base de regras é mostrada na Tabela 5.1 e o mecanismo de inferência utilizado foi o max-prod.

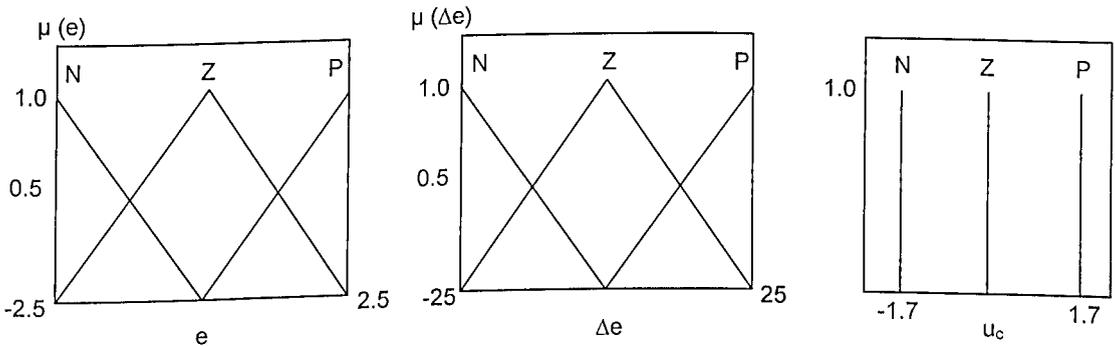


Figura 5.4 - Funções de pertinência *fuzzy* das variáveis de entrada e saída do controlador

Tabela 5.1 - Base de regras do controlador *fuzzy*

		Δe		
		N	Z	P
e	N	P ⁽¹⁾	P	Z
	Z	P	Z	N
	P	Z	N	N

Cada célula da tabela 5.1 representa uma regra construída da seguinte forma:

R1: Se o “e” é N e a “ Δe ” é N então a saída do controlador u_c é P=1.7

5.2.2 Projeto do controlador fuzzy adaptativo

Uma vez projetado o controlador principal, o segundo passo foi adicionar ao mesmo o mecanismo de adaptação proposto no capítulo 3. Para fazer isso, o universo do discurso do controlador principal foi normalizado num intervalo de -1 a 1, foi acoplado ao controlador *fuzzy* principal o mecanismo *fuzzy* de adaptação dos ganhos conforme figura 5.5, cujas regras são dados pela tabela 5.2.

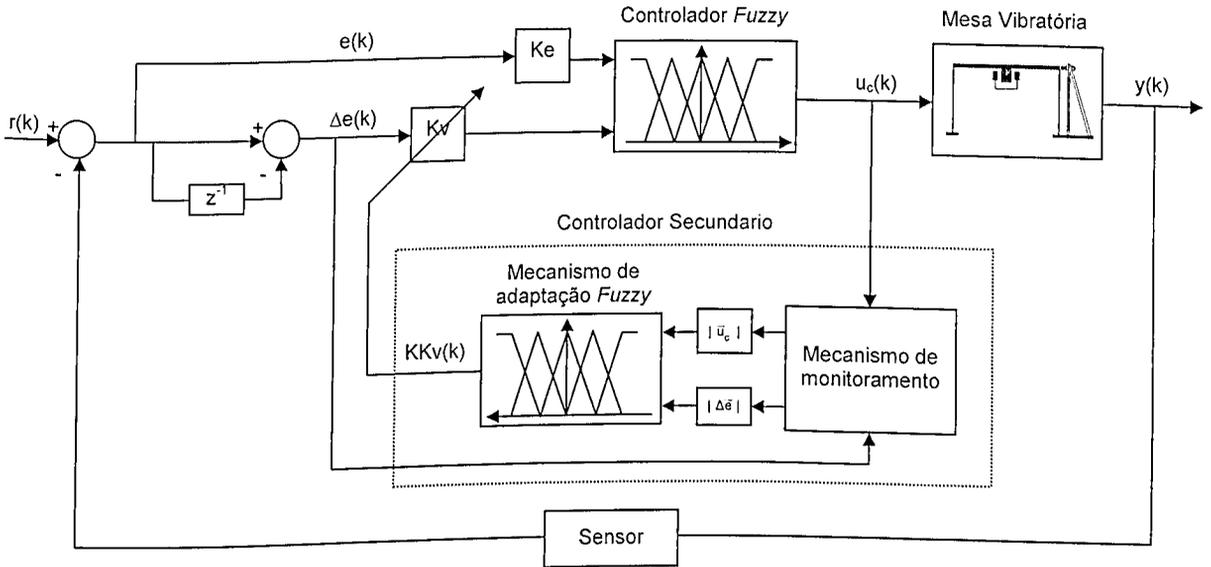


Figura 5.5 – Controlador *fuzzy* adaptativo

Tabela 5.2 - Base de regras do mecanismo de adaptação *fuzzy*

		$ \Delta \bar{e} $		
		P	M	G
$ \bar{u}_c $	P	M	D	A
	M	M	M	D
	G	A	A	D

Cabe destacar que o sistema é monitorado a partir da média absoluta da derivada do erro $|\Delta\bar{e}|$ e da média absoluta da saída do controle $|\bar{u}_c|$ observando os três últimos tempos de amostragem. Estes valores são calculados da seguinte forma:

$$|\Delta\bar{e}| = \frac{|\Delta e(k)| + |\Delta e(k-2)| + |\Delta e(k-3)|}{3} \tag{5.3}$$

$$|\bar{u}_c| = \frac{|u_c(k)| + |u_c(k-2)| + |u_c(k-3)|}{3} \tag{5.4}$$

De forma similar ao controlador *fuzzy*, foram definidas para cada uma das entradas do mecanismo *fuzzy* de adaptação, três funções de pertinência [(P)equeno, (M)édio e (G)rande] do tipo triangular igualmente espaçadas definidas dentro do universo de discurso das variáveis de entrada, e sintonizadas de forma heurística, conforme a figura 5.6. O mecanismo de inferência escolhido foi o max-prod e a saída do tipo *Sugeno* de ordem zero, assumindo os valores $D=0.8$; $M=1$ e $A=1.2$.

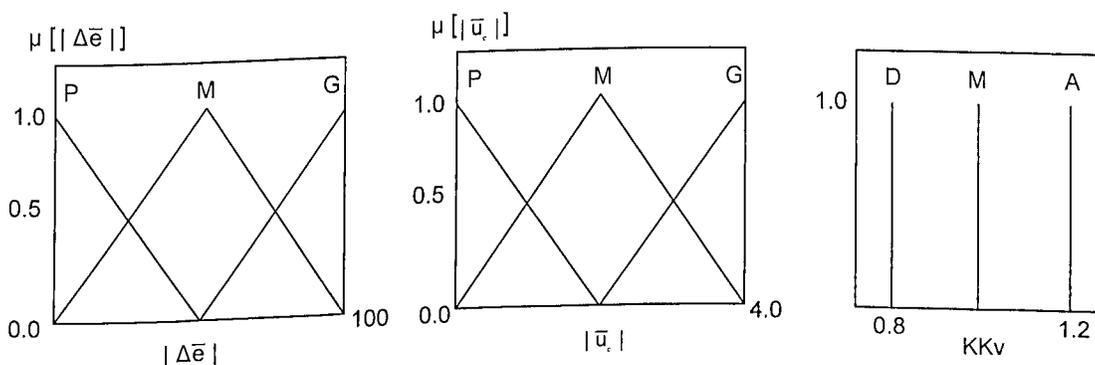


Figura 5.6 - Funções de pertinência e intervalos *fuzzy* das variáveis de entrada do mecanismo de adaptação

5.2.3 Resultados experimentais e discussões

O programa desenvolvido para controlar a mesa é apresentado no Anexo I. A taxa de amostragem utilizada foi de 200 Hz sendo que primeira freqüência natural do sistema é da ordem de 13 Hz (Ribeiro *et al.*, 1996).

Para a verificação da eficiência do controlador adaptativo são apresentados a seguir os testes experimentais realizados para o sistema sem controle e com controle, como também, com mecanismo de adaptação e sem mecanismo de adaptação.

O sistema em malha aberta sem o controlador foi testado experimentalmente com uma excitação inicial de natureza senoidal, na freqüência natural e amplitude de 0.75V. A excitação dura 0.5s, a partir deste instante o sistema oscila livremente como apresentado na figura 5.7.

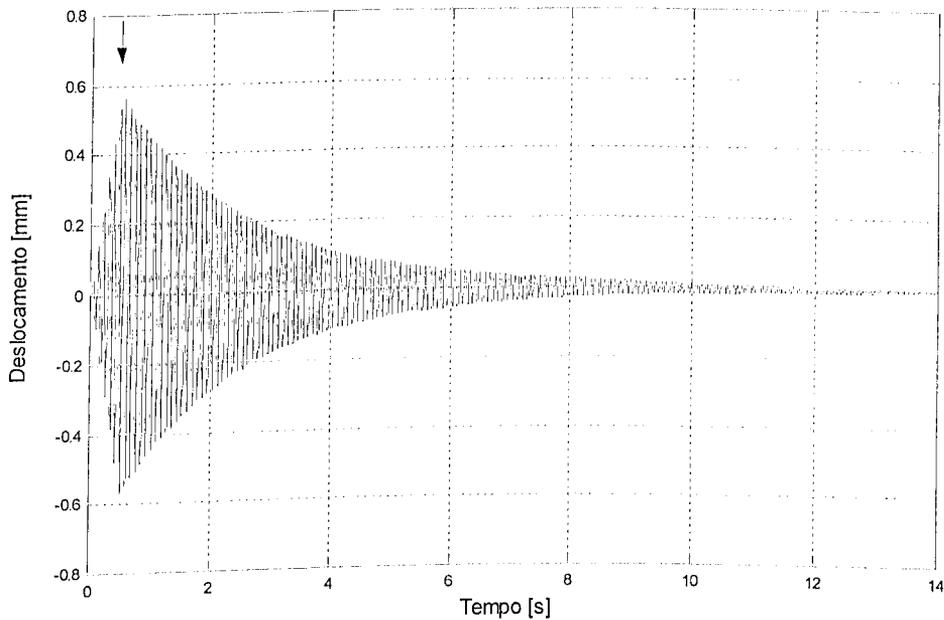


Figura 5.7 - Resposta do sistema livre a uma excitação senoidal inicial

Como se pode observar na figura 5.7, o sistema apresenta um comportamento pouco amortecido e um elevado tempo de acomodação, maior que 12s.

Para avaliar o desempenho e a robustez do controlador foram realizados um conjunto de experimentos variando o valor inicial de $K_v(k=0)$ e o intervalo do universos do discurso dos controladores *fuzzy*, conforme a tabela 5.3.

Em todos os testes o sistema foi excitado por uma força senoidal na sua freqüência natural, com duração de 0.5s. Após esta excitação é aplicado o sinal de controle.

Tabela 5.3 - Tabela de testes realizados para validação do sistema

N° Teste	Condição	Kv	Universo do Discurso do Controlador <i>Fuzzy Principal</i>		Universo do Discurso do Mecanismo <i>Fuzzy</i> de Adaptação	
			e	Δe	$ \bar{u}_c $	$ \Delta \bar{e} $
Teste 1	Sem Adapt.	-	[-2.5 a 2.5]	[-25 a 25]	-	-
Teste 2	Com Adapt.	1/50	[-1.0 a 1.0]	[-1.0 a 1.0]	[0.0 a 4.0]	[0.0 a 100]
Teste 3	Sem Adapt.	-	[-2.5 a 2.5]*150	[-25 a 25]*150	-	-
Teste 4	Com Adapt.	1/50	[-1.0 a 1.0]	[-1.0 a 1.0]	[0.0 a 4.0]*150	[0.0 a 100]*150
Teste 5	Sem Adapt.	-	[-2.5 a 2.5]/150	[-25 a 25]/150	-	-
Teste 6	Com Adapt.	1/50	[-1.0 a 1.0]	[-1.0 a 1.0]	[0.0 a 4.0]/150	[0.0 a 100]/150
Teste 7	Com Adapt.	1/1000	[-1.0 a 1.0]	[-1.0 a 1.0]	[0.0 a 4.0]	[0.0 a 100]

5.2.3.1 Sensibilidade do controlador fuzzy principal ao universo do discurso

Com o propósito de avaliar a influência da escolha do universo do discurso no desempenho do controlador *fuzzy* principal foram realizados os testes 1, 3 e 5. A figura 5.8 apresenta os resultados obtidos.

Como pode ser observado, tais parâmetros afetam de forma significativa a performance do controlador, o que significa dizer que uma escolha mal feita pelo projetista pode ocasionar desvios importantes no desempenho esperado para o sistema. Há sistemas físicos onde a definição deste universo não é tão simples e, mais que isso, o universo do discurso pode variar ao longo do tempo e também em função de perturbações externas de natureza imprevisas.

Deve-se observar que os universos do discurso escolhidos para o teste 1 foram obtidos a partir de simulações numéricas e avaliações experimentais (Cárdenas e Ribeiro, 2001).

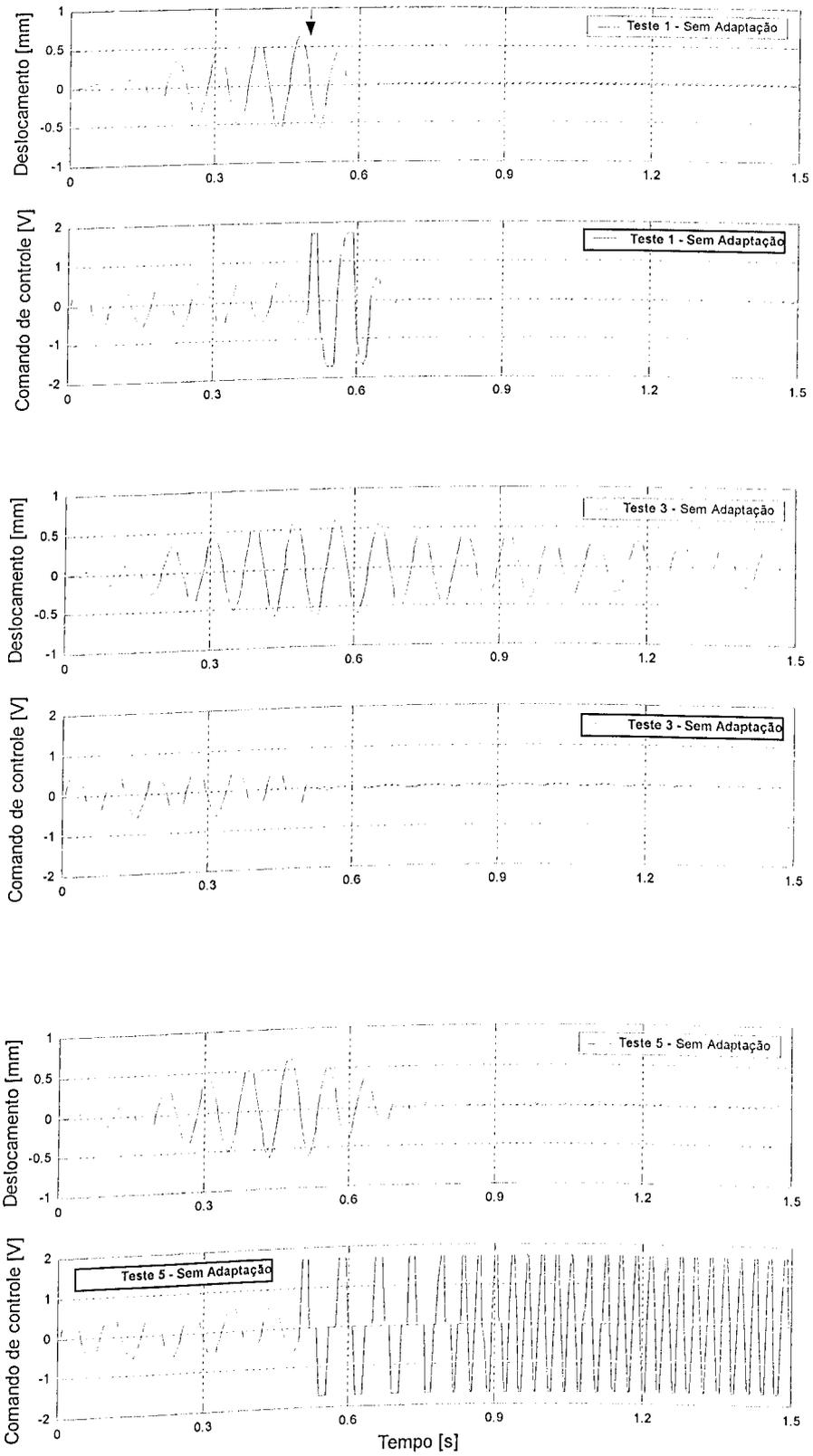


Figura 5.8 - Deslocamento lateral e o comando de controle para os testes 1, 3 e 5

Pode-se observar no teste 1 da figura 5.8 que o sistema é controlado em aproximadamente 0.2s, um resultado muito bom quando comparado com os 12s que o sistema livre demora para amortecer.

