8,1 (oito e um)

ESCOLA POLITÉCNICA
Universidade de São Paulo

PMC 580
RELATÓRIO FINAL

Caracterização e Simulação de Atuadores Piezelétricos Flextensionais Utilizando técnicas de Interferometria Laser.

Sérgio Henrique Soares Ferreira *2368283*
Orientador: *Prof. Dr. Emílio C. Neli Silva*
Colaborador: *Doutorando Gilder Nader*

Serviço de Bibliotecas
Biblioteca de Engenharia Mecânica, Naval e Oceânica

ÍNDICE

1 Introdução

O objetivo deste trabalho consiste em utilizar técnicas de interferometria laser para medir deslocamentos gerados por atuadores piezelétricos. Uma vez obtida a caracterização dos atuadores experimentalmente, comparam-se os resultados com simulações a serem feitas utilizando-se Modelos de Elementos Finitos dos atuadores. A técnica de interferometria laser utilizada é baseada no método de Michelson e o interferômetro está montado no laboratório de ótica da Escola. A medição de cada atuador é um processo demorado, pois exige o alinhamento e ajuste do interferômetro em cada caso sendo que cada medição pode demorar até uma semana para ser realizada. Além disso, o processo que vinha sendo utilizado para fazer a aquisição dos sinais do interferômetro era um processo demorado e bastante manual, portanto também foi desenvolvido durante este trabalho rotinas automáticas de aquisição de dados.

2 Motivação

Está sendo desenvolvido na Escola Politécnica um dispositivo de atuação nanométrico composto por dispositivos flexíveis atuados por cerâmicas piezelétricas (atuadores piezelétricos flextensionais). Como a ordem de grandeza dos deslocamentos a serem medidos é de nanômetros, torna-se necessário desenvolver uma forma de se obter e tratar esses deslocamentos de uma forma dinâmica e eficiente. Com a utilização de interferometria laser torna-se possível fazer a aquisição desses deslocamentos dinamicamente, transferir as medidas para um computador e utiliza-los para os devidos fins.

Originalmente o interferômetro e os recursos disponíveis para operá-lo eram utilizados somente por alunos de pós-graduação. Pretende-se a partir deste trabalho demonstrar a possibilidade de ministrar os conceitos envolvidos a um aluno de graduação.

3 Desenvolvimento Teórico

3.1 Atuadores Piezelétricos Flextensionais

Os Atuadores a serem caracterizados constituem-se de cerâmicas piezelétricas acopladas a dispositivos flexitensionais, desenvolvidos a partir de métodos de otimização topológica. A seguir ambos são descritos com mais detalhes.

3.1.1 Cerâmicas Piezelétricas

Cerâmicas Piezelétricas são responsáveis por converter energia mecânica em energia elétrica e vice-versa. Possuem as mais diversas aplicações como, por exemplo, sensores, acelerômetros, transdutores de ultra-som, motores etc.

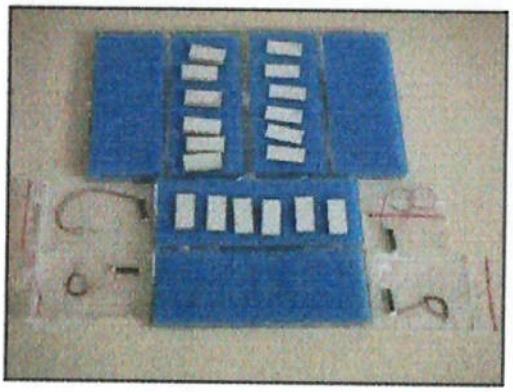

Fig. 1 – Cerâmicas Piezelétricas

Quando excitadas por uma tensão elétrica, dependendo do eixo de polarização as cerâmicas contraem-se ou dilatam-se, de acordo com a figura:

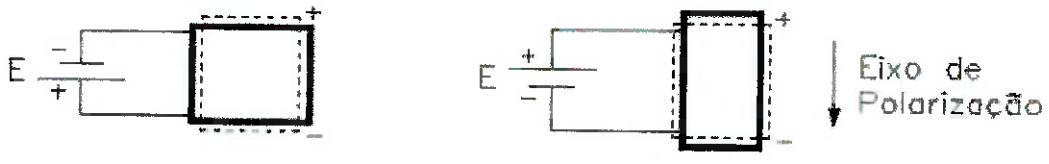

Fig. 2 – Cerâmicas Piezelétricas – Eixo de Polarização