Ao se observar os resultados dos testes 3 e 5 pode-se comprovar a grande influência que o universo do discurso impõe no comportamento do sistema sob controle. O teste 3 revela que um universo de discurso muito grande acarreta esforços de controle muito pequenos que não conseguem controlar o sistema. Em contrapartida, como visto no teste 5, um universo do discurso muito pequeno provoca esforços de controle elevados, próximos à saturação, prejudicando o desempenho do controlador, embora de maneira não tão significativa como o caso do teste 3.

5.2.3.2 Sensibilidade do mecanismo de adaptação à variações do universo do discurso do mecanismo *fuzzy* de adaptação

Tendo em vista avaliar o mecanismo de adaptação proposto e a sensibilidade à variação do universo do discurso do mecanismo *fuzzy* de adaptação foram realizados os testes 2, 4 e 6.

A figura 5.9 apresenta a resposta do deslocamento da mesa e o esforço de controle para o controlador com mecanismo de adaptação. Deve-se lembrar que o universo do discurso do controlador *fuzzy* principal foi normalizado num intervalo de -1 a 1 .

Como se observa na figura 5.9, conseguiu-se um desempenho bastante satisfatório nos três testes efetuados, o que implica dizer que a estratégia de adaptação proposta demonstrou ser robusta, mesmo diante de variações significativas nos universos do discurso das variáveis de entrada do mecanismo *fuzzy* de adaptação. Nos três casos, os tempos de acomodação foram inferiores à 0.2s, enquanto para o sistema sem controle este tempo, como visto, é mais que 12s.

Novamente aqui se observa que universos de discursos muito pequenos (teste 6) conduzem à saturação no controle.

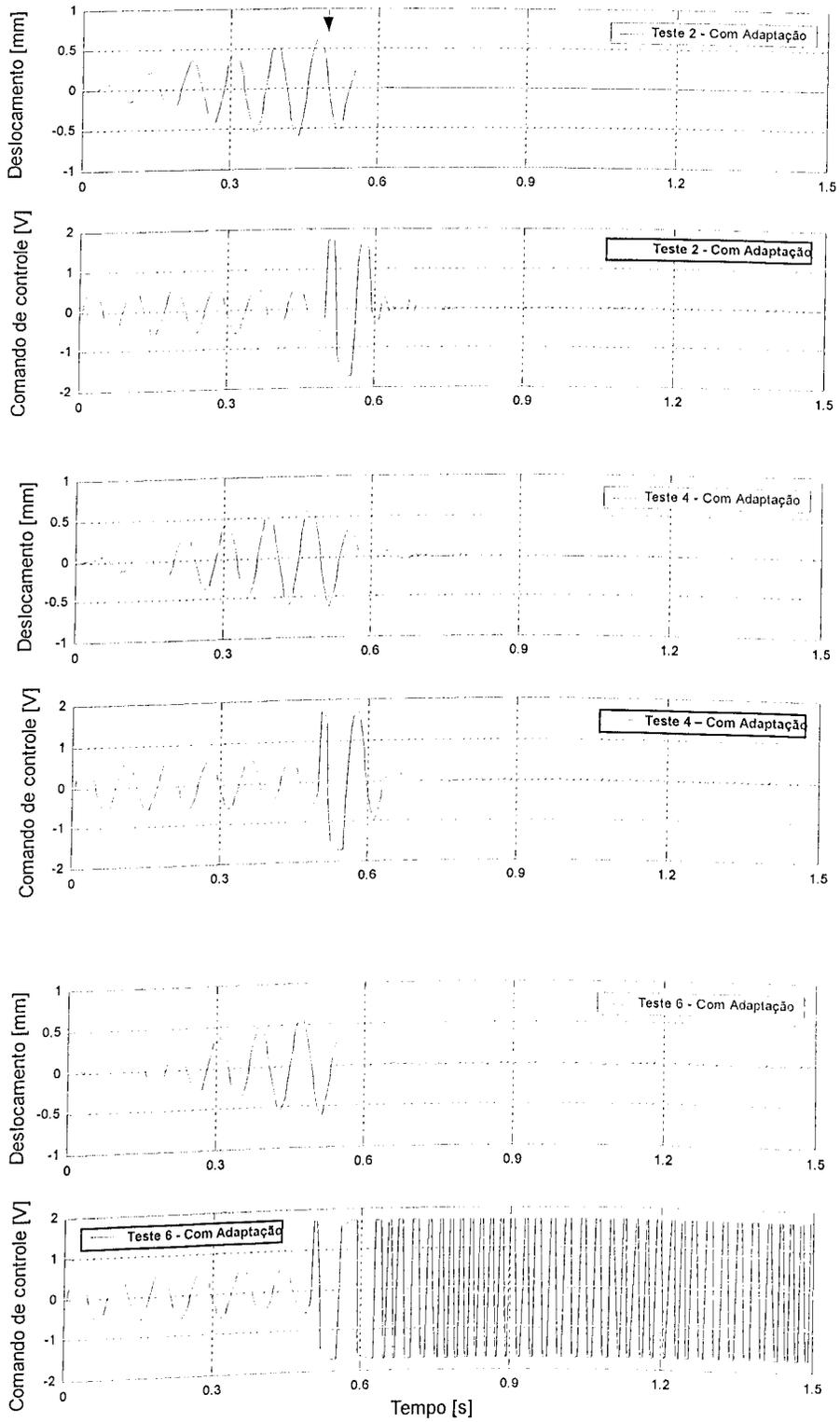


Figura 5.9 - Deslocamento lateral e comando de controle para o Teste 2, 4 e 6

5.2.3.3 Comparação entre o controlador *fuzzy* com e sem o mecanismo *fuzzy* de adaptação

Os teste 1 e 2 mostram o comportamento do sistema controlado com (teste 2) e sem (teste 1) o mecanismo de adaptação proposto. A figura 5.10 mostra os resultados obtidos.

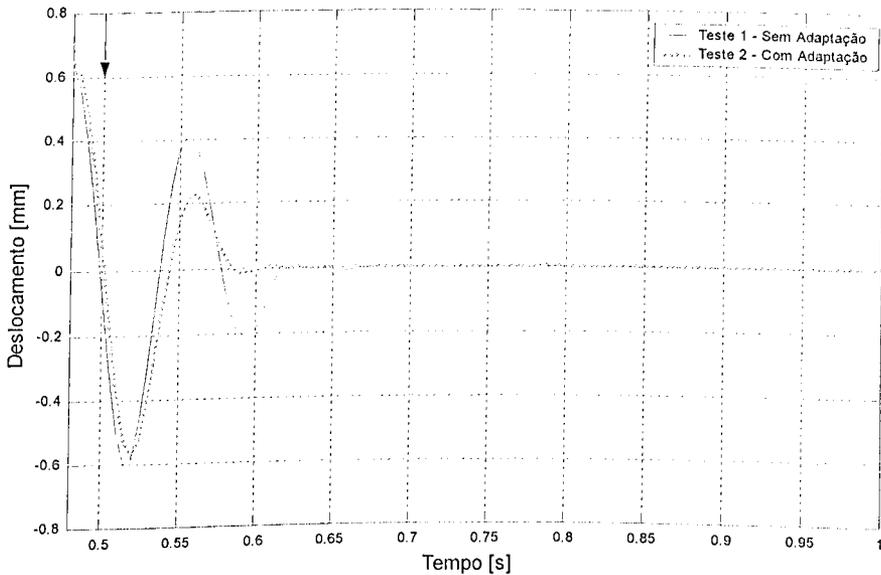


Figura 5.10 - Deslocamento lateral para o sistema em malha fechada com o controle sem adaptação e com adaptação.

Conforme pode ser observado na figura há uma pequena superioridade do controlador que incorpora o mecanismo de adaptação quando comparado com o controlador *fuzzy* convencional. O "overshoot" e o tempo de acomodação do controlador adaptativo são ligeiramente inferiores.

5.2.3.4 Influência do valor inicial do fator de escala K_v

Para investigar a influência do valor inicial escolhido para o fator de escala K_v foram realizados os testes 2 e 7. No teste 2 o valor adotado para K_v foi de 1/50 e no teste 7 um valor de 1/1000.

No teste 7 o valor inicial de K_v , observado na figura 5.12, é muito pequeno o que implica dizer que a derivada do erro tem pouca importância para o controlador principal nos instantes iniciais e que há inicialmente, o envolvimento de um número restrito de regras. Com o passar do tempo o valor de K_v vai aumentando tornando o controlador mais sensível ao Δ erro e, conseqüentemente um número maior de regras do controlador principal é utilizado, o que

melhora o desempenho do controlador. O valor inicial de K_v influencia portanto o número de regras envolvidas no controlador principal ao se inicializar o processo de controle e a "velocidade" com que outras regras são "atingidas". Assim o valor K_v assumido inicialmente afeta o tempo de acomodação do sistema controlado. Um baixo valor de K_v implica num tempo maior de acomodação e vice e versa.

De forma heurística, os testes sugerem que se deva, num primeiro ensaio, escolher um K_v inicial bastante pequeno e observar o valor para o qual ele converge. Deve-se, a partir da experiência realizada, tomar como referência para a inicialização do parâmetro K_v um valor duas ou três vezes inferior ao valor de convergência observado.

De qualquer forma, os testes 2 e 7 demonstram que o desempenho do sistema controlado é altamente satisfatório a despeito das elevadas variações do valor de K_v . Entre um teste e outro observa-se uma diferença de apenas 0.3s no tempo de acomodação, o que é muito bom quando comparado aos 12s do sistema sem controle.

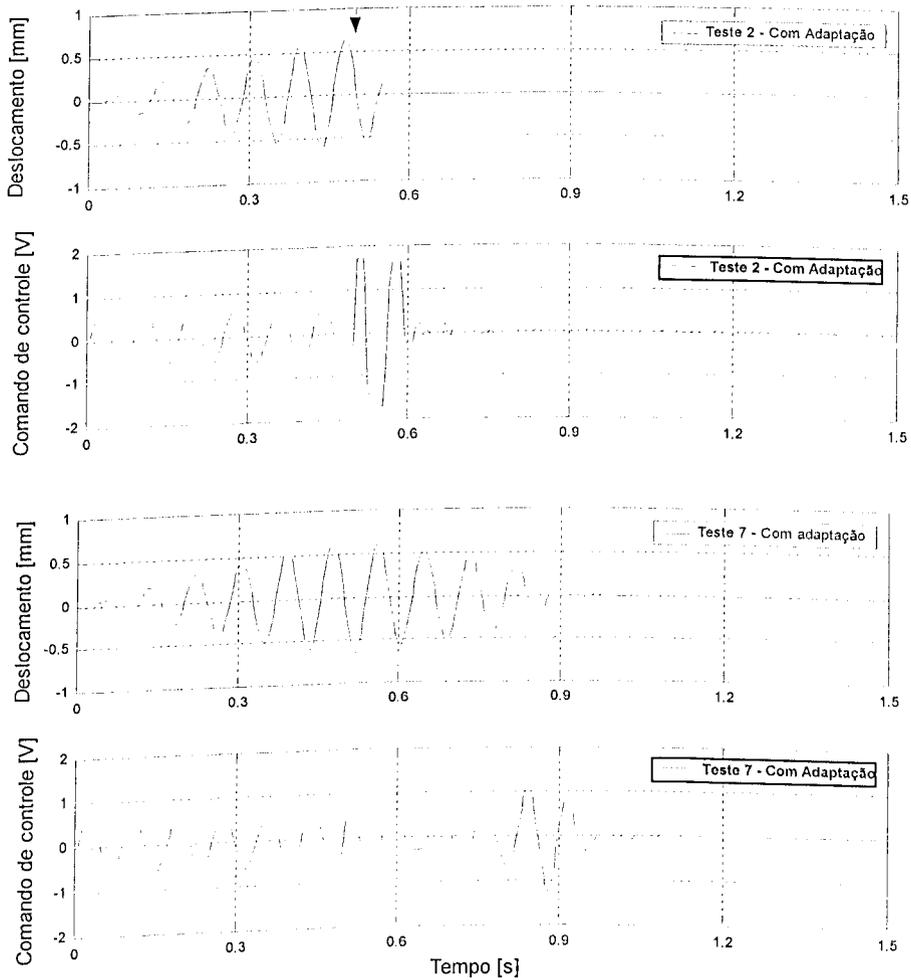


Figura 5.11 - Deslocamento lateral para o teste 2 e 7

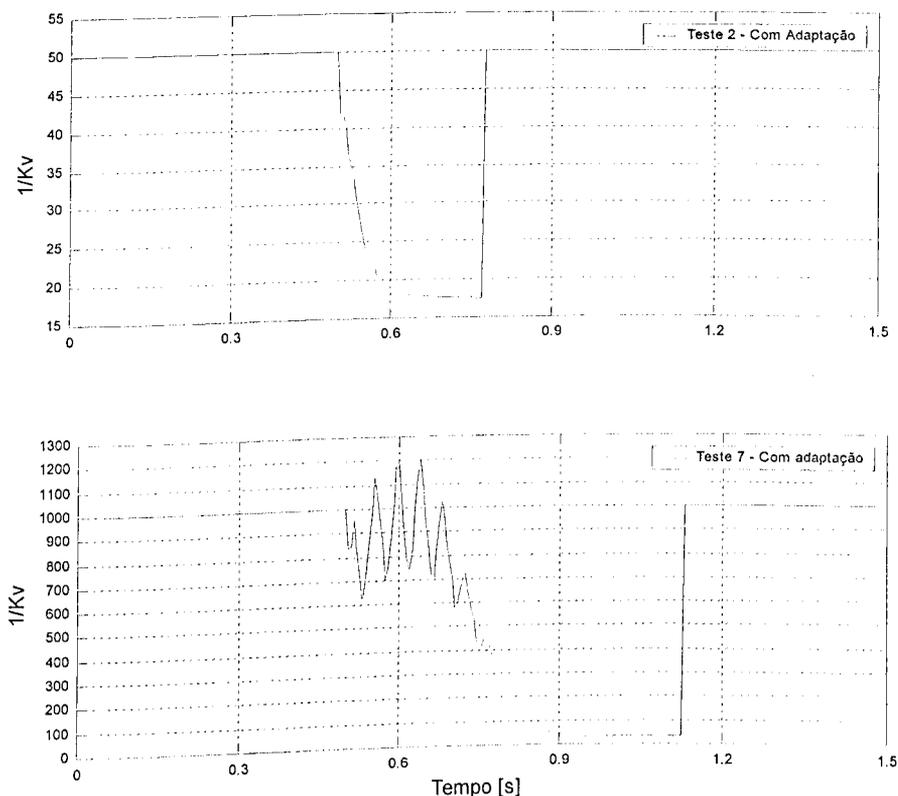


Figura 5.12 – Variação do fator de escala Kv para o teste 2 e 7

A figura 5.12 mostra a evolução no tempo do parâmetro $1/Kv$. Nos testes 2 e 7 o valor de Kv foi monitorado passo a passo e, quando a variação de $1/Kv$ permaneceu constante dentro de uma faixa de $\pm 1\%$ por mais de 20 instantes de amostragem, foi restabelecido o valor do Kv inicial, conforme estratégia já descrita no capítulo III. Como pode ser observado no teste 7 o valor de $1/Kv$ convergiu para um valor de $1/25$. Este valor serviu como referência para a inicialização do teste 2 como anteriormente mencionado.

5.3 Controle *fuzzy* adaptativo de uma viga flexível por meio de um microcontrolador

Esta seção apresenta alguns resultados experimentais relativos ao controle do primeiro modo de vibrar de uma estrutura flexível do tipo viga, utilizando a metodologia de controladores *fuzzy* adaptativos e implementadas em microcontroladores. A bancada experimental mostrada na figura 5.13, é composta por uma viga flexível de aço inoxidável, com duas placas de material piezoelétrico coladas nas duas faces opostas. As dimensões principais do modelo experimental da viga são mostradas na Figura 5.14 e em Abreu *et al.* (2001) uma descrição detalhada deste sistema é apresentada.

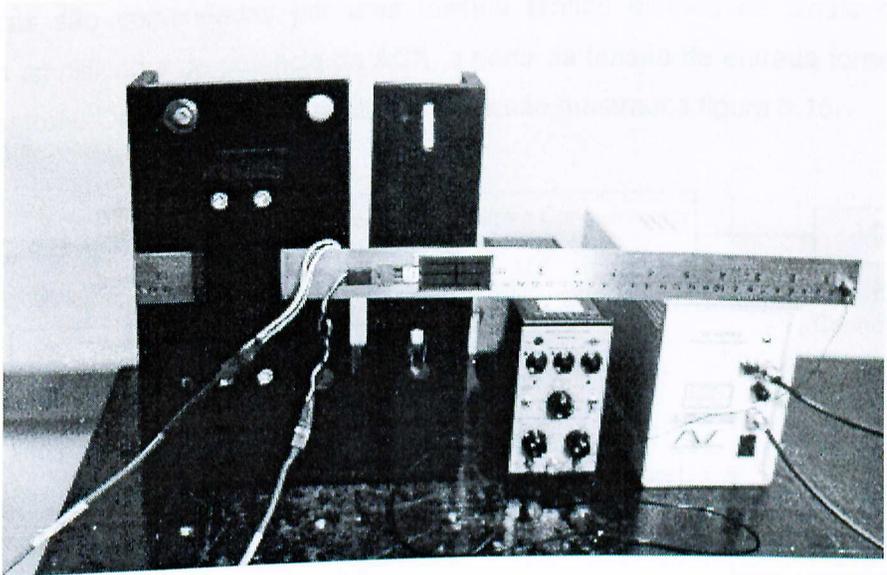


Figura 5.13 - Bancada experimental do sistema mecânico composto pela viga flexível

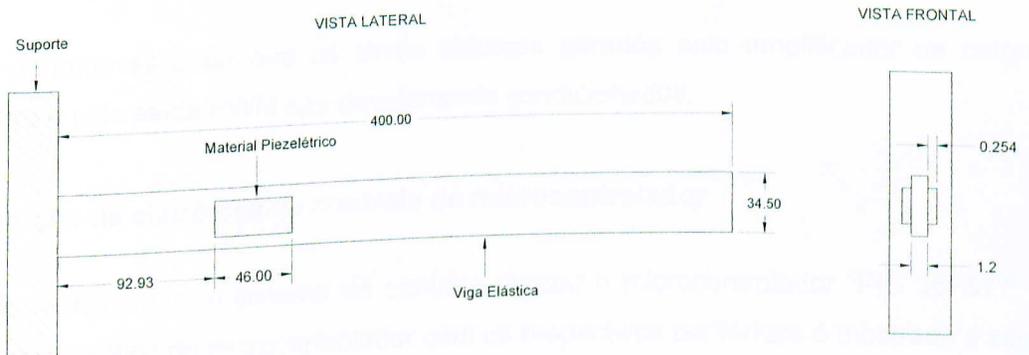


Figura 5.14 - Características dimensionais do modelo experimental [mm]

A viga flexível é instrumentada por um acelerômetro posicionado a 31cm da base engastada o que coincide de forma deliberada com o nó do segundo modo de vibrar da viga, uma vez que a proposta do controle é atenuar as vibrações decorrentes do primeiro modo apenas. As informações do acelerômetro são utilizadas pelo controlador *fuzzy* adaptativo implementado num microcontrolador “PIC16F877”. O microcontrolador adquire o sinal do sensor, calcula o controle e gera um sinal de saída do tipo PWM que alimenta os transdutores piezoelétricos. Para monitorar o experimento foi utilizado um sistema de aquisição de dados que adquire os sinais de controle gerados pelo microcontrolador e também um sensor de proximidade indutivo posicionado a 14cm da base da viga.

O elemento piezoelétrico, fornecido pelo fabricante da empresa ACX (*Active Control aXperts*), é constituído por duas placas de material piezoelétrico colados nas faces opostas da viga. Tais placas são comandadas por uma mesma tensão elétrica de sinais contrários, geradas por um amplificador de potência da ACX, a partir da tensão de entrada fornecida pela eletrônica de controle. Os componentes desta bancada são mostrados figura 5.15.

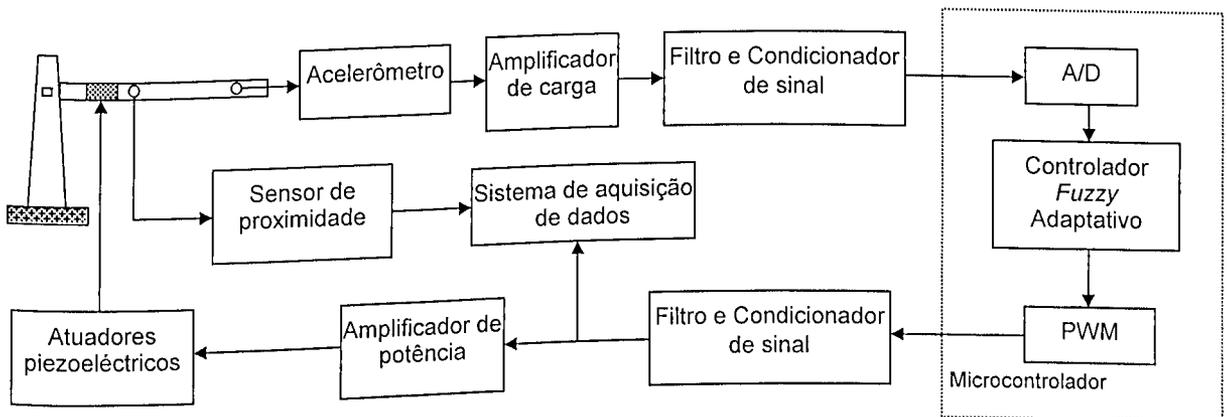


Figura 5.15 - Instrumentação da bancada do sistema de vários graus de liberdade

É importante salientar que os sinais elétricos gerados pelo amplificador de carga do acelerômetro e pela saída PWM são devidamente condicionados.

5.3.1 Descrição da eletrônica de controle do microcontrolador

Como mencionado, o sistema de controle utilizou o microcontrolador “PIC 16F877” e o diagrama esquemático do microcontrolador com os respectivos periféricos é mostrado a seguir:

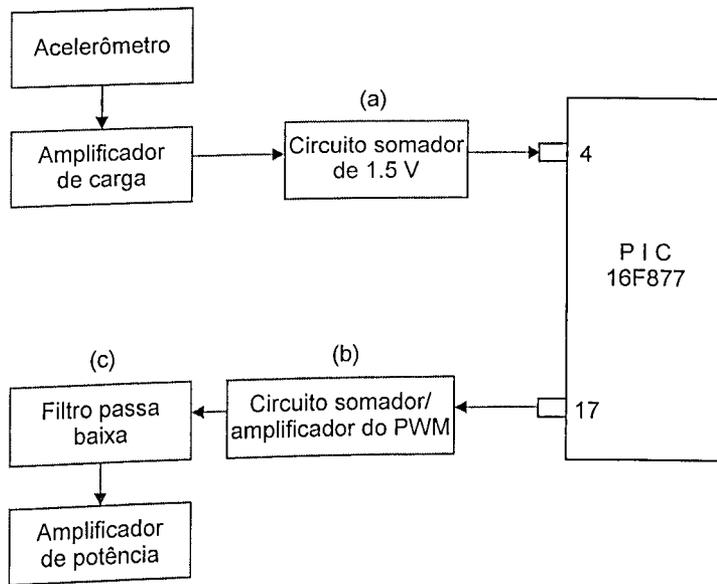


Figura 5.16 - Esquema da eletrônica do microcontrolador PIC16F877

(a) Circuito somador: O sinal de deslocamento da viga proveniente do condicionador de carga do acelerômetro é um sinal de tensão de natureza bipolar. Como o conversor A/D do microcontrolador é do tipo unipolar foi necessário implementar um circuito somador para eliminar o nível de tensão negativa. O circuito mostrado na figura 5.17, soma uma tensão de 1.5V à saída do amplificador de carga e o valor verdadeiro de tensão é internamente (por software) subtraído do sinal de entrada.

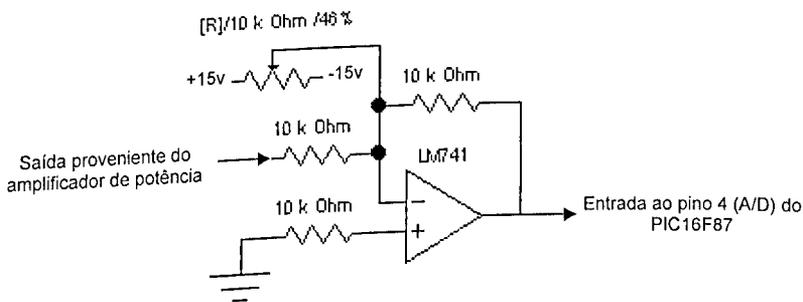


Figura 5.17 - Circuito somador de 1.5V somado ao pino 4 (A/D) do PIC 16F877

(b) Circuito somador/amplificador: Para gerar a saída analógica do microcontrolador utilizou-se a saída PWM a uma frequência de 4 KHZ. Como já foi explicado anteriormente a saída PWM varia de 0V, correspondente a 0% do ciclo em baixa, a +5V, correspondendo a 100% do ciclo em alta. No entanto, o sinal de controle deve suportar valores positivos e negativos, no intervalo de -5V a 5V, isto foi conseguido por meio de um circuito amplificador somador

mostrado na figura 5.18. Este circuito amplifica o sinal duas vezes, e subtrai 5V, obtendo o sinal desejado no intervalo de +5V a -5V.

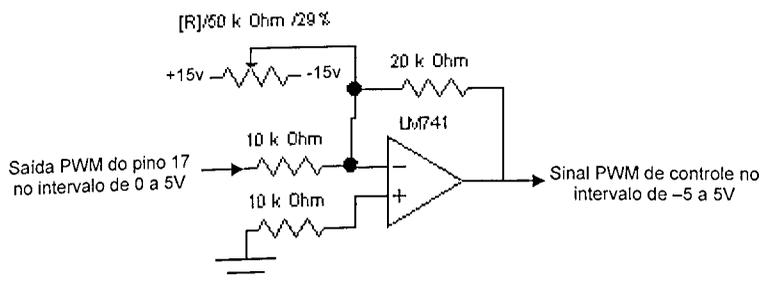


Figura 5.18 - Circuito amplificador/somador do sinal PWM de controle

(c) Filtro passa-baixa: Um filtro passa-baixa de 100 Hz é aplicado ao sinal PWM proveniente do circuito anterior eliminando o comportamento de alta frequência e fazendo dele um conversor A/D físico.

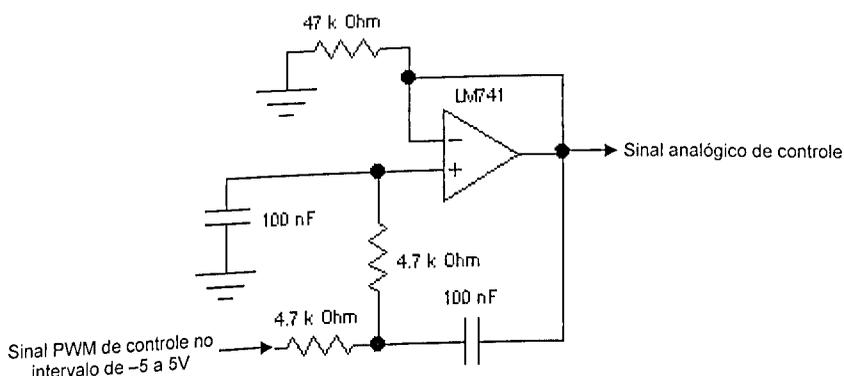


Figura 5.19 - Filtro passa baixa de 100 Hz

5.3.2 Análise em frequência da resposta da viga

O ensaio experimental para a determinação das frequências naturais da viga foram feitas por meio da análise da resposta em frequência. Para a determinação das frequências naturais do sistema utilizou-se um acelerômetro piezoelétrico instalado próximo à extremidade livre. O sistema foi excitado por um martelo de impacto tipo 8200, da Brüel & Kjaer, instrumentado com uma célula de carga piezoelétrica que gera uma entrada impulsiva. O sinal do acelerômetro do tipo B&K 4375 foi condicionado por um amplificador de sinais B&K.

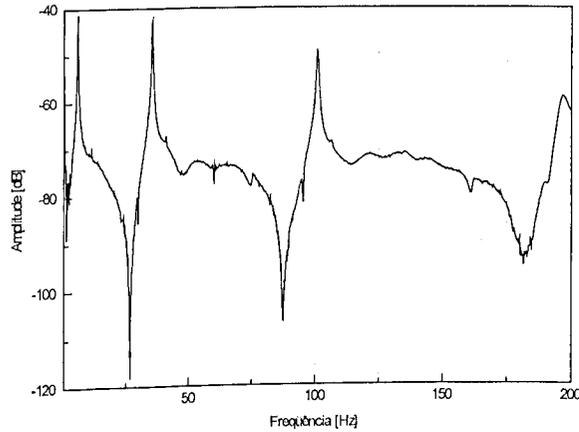


Figura 5.20 - Função resposta em frequência da viga livre

A partir da figura 5.20 foi possível obter o valor das 4 primeiras frequências naturais da estrutura com o elemento piezoelétrico incorporado, como mostra a tabela 5.4.

Tabela 5.4 Frequências naturais experimentais da viga

Modo	Frequência Hz
1	5.75
2	35.5
3	100.75
4	196.80

A partir da análise da função resposta em frequência, foi projetado de forma intuitiva um controlador *fuzzy* e o mecanismo de adaptação para o controle das vibrações da estrutura no seu primeiro modo de vibração (isto é, 5.75Hz) utilizando como atuador o elemento piezoelétrico.

5.3.3 Projeto do controlador *fuzzy* principal

De maneira análoga ao controle da mesa vibratória, foi construído um controlador *fuzzy* com duas entradas, o erro e a variação do erro, e uma saída u_c de controle. Foram definidas para cada uma destas entradas, três funções de pertinência [(N)egativo, (Z)ero e (P)ositivo] do tipo triangular igualmente espaçadas e sintonizadas de forma heurística (Abreu *et al.*, 2001). A saída deste controlador também é do tipo *Sugeno* de ordem zero, admitindo valores “*crisp*” (N=-5, Z=0, e P=5) constantes. Foram utilizadas apenas 5 regras tendo em vista economizar espaço de memória no microcontrolador. O mecanismo de inferência utilizado foi o max-prod.

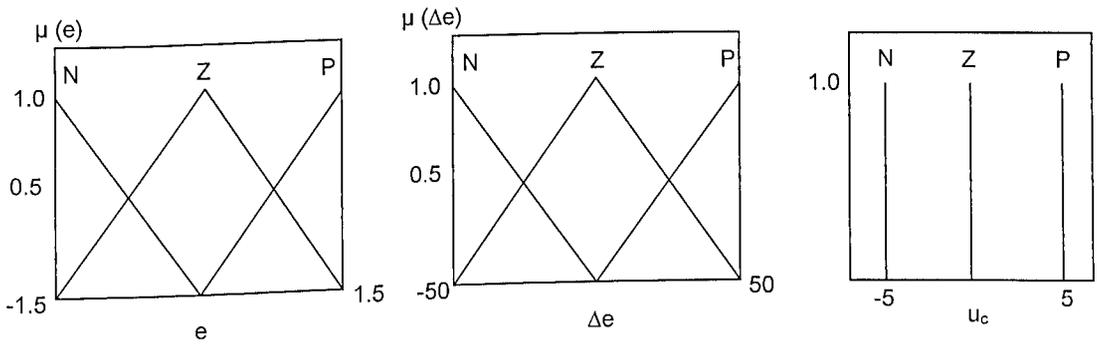


Figura 5.21 - Funções de pertinência e intervalos *fuzzy* das variáveis de entrada do controlador

Tabela 5.5 Base de regras do controlador *fuzzy*.

		Δe		
		N	Z	P
e	N	P	-	-
	Z	P	Z	N
	P	-	-	N

5.3.4 Projeto do controlador *fuzzy* adaptativo

Foi utilizado a mesma metodologia de adaptação do experimento anterior, com as funções de pertinência mostradas na figura 5.22.

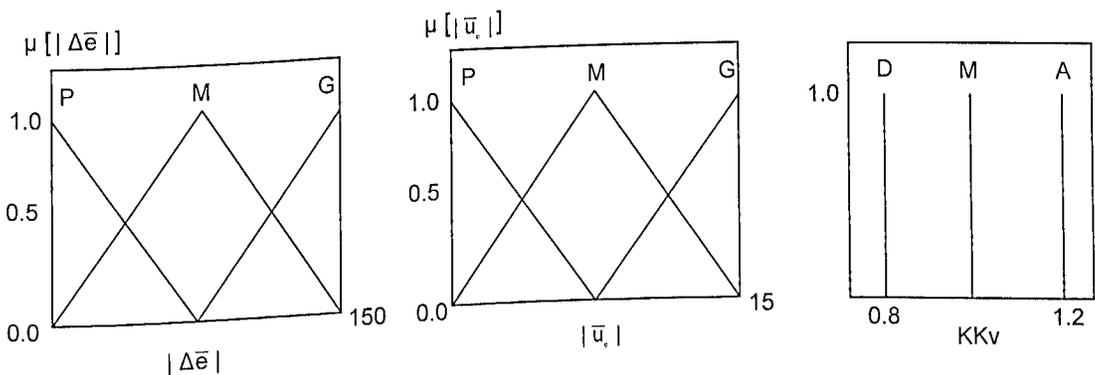


Figura 5.22 - Funções de pertinência e intervalos do mecanismo de adaptação

5.3.5 Implementação do controlador *fuzzy* adaptativo no microcontrolador

O programa desenvolvido para realizar o controle da viga engastada é apresentado no Anexo II e foi desenvolvido na linguagem C, utilizando o software PCW[®]. O programa foi

analisado passo a passo com o software MPLAB® e transferido para o microcontrolador PIC16F877 a partir do gravador de microcontroladores PICSTAR PLUS®.

O fluxograma de controle obedeceu aos seguintes passos:

- a) leitura do sinal do sensor;
- b) cálculo do sinal de controle e
- c) geração do sinal PWM.

A taxa de amostragem utilizada para controlar a primeira freqüência natural (5.75 Hz) foi de 66.6 Hz. Como mencionado anteriormente este microcontrolador possui uma capacidade de memória ROM de 8K que foi ocupado em 73%. Quanto à memória de dados RAM, a capacidade deste microcontrolador é de 368 bytes e foi utilizado 58% desta capacidade.

5.3.6 Resultados experimentais e discussões

O sistema em malha aberta sem o controlador foi testado experimentalmente com uma excitação inicial de natureza senoidal, na primeira freqüência natural (5.75 Hz) e amplitude de 0.75V. A excitação dura apenas 0.5s, a partir deste instante o sistema oscila livremente como apresentado na figura 5.23.

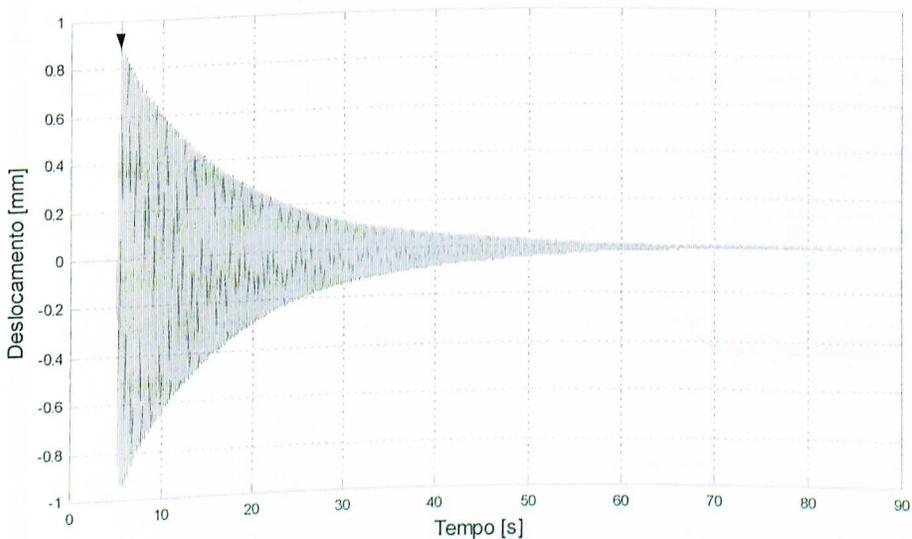


Figura 5.23 - Resposta experimental do sistema sem controle a uma excitação senoidal

A figura mostra o comportamento pouco amortecido do sistema destacando-se o elevado tempo de acomodação, maior que 80s.

Para a verificação da eficiência e robustez do controlador *fuzzy* em conjunto com o mecanismo de adaptação foram realizados uma série de experimentos que são mostrados na tabela 5.6.

Tabela 5.6 Tabela de testes realizados para avaliação do controlador proposto

N° Teste	Condição	Kv	Universo do Discurso do Controlador <i>Fuzzy Principal</i>		Universo do Discurso do Mecanismo <i>Fuzzy</i> de Adaptação	
			e	Δe	$ \bar{u}_c $	$ \Delta \bar{e} $
Teste 1	Sem Adapt.	-	[-1.5 a 1.5]	[-50 a 50]	-	-
Teste 2	Com Adapt.	1/1000	[-1.0 a 1.0]	[-1.0 a 1.0]	[0 a 15]	[0 a 150]
Teste 3	Com Adapt.	1/1000	[-1.0 a 1.0]	[-1.0 a 1.0]	[0 a 15]*4	[0 a 150]*4
Teste 4	Com Adapt.	1/1000	[-1.0 a 1.0]	[-1.0 a 1.0]	[0 a 15]/4	[0 a 150]/4
Teste 5	Com Adapt.	1/30	[-1.0 a 1.0]	[-1.0 a 1.0]	[0 a 15]	[0 a 150]

5.3.6.1 Desempenho do controlador fuzzy com e sem o mecanismo fuzzy de adaptação

Os testes 1 e 2 mostram o comportamento do sistema com e sem mecanismo de adaptação.

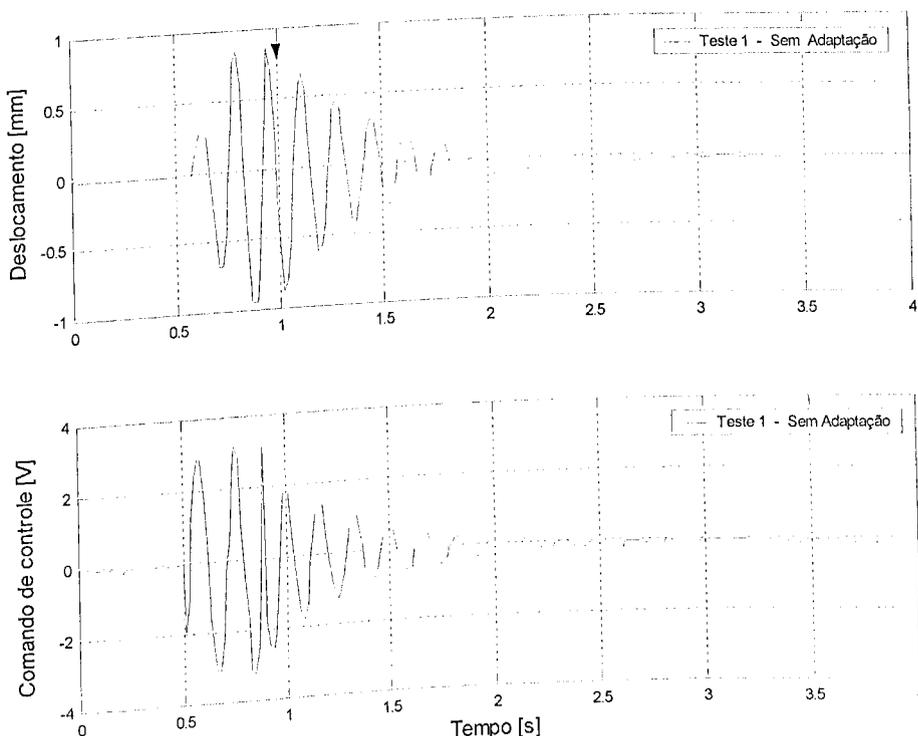


Figura 5.24 - Deslocamento lateral e comando de controle sem mecanismo *fuzzy* de adaptação

Nesta figura é observado que o sistema é controlado aproximadamente em 2s implicando uma atenuação bastante satisfatória, quando comparado com os 80s do sistema livre.

Após este teste foi acoplado ao controlador principal o mecanismo *fuzzy* de adaptação (teste 2) e, como observado na figura 5.25, o desempenho do controlador também é bastante satisfatório com um tempo de acomodação de aproximadamente 1s.

Neste teste (figura 5.25) observou-se um sinal de controle inicialmente pequeno (devido ao baixo valor de K_v inicial) e vai aumentando e que quando as excitações são amortecidas, diminui sensivelmente. Pode-se observar na figura 5.26 a adaptação do valor K_v ao longo do tempo.

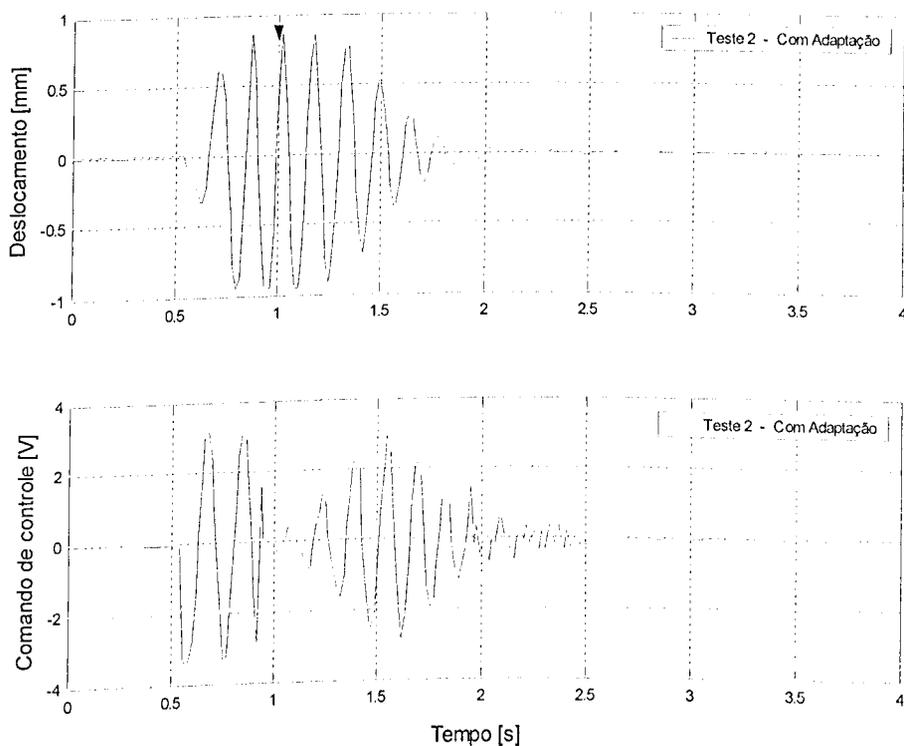


Figura 5.25 - Deslocamento lateral e comando de controle com mecanismo *fuzzy* de adaptação

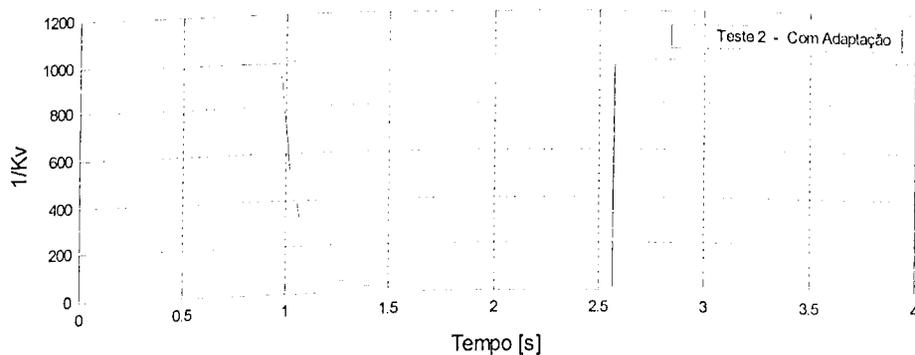


Figura 5.26 - Variação do fator de escala K_v

Na figura 5.27 é observado em detalhe que o tempo de acomodação do controlador com mecanismo de adaptação é inferior ao do controlador *fuzzy* convencional, apresentando assim uma eficiência ligeiramente superior.

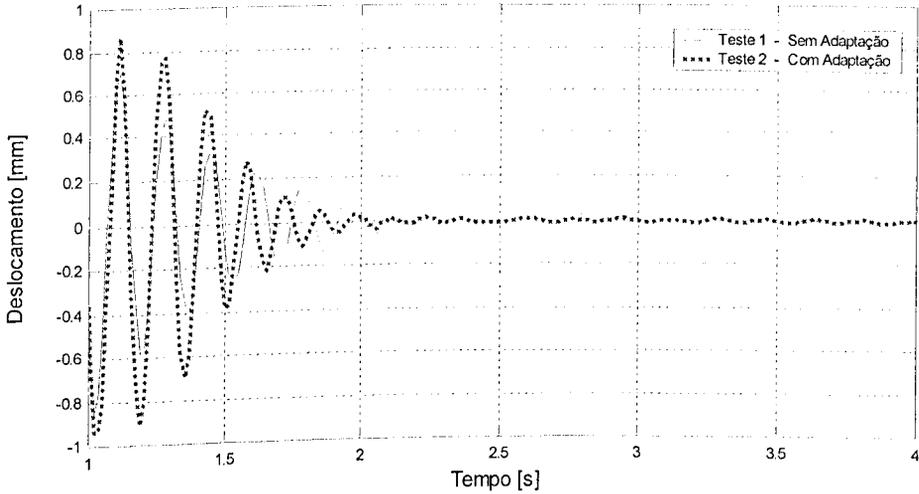


Figura 5.27 - Deslocamento lateral do sistema com e sem mecanismo de adaptação

5.3.6.2 Sensibilidade do mecanismo de adaptação à variações do universo do discurso do mecanismo *fuzzy* de adaptação

Para avaliar a robustez do controlador e a influência do universo do discurso do mecanismo de adaptação, foram realizados os teste 3 e 4:

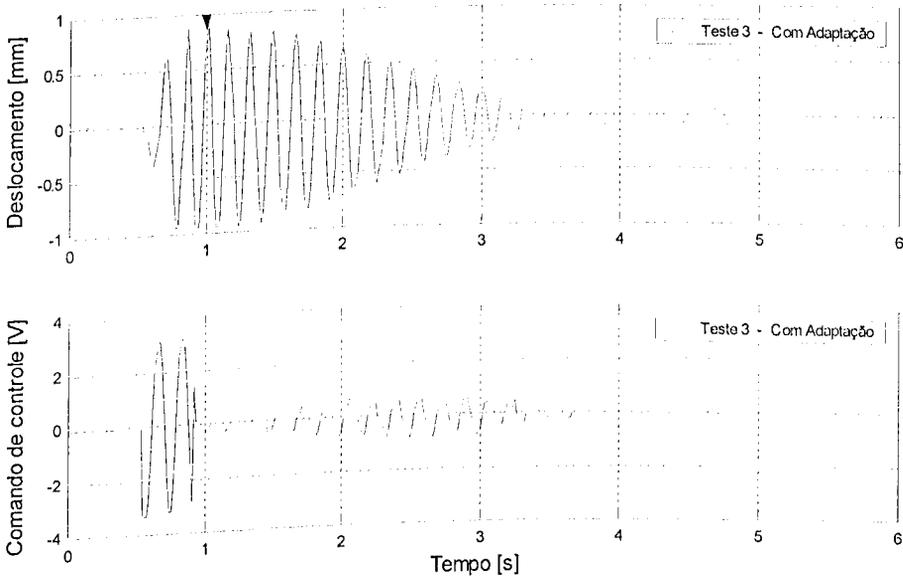


Figura 5.28 - Deslocamento lateral e comando de controle do sistema *fuzzy*

Pode-se observar nesta figura que há uma atenuação da amplitude do deslocamento do sistema, porém o controle não apresenta um desempenho tão satisfatório quanto a do teste 2, pois o universo do discurso do mecanismo *fuzzy* de adaptação é muito grande, e o controlador perde um pouco sua sensibilidade, não chegando todavia à divergir.

Na figura seguinte é mostrado o controle do sistema com um universo do discurso do mecanismo *fuzzy* de adaptação muito pequeno, conforme tabela 5.6. Como se pode observar o controlador conseguiu introduzir um apreciável amortecimento no sistema, a despeito da maior “sensibilidade” do controlador, observado pela variação do sinal de controle.

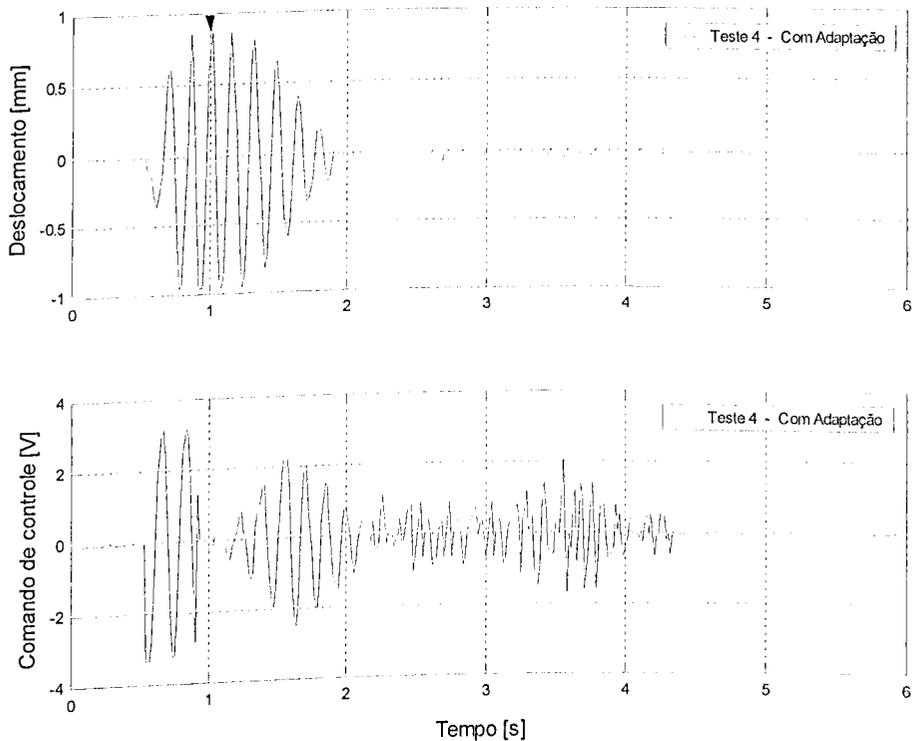


Figura 5.29 - Deslocamento lateral e comando de controle do sistema *fuzzy* adaptativo

5.3.6.3 Sensibilidade do controlador *fuzzy* adaptativo à variações do K_v inicial

Para avaliar a sensibilidade do sistema à escolha do valor inicial de K_v foram realizados os testes 2 e 5. Na metodologia proposta, conforme descrito no capítulo III, o valor inicial de K_v deve ser, num primeiro ensaio, inicialmente pequeno ($1/K_v=1000$, por exemplo). Deve-se

observar experimentalmente o valor para o qual K_v converge e então escolher, para o segundo ensaio, um valor duas ou três vezes superior ao valor de convergência observado.

O teste 2 (figura 5.26) revela que $1/K_v$ converge para um número em torno de 10. Por isso, conforme a metodologia proposta, no teste 5 escolheu-se para a inicialização $1/K_v=30$. Os resultados obtidos são mostrados nas figuras abaixo.

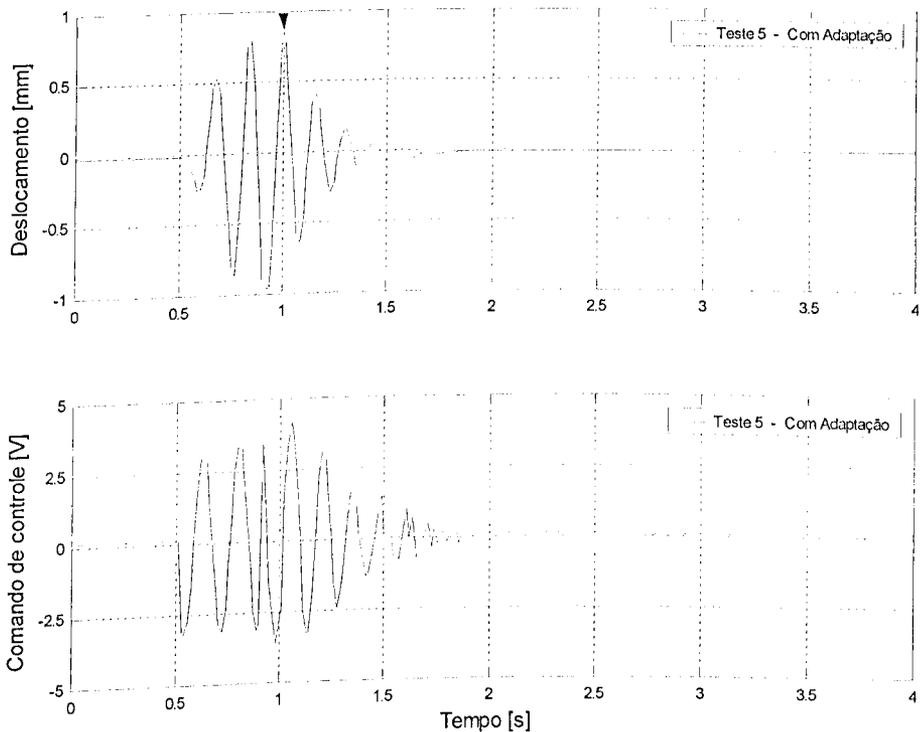


Figura 5.30 - Deslocamento lateral e comando de controle do sistema *fuzzy* adaptativo

Na Figura 5.31 é observado o desempenho deste controlador quando comparado ao controlador convencional.

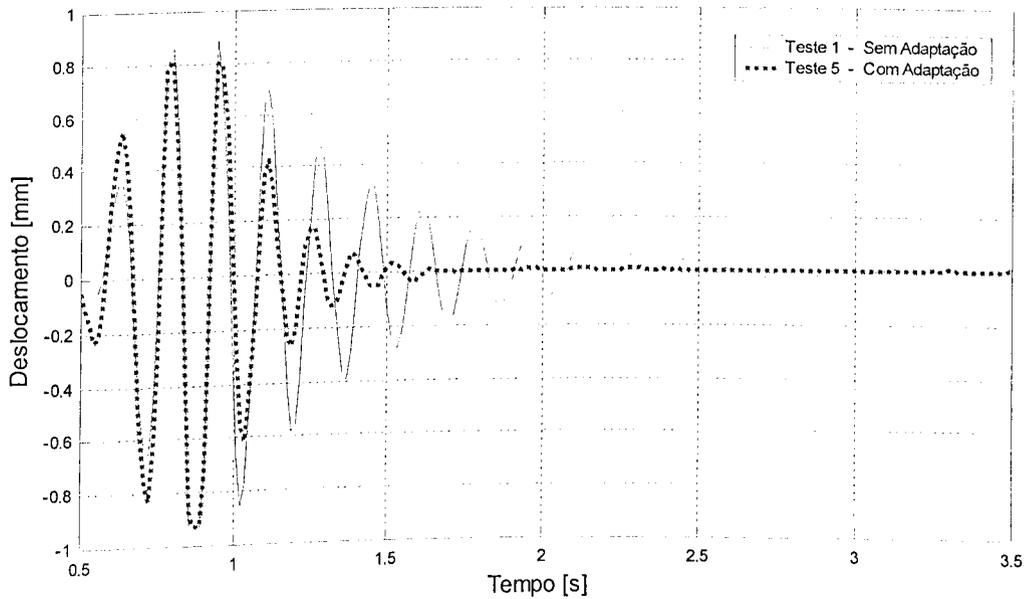


Figura 5.31 – Deslocamento lateral do sistema com e sem mecanismo de adaptação

Ao se observar os resultados obtidos com $1/K_v$ inicial igual à 1000 (fig. 5.25) e $1/K_v=30$ (fig. 5.32) observa-se que os dois controladores apresentam comportamentos bastante satisfatórios o que revela a pequena sensibilidade do algoritmo ao valor inicial de K_v . Como referência de projeto tem-se que quanto menor o valor inicial de K_v menor são os esforços de controle.

Para avaliar a capacidade de adaptação do procedimento de adaptação proposto foram aplicadas excitações seqüenciais ao sistema. No começo o sistema está em repouso, o mesmo é perturbado por uma força excitadora com duração de 0.5s; passados 3.5s, uma nova força excitadora é aplicada ao sistema. A Figura 5.32 mostra o comportamento do controlador à estas excitações.

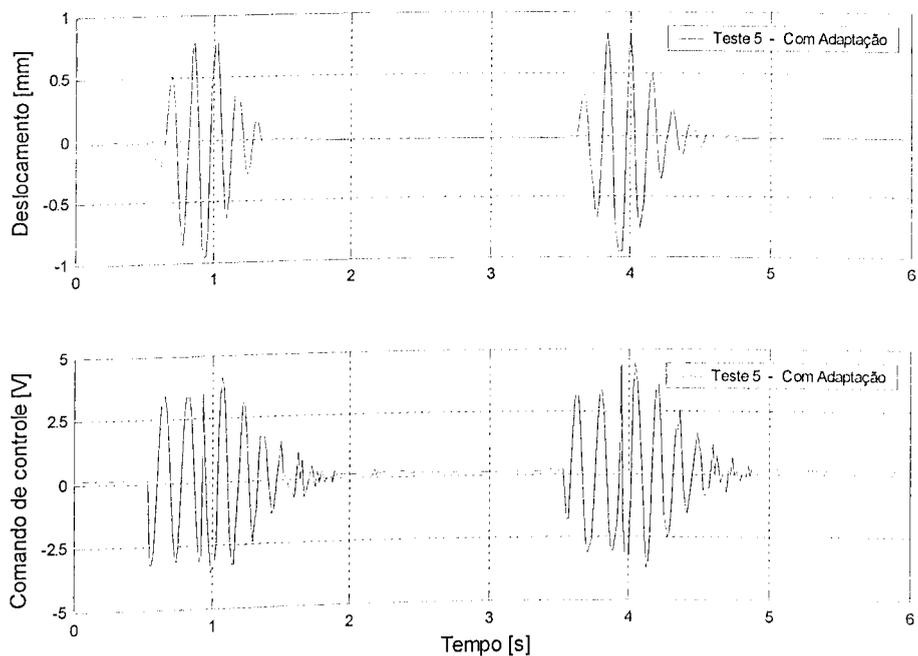


Figura 5.32 – Deslocamento lateral e comando de controle, para duas excitações seqüências, do sistema *fuzzy* adaptativo

A figura 5.33 revela o comportamento do valor de $1/K_v$ ao longo do tempo para a experiência descrita anteriormente. Como se pode verificar, o mecanismo de adaptação revela-se bastante eficiente, atendendo plenamente aos objetivos propostos de desempenho e adaptabilidade.

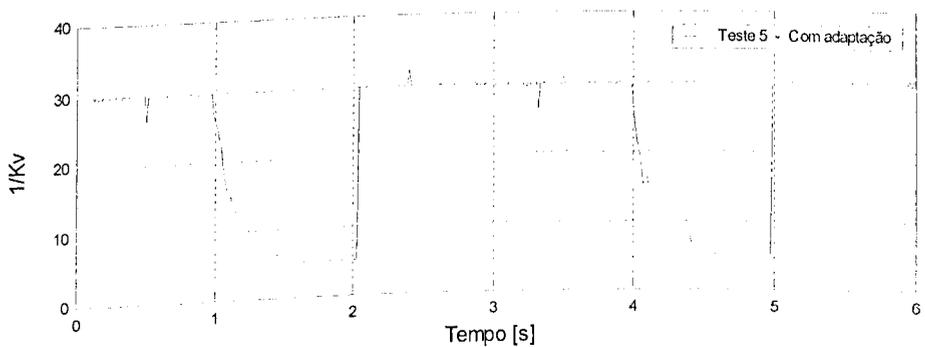


Figura 5.33 - Variação do fator de escala K_v

Capítulo VI

Conclusões e Futuros Desdobramentos

Este trabalho, tendo como motivação o controle de vibrações de sistemas mecânicos, apresentou uma formulação de controle ativo de vibrações baseado na lógica *fuzzy*. Propôs-se um novo mecanismo de adaptação do controlador *fuzzy*, que foi implementado num microcontrolador e avaliado experimentalmente em dois sistemas mecânicos: uma viga flexível e uma mesa vibratória.

Na realização do trabalho foram discutidos os fundamentos teóricos da lógica e do controle *fuzzy*, assim como apresentados os conceitos dos controladores *fuzzy* adaptativos. Fez-se ainda uma revisão geral dos microcontroladores, sendo que o mecanismo *fuzzy* de adaptação proposto foi ensaiado experimentalmente e os resultados dos testes discutidos.

O trabalho aponta para algumas conclusões importantes. No que concerne ao emprego de controladores *fuzzy* para atenuar vibrações mecânicas indesejadas, observou-se a potencialidade desta ferramenta. Não foi necessário elaborar modelos matemáticos dos sistemas e a compreensão intuitiva do processo foi de grande valia, especialmente na formulação da base de regras. Seguramente o tempo de desenvolvimento deste controlador foi inferior ao desenvolvimento de controladores clássicos convencionais já desenvolvidos para outros sistemas equivalentes (Ribeiro *et al.*, 1998).

O mecanismo de adaptação proposto mostrou-se também bastante eficiente, pois agregou ao sistema uma maior versatilidade diminuindo, de forma significativa, a sensibilidade do projeto à escolha de parâmetros do controlador *fuzzy*. Ficaram mais fáceis de serem definidos os universos do discurso envolvidos e o valor inicial do fator de escala (K_v) do mecanismo de adaptação.

A implementação do controlador e o respectivo mecanismo *fuzzy* de adaptação em microcontroladores mostrou-se adequada e promissora. A limitação na velocidade de processamento encontrado no caso do experimento da mesa vibratória se deu por conta da disponibilidade circunstancial do “chip” que se dispunha para serem realizados os testes. Na verdade existem outros componentes similares, na função e no preço, que atenderiam seguramente aos requisitos postos nos projetos aqui desenvolvidos. De uma maneira geral, o algoritmo de controle e o mecanismo *fuzzy* de adaptação proposto são de implementação

bastante fácil e não demandam requisitos computacionais importantes que inviabilizem o uso de microcontroladores; pelo contrário, o emprego destes componentes é cada vez mais promissor.

Os testes experimentais desenvolvidos propiciaram a validação em laboratório da metodologia proposta. Os controladores mostraram-se eficientes e amorteceram de forma significativa as vibrações introduzidas a despeito de envolverem tecnologias de sensoriamento e atuação muito diferentes. Num sistema trabalhou-se com sensores indutivos de posição e atuadores eletromagnéticos, enquanto que, no outro, se operou com acelerômetros e atuadores piezelétricos. Sob o ponto de vista do controlador (*software*) as diferenças entre o experimento da mesa e o da viga limitou-se a modificações nos parâmetros do universo do discurso e no valor inicial do ganho (K_v).

Diante do exposto, conclui-se que o objetivo inicial do trabalho qual seja o de ***“investigar o uso controladores fuzzy adaptativos, implementado em microcontroladores e voltados para o problema de controle de vibrações de sistemas mecânicos”*** foi plenamente alcançado e até superado com a formulação de ***um novo mecanismo de adaptação para o controlador fuzzy.***

Para o futuro, pretende-se investigar a aplicação desta técnica de controle em sistemas mecânicos mais complexos como placas, por exemplo. Pretende-se ampliar os estudos com o propósito de tratar sistemas mecânicos com múltiplas entradas e múltiplas saídas e assim controlar mais de um modo de vibrar. Os estudos devem evoluir na perspectiva de ampliar também o uso de materiais piezelétricos como sensores/atuadores neste tipo de problema. O emprego de microcontroladores especialmente projetados para tratar de sinais digitais (DSPs) neste tipo de aplicação é outra área que merece investigação futura.

Referências Bibliográficas

Ånström, K.J., and Wittenmark, B., *Adaptive Control*, Addison-Wesley, 1989.

Bandler, W. and Kohout, L.J., 1980, "Fuzzy power sets and fuzzy implication operators", *Fuzzy sets and systems* 14, pp. 155-174.

Bartoloni, G., Casalino, G., Davoli, F., Mastretta, M., Minciardi, R., and Morten, E., "Development of Performance Adaptive Fuzzy Controllers with Application to Continuous Casting Plants." In: R. Trappl (ed.), *Cybernetics and Systems Research*, Amsterdam, North-Holland, 1982, pp. 721-728.

Cunha Jr. S.S., 1999, "Estudo numérico e teórico de absorvedores dinâmicos de vibrações", *Dissertação de Mestrado*, Universidade Federal de Uberlândia, Uberlândia – MG.

Dubois e Prade, 1996, "New results about properties and semantics of fuzzy set-theoretic operators ". In P.P. Wang and S.K. Chang(Eds), *Fuzzy Sets*, pp. 59-75. New York, USA: Plenum press.

Frahm, H., 1991, *Device for Damping Vibrations of Bodies*, *US Patent* 989, 958.

Giles, R., 1976, "Lukasiewicz logic and fuzzy set theory", *International Journal of Man-Machine Studies* 8, 313-327.

Garcia D., 1998, "Curso de Introducción a los Microcontroladores", pp. 6-40, Bogota, Colombia.

Graham, B., *Fuzzy Identifications and Control*, Ph.D. thesis, University of Queensland, Australia, 1986.

Graham, B., and Newell, R., "Fuzzy Identifications and Control of a Liquid Level Rig," *Fuzzy Sets and System*, 26 (1988)255-273.

Graham, B., and Newell, R., "Fuzzy Adaptive Control of a First-order Process," *Fuzzy Sets and System*, 31 (1989)47-65.

Goguem, J.A., "On Fuzzy Robot Planning." In: Zadeh, L.A. et al. (eds), *Fuzzy Sets and Their Application to Cognitive and Decision Process*, New York, Academic Press, 1975, pp. 429-447.

Hamacher, H., 1978, "Über logische verknüpfungen unscharfer aussagen and deren zugehörige bewertungs funktionen", In G.J. Klir T, Trappl an L. Ricciardi (Eds.), *Progress in Cybernetics and System Research*, Vol.3, Washington (DC), USA: Hemisphere.

Jones, R. T., Pretlove, A. J., Eyre, R.; Two Case Studies in the Use of Tuned Vibration Absorber on Footbridges, *The Structural Engineer*, vol. 59B, n 2, pp. 27-32, 1981.

Koronev, B. G., Reznikov, L. M.; *Dynamic Vibration Absorbers. Theory and Technical Applications*. Jonh Wiley & Sons, Ltd., 1993.

Klasztorny, M.; Reduction of Steady-State Forced Vibrations of Structures with Dynamic Vibration Absorber, *Earthquake Engineering and Structural Dynamics*, vol. 24, 1155-1172, 1995.

Lee, C.C., 1980, "*Fuzzy* logic in control system: *fuzzy* logic controller-part i & ii", *IEEE Transactions on system, Man and Cybernetics*, Vo. 20, No.02, pp. 404-418.

Larsen, P.M., 1980, "Industrial applications of *fuzzy* logic control", *International Journal of Man-Machine Studies* 12, (1),3-10.

Mamdani, E.H., 1974, "Applications of *fuzzy* algorithm for simple dynamic plant", *Proceedings IEE* 121(12), 1585-1588.

Mamdani, E.H., and Assilian, S., 1975, "An experiment in linguistic synthesis whit a *fuzzy* logic controller", *International Journal of Man-Machine Studies* 7,1-13.

Maeda, A., Someya, R., and Funabashi, M., "A Self-tuning Algorithm for *Fuzzy* Membership Functions using a Computational Flow Network," *Proc. of the IFSA '91*, 1991.

Microchip PIC Microcontrollers, 2000, "Data Book, Technology Inc." PIC16C74A", USA.

- Marques, R.F.A., 2000, "Estudo Teórico e Numérico de Absorvedores Dinâmicos de Vibrações Ativos e Adaptativos", Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia-MG.
- Nessler, G. I., Brown, D. L., Stouffer, D. C., Maddox, K. C.; Design of a Viscoelastic Dynamic Absorber for Machine Tool Applications, Journal of Engineering for Industry, Transactions of the ASME, paper n 76-WA, pp. 620-623, 1977.
- Nomura, H., Hayashi, I., and Wakami, N., "A Self-tuning Method of *Fuzzy* Control by Descent Method," Proc. of the IFSA '91, Brussels, 1991, pp. 155-158.
- Pedrycz, W., Fuzzy Control Systems, 2nd revised edition, research Studies Publ., 1992.
- Ribeiro, J.F., Lépure, F.P. e Ribeiro, P.E., 1998, "O uso de atuadores magnéticos no controle ativo de vibrações de sistemas mecânicos", IV CEM-NNE, pp.439-444.
- Ram Y. M., Elhay, S.; The theory of a Multi-Degree-of-Freedom Dynamic Absorber, Journal of Sound and Vibration, vol. 195, n 4, pp. 607-615, 1996.
- Sun, J. Q., Jolly, M. R., Norris, M. A.; Passive, Adaptive and Active Tuned Vibration Absorber – A Survey, Transactions of the ASME, vol. 117, pp. 234-242, June 1995.
- Silva J. V. P., 1997, "Microcontroladores PIC", SP-Brasil.
- Sugeno, M., "An Introductory Survey of Fuzzy Control," Information Sciences, 36(1985) 59-83.
- Teodoro, E. B., 1994; Dynamics of a Power Line When Supported by a Compliant Energy Absorber, Ph. D. Thesis, ISU.
- Texeira, R.L., 2001, "Uma Metodologia De Projeto De Controladores Híbridos Inteligentes Com Aplicações No Controle Ativo De Vibrações Mecânicas", Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia-MG.
- Willmott, R., 1980, "Two fuzzier implications operators in the theory of *fuzzy* power sets". *Fuzzy Sets and Systems* 20,67-78.

- Weber, S., 1983, A general concept of *fuzzy* connectives, negations and implications based on t-norm and t-c0-norms. *Fuzzy Sets and Systems* 11,115-134.
- Wu, W., 1986, "Fuzzy reasoning and *fuzzy* relational equations". *Fuzzy Sets and Systems* 20,67-78.
- Yager, R.R., 1980a, "An approach to inference in approximate reasoning", *International Journal of Man-Machine Studies* 13, 323-338.
- Yager, R.R., 1980b, "On a general class of *fuzzy* connectives". *Fuzzy Sets and Systems* 4, pp. 235-242.
- Yager, R.R., and Filev, D.P., 1993, SLIDE: a simple adaptive defuzzification method. *IEEE Transactions on Fuzzy Systems*1 (1), 69-78.
- Yester, J. L. and McFall, R. H. [1992]. Fuzzy Logic Controller for active suspension. *Proceedings of Society of Automotive Engineers*, paper P-260, pp. 259-270.
- Yamashita, Y., Matsumoto, S., and Suzuki, M., "Start-up of a catalytic Reactor by *Fuzzy* Controller," *J. Chemical Engineering of Japan*, 21(1988) 277-281.
- Zadeh, L.A., 1965, "*Fuzzy* Sets, Information and Control", Vol. 8, pp. 338-353.
- Zadeh, L.A., 1973, "Outline of a new approach to the analysis of complex system and decision processes". *IEEE-Transactions on System, Man and Cybernetics* SMC-3, pp. 28-44.

Apêndice I

Programa em linguagem C para o controle da mesa vibratória (PC)

```
=====
//
// UNIVERSIDADE FEDERAL DE UBERLÂNDIA
// FACULDADE DE ENGENHARIA MECÂNICA
// LABORATÓRIO DE SISTEMAS MECÂNICOS
//
// Nome do Arquivo: control.cpp
// Projeto: Controle de uma mesa vibratória de 1gdl utilizando Lógica Fuzzy Adaptativa
// Data: Maio 2001
// Ferramentas Usadas: Microsoft Visual C++ 5.0
//
// Notas:
// - BIBLIOTECA MULTIMÍDIA DO WINDOWS
// Para usar os arquivos timer.h e timer.cpp é necessário ajustar:
// menu: Project/Settings
// 3- Link (Tab) - Category: General (drop-list)
// - Object/library modules: winmm.lib (editar)
// - MODO CONSOLE:
// Para usar os arquivos tim er.h e timer.cpp no MODO CONSOLE é necessário ajustar:
// menu: Project/Settings
// 1- Settings for: All Configurations (drop-list)
// 2- General (Tab) - Microsoft Fondation Class: Use MFC in a Shared DLL (drop-list)
=====
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <math.h>
#include "Fis.h" // Biblioteca do Controlador Nebuloso
#include "Lib//Timer//timer.h" // Biblioteca do Temporizador
#include "Lib//Daq//daq.h" // Biblioteca do Conversor AD e DA

char FismainFile[50];
FIS *Controle, *Adapta;
const double pi = acos(-1.0);

// Ajustes principais do controle
#define data_row_n 1
#define data_col_n 2
#define n_in 1
#define n_out 1 // Tempo de Amostragem (milessegundos)
#define TIME_OUT 5 // Número máximo de iteração do controle
#define MAX_CONTROL_LOOP 1000000 // Número máximo de Funções de Pertinência
#define MAX_MFS 10 // Número do canal de AD (canal de entrada)
#define AD_CHANNEL 5 // Número do canal de DA (canal de saída)
#define DA_CHANNEL 0

// Variáveis de temporização do controle
unsigned int cont, cont_adap = 0, j, i = 0; // Contadores auxiliares
unsigned int Time = 0; // Tempo atual do controle em Milessegundos
double Data_y[MAX_CONTROL_LOOP]; // Armazena a história temporal do sinal de entrada do AD (sinal do sensor de proximidade) a cada tempo de amostragem
double Data_dy[MAX_CONTROL_LOOP]; // Armazena a história temporal da derivada do sinal de entrada do AD a cada tempo de amostragem
double Data_c[MAX_CONTROL_LOOP]; // Armazena a história temporal do sinal de controle
double Data_Kv[MAX_CONTROL_LOOP]; // Armazena a história temporal do fator de adaptação
double ref=0.0, e0 = 0.0, NDC = 0.0; // Variáveis utilizadas na determinação do nível DC do sensor de proximidade
void ControlFnc(void); // Função de Controle
void SaveData(void); // Função de grava um arquivo com as variáveis pertinentes ao processo de controle

=====
// Modulo principal do programa
=====
void main(void)
{
// Inicialização
printf("////////////////////////////////////\n");
printf("// UNIVERSIDADE FEDERAL DE UBERLANDIA //\n");
}
```

```

printf("// FACULDADE DE ENGENHARIA MECANICA  /\n");
printf("// LABORATORIO DE SISTEMAS MECANICOS  /\n\n");
printf("// CONTROLE DE SISTEMAS MECANICOS  /\n");
printf("// VIA LÓGICA FUZZY ADAPTATIVA  /\n");
printf("// UBERLÂNDIA, MAIO DE 2001  /\n");
printf("////////////////////////////////////\n\n");

printf("////////////////////////////////////\n");
printf("// INICIO DO PROGRAMA DE CONTROLE /\n");
printf("////////////////////////////////////\n\n");

strcpy(FismainFile,"Control.fis"); // Lendo o arquivo ".fis" contendo a estrutura do controlador fuzzy responsável pelo
Controle = CreateFisStructure(FismainFile); // controle da mesa vibratória (controlador fuzzy principal)

strcpy(FismainFile,"Adapt.fis"); // Lendo o arquivo ".fis" contendo a estrutura do controlador fuzzy responsável pela
Adapta = CreateFisStructure(FismainFile); // adaptação do sistema de controle
InitDaq(); // comando de inicialização da placa
SetADInputMode(AD_BIPOLAR); // configura o modo de operação do conversor AD para -5 à +5 Volts (bipolar)

//=====
// Rotina para se adquirir (calcular) o valor DC do sinal do sensor. Este valor será tomado como referência do sistema em equilíbrio (sem vibração)
for (cont=1;cont<=1000;cont++) {

    e0 = ReadAD(AD_CHANNEL);
    ref=ref+e0;
}

ref = ref/cont;

NDC = ref;
printf("\n Nivel DC = %f \n",NDC);
//=====
Time = 0;

TimerSetTimeFnc(ControlFnc); // Ajustando para o timer a função de chamada
//=====
// Acionamento do controle
TimerStart(TIME_OUT); // Inicializando o timer com TIME_OUT em Milesegundos

printf("\n Tecle Qualquer tecla para finalizar o controle ... \n");

while( TimersRunning() ) // Aguarda qualquer tecla ser pressionada ou acabar o controle para finalizar o programa
{
    if( _kbhit() ) // Se uma tecla é pressionada a rotina de finalização é chamada
        TimerStop();
}
//=====

// Finalização
TimerStop(); // Finaliza o Timer

WriteDA(DA_CHANNEL,0.0); // Zera o canal de DA
UninitDaq(); // Finaliza a Placa de Aquisição
SaveData(); // Salva os dados em disco
printf("\n////////////////////////////////////\n");
printf( "// FIM DO PROGRAMA DE CONTROLE /\n");
printf( "////////////////////////////////////\n\n");
getch();
getch();

} // Fim da rotina main()

//=====
// Função de controle
//=====
void ControlFnc(void) // Chamada internamente pelo Sistema Operacional a cada TIME_OUT milesegundos
{
//=====
// Inicio da rotina de controle
// Variáveis do controle
static double e0=0.0,e1=0.0,e2=0.0,de=0.0,u=0.0;
static double dea1=0.0,dea2=0.0,dea3=0.0;

```

```

static double Sde=0.0,Su=0.0,aux_Sde=0.0,aux_Su=0.0,Fsde=1.0,Fsu=1.0;
static double ua1=0.0,ua2=0.0,ua3=0.0;
static double Kkv = 1.0,Kv = 1000.0,Ke = 1.0,aux_e=0.0,aux_de=0.0;
static double Refer=0.0,U=0.6,erro=0.0,derro=0.0;
static double E0[]= {-1.0 , 1.0}; // universo da variável de entrada, erro, do controlador fuzzy principal
static double DE0[]= {-1.0 , 1.0}; // universo da variável de entrada, derivada do erro, do controlador fuzzy principal
static double range_SDE[]= {-1.0 , 1.0}; // universo da variável de entrada, Sde, do mecanismo de adaptação fuzzy
static double range_SU[]= {-1.0 , 1.0}; // universo da variável de entrada, Su, do mecanismo de adaptação fuzzy
// Critério de parada
if( i >= MAX_CONTROL_LOOP ) // O programa de controle é interrompido quando o número máximo de dados é atingido
    if( Time > 10000 ) // O programa de controle é interrompido quando o tempo de controle atinge o valor
                        // estipulado para comparação
    {
        TimerStop(); // Finaliza o Timer
        return; // Sai da Função
    }

//=====
// Aquisição dos dados
e2 = e1; // Atualiza os valores das variáveis com os erro anteriores para posterior utilização no cálculo da derivada do erro
e1 = e0;
e0 = (ReadAD(AD_CHANNEL)-NDC); // Lê o canal AD de entrada (sinal do sensor de proximidade) subtraindo deste o nível DC
                                // previamente calculado.
//=====
// Definição da Referência do controle
Refer = 0.0; // Valor do sinal de referência. Para este caso, como se subtrai da leitura do sensor o seu valor DC
             // e como se deseja minimizar as vibrações em torno do ponto de equilíbrio da mesa, o valor do
             // sinal de referência é nulo
e0 = Refer-e0; // Calcula-se o valor do erro = (Posição de equilíbrio - Posição atual da mesa)
//=====
// Cálculo da derivada do erro
de = (e0-e1)/(TIME_OUT/1000.0); // Cálculo da Derivada do sinal de erro
erro = e0; // Atribui a novas variáveis os sinais do erro e sua derivada.
derro = de;
double fn = 12.00; // Valor da frequência natural da mesa vibratória em Hz
//=====
// Excitação do sistema mecânico
if (Time < 500) // A mesa vibratória será excitada por uma força senoidal em sua frequência natural
                // durante 500 ms (0.5segundos)
    u = U*sin(fn*2*pi*Time/1000.0); // Gera um sinal de controle/excitação senoidal de amplitude "U=0.6" e frequência "fn=12"
//=====
// Controle do sistema mecânico
if (Time >= 500) // Após o tempo de excitação o controlador se torna ativo
    {
        dea1 = dea2; // Atualização das variáveis utilizadas na determinação do valor médio das 3 últimas
        dea2 = dea3; // derivadas do erro em módulo (uma das entradas do mecanismo de adaptação)
        // Cálculo do módulo do valor da derivada do erro
        if (de<0)
            dea3 = de*(-1);
        if (de>=0)
            dea3 = de;
        Sde = (dea1+dea2+dea3)/3; // Valor de entrada do fuzzy adaptativo – entrada 1 = média do módulo dos 3 últimos valores da derivada do
//erro
        ua1 = ua2; // Atualização das variáveis utilizadas na determinação do valor médio dos 3 últimos
        ua2 = ua3; // sinais de controle em módulo (outra entrada do mecanismo de adaptação)
        // Cálculo do módulo do valor do sinal de controle
        if (u<0)
            ua3 = u*(-1);
        if (u>=0)
            ua3 = u;
        Su = (ua1+ua2+ua3)/3; // Valor de entrada do fuzzy adaptativo – entrada 2 = média do módulo dos 3 últimos valores sinal de controle
        Fsde = 100.0; // Valor do range do Sde (entrada 1) do fuzzy adaptativo
        Fsu = 4.0; // Valor do range do Su (entrada 2) do fuzzy adaptativo
        Fsde = 100.0*150; // Valor do range do Sde (entrada 1) do fuzzy adaptativo – valor modificado
        Fsu = 4.0*150; // Valor do range do Su (entrada 2) do fuzzy adaptativo – valor modificado
//=====
// Como as entradas do fuzzy adaptativo estão normalizadas no universo [-1 1], as variáveis utilizadas como entrada serão normalizadas sendo primeiramente
// divididas pelo valor do range e caso elas ainda sejam maiores que a unidade em módulo elas passarão por um limitador de modo a sempre se ter as variáveis
// de entrada no universo em que se encontra o fuzzy adaptativo

```

```

    aux_Sde = Sde/Fsde;
    aux_Su = Su/Fsu;
// Saturação das variáveis
    if( aux_Sde > range_SDE[1] ) aux_Sde = range_SDE[1];
    if( aux_Sde < range_SDE[0] ) aux_Sde = range_SDE[0];
    if( aux_Su > range_SU[1] ) aux_Su = range_SU[1];
    if( aux_Su < range_SU[0] ) aux_Su = range_SU[0];
//=====

    Ke = 10.0;          // Fator de escala para a entrada "erro" no controlador fuzzy principal
//=====
// Calculando o fator de adaptação através da avaliação do fuzzy adaptativo para as entradas normalizadas
    KKv = Fuzzy(data_row_n,data_col_n,aux_Sde,aux_Su,Adapta);
//=====
// Caso o fator de adaptação seja muito reduzido, ou seja, a adaptação seja mínima, para se evitar a divergência em um longo tempo de controle admite-se um
fator de adaptação igual a 1.0. Para pequenas adaptações o fator de escala não é alterado pelo fator de adaptação
    if ( ((1-KKv)*(1-KKv)) <= (0.002*0.002) )
        KKv = 1.0;
//=====
// Caso o sistema fica sme ser adaptado por 20 tempos de aquisição ou mais, o fator de escala da derivada do erro assume o valor inicial (Kv = 1000.0)
    if (KKv == 1.0)
        cont_adap = cont_adap+1;

    if (cont_adap >= 20) {
        Kv = 1000.0;
        cont_adap = 0;
    }
//=====
    Kv =Kv*KKv;          // Alteração do fator de escala da derivada do erro pelo fator de adaptação
                        // A modificação do fator de escala é feita recursivamente a cada amostragem
//=====
// Assim como no fuzzy adaptativo, o controlador fuzzy principal tem suas variáveis de entrada normalizadas no universo [-1 1]. Assim as variáveis de entrada
deste controlador, erro e derivada do erro, devem ser normalizadas. Para isto elas passarão primeiro por uma divisão pelo fator de escala, que tem o mesmo
significado que o range do controlador, e caso estas variáveis ainda sejam superiores à unidade em módulo elas serão submetidas a um limitador. Isto garante
que as variáveis de entrada sempre estarão no universo normalizado [-1 1]
    aux_e = e0/Ke;
    aux_de = de/Kv;

    // Saturação das variáveis e0 e df
    if( aux_e > E0[1] ) aux_e = E0[1];
    if( aux_e < E0[0] ) aux_e = E0[0];
    if( aux_de > DE0[1] ) aux_de = DE0[1];
    if( aux_de < DE0[0] ) aux_de = DE0[0];
//=====
// Cálculo do valor do sinal de controle a ser aplicada no sistema mecânico. Como a saída do controlador fuzzy principal é normalizada no universo [-1 1] e o
range de controle máximo do sistema físico é de [-1.7 1.7], a saída normalizada do fuzzy principal é multiplicada por 1.7. O sinal negativo aparece na
multiplicação apenas para adequar o sinal de controle ao sistema físico. Para eliminar este sinal, basta apenas inverter a saída do controlador

    u = -1.7*Fuzzy(data_row_n,data_col_n,aux_e,aux_de,Controle);
}
//=====
// Enviando o sinal de controle para a placa de aquisição

    if( fabs(u) > 1.7 ) u =1.7*u/fabs(u); // A favor da segurança limita-se o sinal de saída do controle em [-1.7 1.7] para não
                                        // causar danos ao driver de corrente do atuador

    WriteDA(DA_CHANNEL,u);          // Escrevendo no canal DA o valor do controle. O sinal de controle foi enviado pela placa de aquisição
//=====
// Criando os vetores de dados pertinente ao controlador. Estes dados serão gravados em arquivo posteriormente
    Data_y[i] = erro;
    Data_dy[i] = derro;
    Data_cf[i] = u;
    Data_Kv[i] = Kv;
// Incrementando as variáveis de temporização do controle
    i++;

```

```
        Time += TIME_OUT;
    }
//=====
// Rotina que salva em arquivo os dados pertinentes ao controlador
void SaveData(void)
{
    char ArName[20];
    FILE *pFileOut;
    unsigned int j,n;
    n = i-1;
    printf("\n NUMERO DE LOOPS EXECUTADOS: %d\n",n);
    printf("\n Entre com o nome do arquivo de saida de dados: ");
    scanf("%s",ArName);
    pFileOut = fopen(ArName,"w");
    if (pFileOut==NULL)
    {
        printf("Erro SaveData(): Arquivo de saida de dados não pode ser aberto\n");
        return;
    }
    printf("\n Salvando dados em disco...");
    for(j=0;j<n;j++)
        fprintf(pFileOut,"%f %f %f %f\n",Data_y[j],Data_dy[j],Data_c[j],Data_Kv[j]);
    printf("OK.\n");
    fclose(pFileOut);
}
//=====
```

Apêndice II

Programa em linguagem C para o controle da viga flexível (Microcontrolador)

```
=====
// UNIVERSIDADE FEDERAL DE UBERLÂNDIA
// FACULDADE DE ENGENHARIA MECÂNICA
// LABORATÓRIO DE SISTEMAS MECÂNICOS
=====
//
// CONTROLE DE SISTEMAS MECÂNICOS ATRAVÉS DE MICROCONTROLADORES PIC E LÓGICA FUZZY ADAPTATIVA
//
// PROJETO: Controle de uma viga flexível com atuadores piezelétricos incorporados utilizando lógica fuzzy adaptativa
//
// ==> Estrutura fuzzy do tipo sugeno de ordem zero com funções de pertinência triangulares
//
// FERRAMENTAS UTILIZADAS: ==> PIC C Compiler - CCS
//                               ==> MpLab - Microchip
//
// MICROCONTROLADOR UTILIZADO: PIC16F877 - 20MHz
//
// Entrada analógica: pino RA0 - pino A0 da porta A
//
// Saída do sinal de controle (PWM): pino RC2/CCP1 - pino C2 da porta C
//
=====
#include <CONTROL.H> // Inclui uma rotina com as configurações de registradores e portas de comunicação do microcontrolador utilizado
#fuses NOLVP,PUT,NOWDT,HS,NOBROWNOUT // Ajustando os bits de configuração do microcontrolador
#use standard_io(A)
#use standard_io(B)
#use standard_io(C)
#use standard_io(D)

#separate
fuzzy ();

#separate
fuzzy_adap ();

#separate
func_seno();

float erro,derro,erro_anterior,P,U,KKv,antecedente,consequente,B1,B2,input1,a,b,c, time,x,xx,seno,aux2,ref,n,aux,;
long int duty, Kv_duty, leitura;
int i,dummy, cont;
short int sinal;
float Sde,Su,de1,de2,de3,du1,du2,du3,erro_f,derro_f,Kv,Sde_f,Su_f;

#byte ADCON1=0x9F
#byte ADCON0=0X1F
=====
//
// MATRIZES DO CONTROLADOR FUZZY
float const MF_CONTROL [3][3]={
-2.0, -1.0, 0.0,
-1.0, 0.0, 1.0,
0.0, 1.0, 2.0};

int const REGRAS_ANTEC_CONTROL [5][2]={
0,0,
2,2,
1,0,
1,2,
1,1};

float const REGRAS_CONSEQ_CONTROL [5]={
5.0,
```

```

-5.0,
 5.0,
-5.0,
 0.0);
=====
// MATRIZES DO MECANISMO DE ADAPTAÇÃO FUZZY
float const MF_ADAP_S [3][3]={
-0.5, 0.0, 0.5,
 0.0, 0.5, 1.0,
 0.5, 1.0, 1.5};
// Regras anteriores
int const REGRAS_ANTEC_ADAP [9][2]={
 0,0,
 0,1,
 0,2,
 1,0,
 1,2,
 1,1,
 2,0,
 2,2,
 2,1};

float const REGRAS_CONSEQ_ADAP [9]={
 1.0,
 1.0,
 1.2,
 0.8,
 1.2,
 1.0,
 1.2,
 0.8,
 0.8};
=====
// ROTINA DE INTERRUPÇÃO ==> TEMPO DE AQUISIÇÃO
#int_timer1
timer1_isr() { // Timer 1 overflow

set_timer1(56161); // Usa o contador timer 1 configurando-o para gerar uma interrupção a cada 15 ms

output_high(PIN_B7); // Led para controle visual do operador

if ( (bit_test(port_b,6)) && (time >=0.5) )
{
time = 0; // Se a chave de partida for fechada, então o contador de tempo é zerado. Com isto o sistema é novamente excitado por 0,5 segundos
// e após este tempo o controlador atua novamente.
}

if (time <= 0.5)
{
func_seno(); // Enquanto o contador de tempo for inferior a 0,5 segundos, a função de cálculo do valor da excitação é chamada (excitação senoidal)
}

goto PWM; // Como neste tempo não haverá controle, apenas excitação, o programa é desviado para os comandos que fornecem a saída do
//microcontrolador
}

// Caso o tempo seja superior a 0,5 segundos, haverá ação de controle dada pelos comandos seguintes:
// Leitura do canal analógico e conversão deste em valor digital
dummy = Read_ADC();
*(&leitura+1) = *0x1E; //byte alto (HI) do registrador A/D
*(&leitura) = *0x9E; //byte baixo (LOW) do registrador A/D

input1 = (float) leitura;
input1 = (5.0*input1)/1023.0; // Conversão do valor inteiro relativo de 10 bits em valor absoluto do tipo ponto flutuante.
// 5,0 volts = 1023

erro = input1 - ref; // Cálculo do valor do erro = entrada - referência.

derro = (erro-erro_anterior)/(0.015); // Cálculo do valor da derivada do erro por diferenças finitas

erro_anterior = erro; // armazenando o valor anterior do erro

de1=de2; // Variáveis auxiliares para o
de2=de3; // cálculo do valor da soma das derivadas do erro

```

```

if (derro >= 0.0) // Realizando o cálculo do módulo do valor do erro
  de3 = derro;
else
  de3 = -derro;

du1=du2; // Variáveis auxiliares para o
du2=du3; // cálculo do valor da soma dos sinais de controle

if (U >= 0.0) // Realizando o cálculo do módulo do valor do sinal de controle
  du3 = U;
else
  du3 = -U;

//Cálculo do RMS do DERRO
Sde = (de1+de2+de3); // Cálculo da soma das 3 últimas derivadas do erro. Este valor é a primeira entrada do fuzzy adaptativo

//Cálculo do RMS de U
Su = (du1+du2+du3); // Cálculo da soma dos 3 últimos sinais de controle. Este valor é a segunda entrada do fuzzy adaptativo
Sde_f=Sde/150;
Su_f=Su/15;
fuzzy_adap(); // Chamando a sub-rotina do mecanismo de adaptação fuzzy
// Entradas = Sde_f e Su_f
// Saídas = fator de adaptação ==> Kkv = [0 1]
if ( (Kkv < 1.010) && (Kkv > 0.990) )
{
  // Para evitar adaptações constantes que sejam pouco significativas e que podem levar o sistema à instabilidade, caso o fator de adaptação seja
  // muito próximo de 1.000, o fator de adaptação é admitido igual a 1.000
  cont = cont+1; // Contador que irá indicar quantas vezes o fator de adaptação ficou próximo da unidade.
  Kkv = 1.000;
}
else
{
  cont = 0; // Caso o fator de adaptação não esteja próximo da unidade, o contador "cont" é zerado e o valor de Kkv é mantido igual ao resultado do fuzzy
  adaptativo
  output_low(PIN_B1); // Led para controle visual do operador
}

if (cont >= 20)
{
  // Caso durante 20 ciclos de aquisição o fator de adaptação tenha ficado próximo da unidade, isto indica que o sistema está controlado. Então o
  controlador é inicializado com as condições iniciais para os fatores de escala
  cont = 0; // Contador zerado
  Kv = 1000; // O controlador assume as condições iniciais para os fatores de escala
  output_high(PIN_B1); // Led para controle visual do operador
}

Kv = Kv*Kkv; // O fator de escala é dado de modo recursivo, multiplicando-se o valor anterior pelo fator de escala atual

if (Kv > 1000.0)
{
  // Para prevenir valores muito elevados de Kv, este é limitado em 1000,0
  Kv = 1000.0;
}

Kv_dutty = Kv; // Atribuindo o valor de Kv ao valor do ciclo em alta para a saída PWM do fator de escala, para posterior aquisição pelo computador

output_low(PIN_B0); // Led para controle visual do operador
output_low(PIN_B2); // Led para controle visual do operador

if (Kkv < 1.010)
  output_high(PIN_B2); // Led para controle visual do operador
if (Kkv > 0.990)
  output_high(PIN_B0); // Led para controle visual do operador

erro_f = erro; // Cálculo do valor da entrada erro do controlador fuzzy

derro_f = derro/Kv; // Cálculo do valor da entrada derivada do erro do controlador fuzzy

fuzzy(); // Chamada da sub-rotina de controle fuzzy. Entradas: erro_f = valor do erro real, derro_f = (valor da derivada do erro real) / Kv e Saída: U = sinal
de controle

PWM: // Procedimentos para saída dos sinais do tipo PWM: sinal de controle e fator de adaptação

```

```

if (U > 5.0) U = 5.0; // Limitação do sinal de controle em 5,0 volts
if (U < -5.0) U = -5.0; // para proteção da placa de aquisição e atuador

duty = (0.5+U/10.0)*1023; // Cálculo do valor do ciclo em alta do sinal PWM de controle
// Valor do ciclo = Sinal de controle
// 1023 = 5,0 volts
// 0 = -5,0 volts

set_pwm1_duty(duty); // Comando de saída do sinal PWM do sinal de controle
set_pwm2_duty(Kv_duty); // Comando de saída do sinal PWM do fator de adaptação
output_low(PIN_B7); // Led para controle visual do operador
}
//=====
// PROGRAMA PRINCIPAL
//
#separate
main() {

    setup_ccp1(CCP_PWM); // Configurando o pino CCP1 como saída PWM
    setup_ccp2(CCP_PWM); // Configurando o pino CCP2 como saída PWM

    setup_timer_2(T2_DIV_BY_4, 255, 1); // Configurando o contador interno de tempo TIMER2 que será usado como base de tempo para o sinal PWM. Com
    esta configuração o PWM tem uma frequência de 4,9 KHz utilizando um clock de 20 MHz

    set_tris_c(0b00000000); // Configurando a porta C como saída

    set_pwm1_duty(511); // Inicializando o PWM_1 com 511 que equivale a um sinal de controle nulo
    set_pwm2_duty(1000); // Inicializando o PWM_2 com 1000 que equivale a Kv = 1000 ==> sem adaptação

    erro = 0.0; // Inicializando as variáveis
    erro_anterior=0.0;
    derro = 0.0;

    time = 0.0; // Contador de tempo para excitação igual a zero
    x = 0.0; // x e xx são variáveis auxiliares para o cálculo da excitação
    xx = 0.0;

    Kv = 1000.0;
    Kkv = 1.0;
    Kv_duty = 1000;
    cont = 0;

    de1 = 0.0;
    de2 = 0.0;
    de3 = 0.0;
    du1 = 0.0;
    du2 = 0.0;
    du3 = 0.0;
    Sde = 0.0;
    Su = 0.0;
    erro_f = 0.0;
    derro_f = 0.0;

    setup_timer_1(T1_INTERNAL | T1_DIV_BY_8);
    set_timer1(56161);
    // Configurando o contador interno de tempo TIMER1 para gerar uma interrupção a cada 15 milissegundos. A cada interrupção de TIMER1 é realizada
    uma aquisição e controle do sistema

    setup_port_a(RA0_RA1_ANALOG_RA3_REF); // Configura os pinos RA0 e RA1 como entrada E o pino RA3 como tensão de referência dos canais A/D do
    microcontrolador

    setup_adc(ADC_CLOCK_INTERNAL); // Configuração do A/D
    bit_clear(ADCON1,3);
    bit_set(ADCON1,7);

    set_adc_channel(1); // Ativando o canal 1 = pino RA1. As conversões A/D serão feitas utilizando-se este pino de entrada, que é conectado ao acelerômetro

    set_tris_b(0b01000000); // Configurando a porta B

    output_high(PIN_B0); // Led para controle visual do operador

    ref = 0.0;
    n = 0;

```

```

loop1:    // Aguarda a chave de partida ser fechada para iniciar o cálculo do sinal de referência - sistema controlado
if ( bit_test(port_b,6) )
    goto referenc1;
goto loop1;

referenc1:
    delay_ms(50); // Aguarda 50 ms caso a chave de partida ainda esteja fechada

referencia:
    output_high(PIN_B2); // Led para controle visual do operador
// Realizando a aquisição do sinal de entrada analógico
dummy = Read_ADC();
*(&leitura+1) = *0x1E; //byte alto (HI) do registrador
*(&leitura) = *0x9E; //byte baixo (LOW) do registrador

input1 = (float) leitura;
input1 = (5.0*input1)/1023.0; // Conversão do valor de 10 bits em valor absoluto, 5,0 volts = 1023

ref = ref+input1; // Acumula o valor lido para posterior cálculo da média
n = n+1;
delay_ms(3);

if ((bit_test(port_b,6)) | (n>65000)) // Caso a chave de partida seja fechada, o cálculo do valor de referência é finalizado e tem-se início o processo de controle
    goto control;
goto referencia;

control:

    output_low(PIN_B2); // Led para controle visual do operador
    output_low(PIN_B0); // Led para controle visual do operador

    ref = ref/n; // Calculo do valor médio para o sinal de referência

    enable_interrupts(INT_TIMER1); // Abilitando a interrupção do contador TIMER1 a cada interrupção de TIMER1 é realizada uma aquisição
    enable_interrupts(GLOBAL); // Abilitação global das interrupções previamente abilitadas

    while(TRUE); // Loop infinito: Sempre que ocorrer uma interrupção devido a TIMER1, esta é atendida. Assim é realizada uma aquisição, cálculo e saída do
// sinal de controle. Após isto o programa volta para o Loop infinito aguardando outra interrupção de TIMER1, que ocorre a cada 15 milissegundos
}
//
// FIM DO PROGRAMA PRINCIPAL

//=====
// CÁLCULO DO VALOR DE PERTINÊNCIA DE UMA DADA ENTRADA EM UMA DADA MF
//
float valor_pertinencia(float x)
{
    // Função de pertinência triangular definida por a, b e c, a, b e c ==> Extremo esquerdo, central e direito da curva de pertinência, x ==> valor de entrada

    if (x <= a) return 0;
    if (x <= b && x >= a) return (x-a)/(b-a);
    if (x <= c && x >= b) return (c-x)/(c-b);
    if (x >= c) return 0;
}

//=====
// SUB-ROTINA DO CONTROLADOR FUZZY CÁLCULO DO VALOR DO SINAL DE CONTROLE
//
#separate
fuzzy()
{
// Limitação das variáveis de entrada ==> variáveis normalizadas
if (erro_f >= 1.0) erro_f= 1.0;
if (erro_f <= -1.0) erro_f=-1.0;
if (derro_f >= 1.0) derro_f= 1.0;
if (derro_f <= -1.0) derro_f=-1.0;

U=0;
P=0;

// Cálculo da saída fuzzy
for (i=0; i<=8; i++)

```

```

{ // Percorrendo todas as regras de controle fuzzy

a=MF_CONTROL[REGRAS_ANTEC_CONTROL[i][0]] [0]; // Recuperando os pontos a, b e c da
b=MF_CONTROL[REGRAS_ANTEC_CONTROL[i][0]] [1]; // curva de pertinência para a primeira
c=MF_CONTROL[REGRAS_ANTEC_CONTROL[i][0]] [2]; // entrada da regra "i"

B1 = valor_pertinencia(erro_f); Chamando a sub-rotina para cálculo do valor de pertinência da entrada erro_f na curva de pontos a, b e c

a=MF_CONTROL[REGRAS_ANTEC_CONTROL[i][1]] [0]; // Recuperando os pontos a, b e c da
b=MF_CONTROL[REGRAS_ANTEC_CONTROL[i][1]] [1]; // curva de pertinência para a segunda
c=MF_CONTROL[REGRAS_ANTEC_CONTROL[i][1]] [2]; // entrada da regra "i"

B2 = valor_pertinencia(derro_f); // Chamando a sub-rotina para cálculo do valor de pertinência da entrada derro_f na curva de pontos a, b e c

antecedente=(B1*B2); // Cálculo do valor do antecedente da regra "i" dado pelo produto entre os valores de pertinência desta regra

consequente=REGRAS_CONSEQ_CONTROL[i]; // Recuperando o valor do consequente da regra "i"

U=U+(antecedente*consequente); // Acumulando o produto entre a saída fuzzy e o antecedente de cada regra

P=P+antecedente; // Acumulando os valores dos antecedentes
}

if (P!=0) U=U/P; // Cálculo final da saída fuzzy do tipo sugeno de ordem zero

return(U); // Retornando o sinal de controle em volts
}

//=====
// SUB-ROTINA DO MECANISMO DE ADAPTAÇÃO FUZZY CÁLCULO DO FATOR DE ADAPTAÇÃO Kkv
#separate
fuzzy_adap()
{
// Limitação das variáveis de entrada ==> variáveis normalizadas
if (Sde_f >= 1.0) Sde_f = 1.0;
if (Sde_f <= 0.0) Sde_f = 0.0;
if (Su_f >= 1.0) Su_f = 1.0;
if (Su_f <= 0.0) Su_f = 0.0;

KKv=0;
P=0;

// Cálculo da saída fuzzy

for (i=0; i<=8; i++)
{ // Percorrendo todas as regras de controle fuzzy

a=MF_ADAP_S[REGRAS_ANTEC_ADAP[i][0]] [0]; // Recuperando os pontos a, b e c da
b=MF_ADAP_S[REGRAS_ANTEC_ADAP[i][0]] [1]; // curva de pertinência para a primeira entrada
c=MF_ADAP_S[REGRAS_ANTEC_ADAP[i][0]] [2]; // da regra "i"

B1 = valor_pertinencia(Sde_f); // Chamando a sub-rotina para cálculo do valor de pertinência da entrada Sde na curva de pontos a, b e c

a=MF_ADAP_S[REGRAS_ANTEC_ADAP[i][1]] [0]; // Recuperando os pontos a, b e c da
b=MF_ADAP_S[REGRAS_ANTEC_ADAP[i][1]] [1]; // curva de pertinência para a primeira entrada
c=MF_ADAP_S[REGRAS_ANTEC_ADAP[i][1]] [2]; // da regra "i"

B2 = valor_pertinencia(Su_f); // Chamando a sub-rotina para cálculo do valor de pertinência da entrada Su na curva de pontos a, b e c

antecedente=(B1*B2); // Cálculo do valor do antecedente da regra "i" dado pelo produto entre os valores de pertinência desta regra

consequente=REGRAS_CONSEQ_ADAP[i]; // Recuperando o valor do consequente da regra "i"

KKv=KKv+(antecedente*consequente); // Acumulando o produto entre a saída fuzzy e o antecedente de cada regra

P=P+antecedente; // Acumulando os valores dos antecedentes
}

if (P!=0) KKv=KKv/P; // Cálculo final da saída fuzzy do tipo sugeno de ordem zero

return(KKv); // Retornando o valor do fator de adaptação Kkv
}

```

```

}
//=====
// FUNCAO QUE CALCULA O VALOR DA EXCITAÇÃO SENOIDAL
#separate
func_seno() {

    time = time + 0.015; // Incrementando o contador de tempo sempre que a função de excitação é chamada

    xx = xx + 0.5419247; // 5.75 Hz e delta_time = 0.015
    // A função seno está sendo calculada por Série de Taylor. A cada intervalo de aquisição deve ser dado um incremento no valor de entrada para a série
    // de Taylor. Este valor de entrada é o ângulo, que depende da frequência e tempo de aquisição. Para a frequência da excitação senoidal de 5,75 Hz e tempo de
    // aquisição de 0,015 segundos o fator de incremento é 0,5419247.

    if (xx > 6.2832)
        xx = xx - 6.2832; // Limita-se o valor do ângulo em 2*PI. Se ele for superior é realizada a conversão para que ele fica inferior a 2*PI

    x = xx; // Variável auxiliar
    sinal = 0; // Variável auxiliar para verificar se o ângulo é superior a PI ==> seno < 0

    if (x > 3.1416)
    {
        // Se o ângulo for superior a PI ele é convertido para um ângulo inferior a PI que tenha mesmo seno, porém com sinal oposto
        // Tal manobra é realizada pelo fato da série de Taylor utilizada ter sido truncada em 4 termos, o que impossibilita o cálculo de valores negativos com
        // precisão satisfatória Não é utilizada uma série de Taylor com mais termos por motivo de tempo de cálculo

        x = xx - 3.1416;
        sinal = 1; // Se sinal=1 significa que o valor do seno calculado deve ter o sinal invertido
    }

    // Cálculo da série de taylor:
    //
    // seno = x - (x^3)/6 + (x^5)/120 - (x^7)/5040

    seno = x;
    aux2 = x*x*x;
    seno = seno - aux2/6;
    aux2 = aux2*x*x;
    seno = seno + aux2/120;
    aux2 = aux2*x*x;
    seno = seno - aux2/5040;

    if (sinal == 1)
        seno = (-1)*seno; // Se o ângulo é maior que PI o valor do sinal do seno deve ser invertido

    U = 5.0*seno; // A variável seno possui amplitude unitária, porém deseja-se que o sinal de excitação tenha amplitude de 5 volts.
}
//=====

```

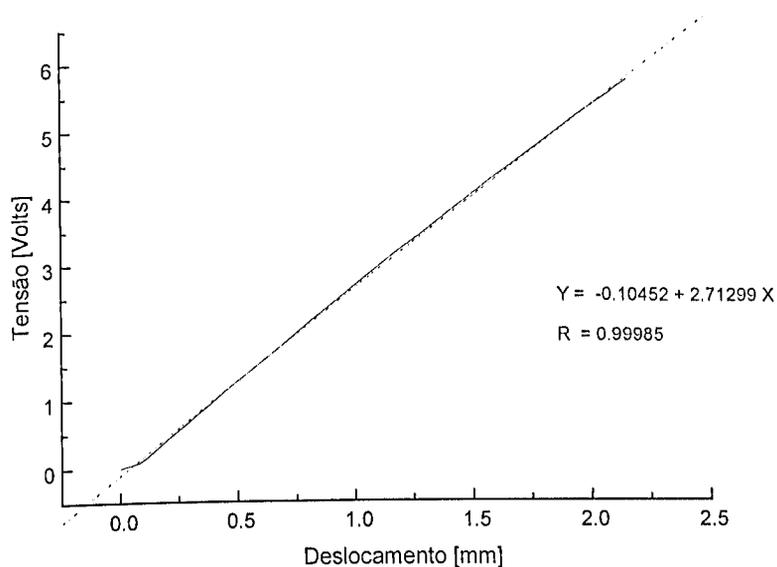
Apêndice III

Componentes do sistema mecânico da mesa vibratória

Equipamento	Fabricante	Especificações
Sensor de deslocamento eletromagnético	DYMAC	
2 Fontes de Potência		EMG 18131 - [30 Volts ; 2,5 A]
Fonte de Corrente	TECTROL	Modelo TC 50-02 - [50 V ; 2 A]
Osciloscópio	HP - Hewlett Packard	Oscilloscope 54603B
Analisador de Sinais.	Scientific-Atlanta Spectral Dynamics	SD380 Signal analyser Banda de análise 0- 100 Hz Janela retangular - 800 linhas Trigger externo
Condicionador de Sinais do Sensor de deslocamento eletromagnético	Laboratório de Sistemas Mecânicos	
Atuador Eletromagnético	Laboratório de Sistemas Mecânicos	
Driver de Corrente	Laboratório de Sistemas Mecânicos	

III.I Curva de calibração do sensor de deslocamento eletromagnético

O sensor de proximidade, do tipo DYMAC, detecta as posições da mesa. Ele não apresenta contato entre as partes. É um transdutor linear do tipo deslocamento-tensão, conforme mostra sua curva de calibração estática.



Apêndice IV

Componentes do sistema mecânico da viga flexível engastada

Equipamento	Fabricante	Especificações
Sensor de deslocamento eletromagnético	DYMAC	
Acelerômetro	Brüel & Kjaer	Force Transducer Type 4375
Condicionador de Sinais Do Pêndulo de Impacto	Brüel & Kjaer	Charge Amplifier Type 2635 [0.351 pC / m / s ²] e [316 mV / mm] Lower Freq. Limit 1Hz Upper Freq. Limit 1KHz
Atuador Piezelétrico	ACX	QP10N – Máxima Tensão [150 V]
Amplificador de sinal Do atuador piezelétrico	AVC Instrumentation	Power amplifier 790 series - [30 V / V]
Osciloscópio	HP - Hewlett Packard	Oscilloscope 54603B
Analizador de Sinais.	Scientific-Atlanta Spectral Dynamics	SD380 Signal analyser Banda de análise 0- 500 Hz Janela retangular - 800 linhas Trigger externo
Condicionador de Sinais do Sensor de deslocamento eletromagnético	Laboratório de Sistemas Mecânicos	
Placa de aquisição	UEI - United Electronic Industries	12 Bits
PIC 16F877	Microchip	40 pinos

VI.I Curva de calibração do sensor de deslocamento eletromagnético

Uma nova calibração foi necessária pois neste experimento com a viga o sensor de proximidade detecta um outro tipo de material (aço inoxidável).

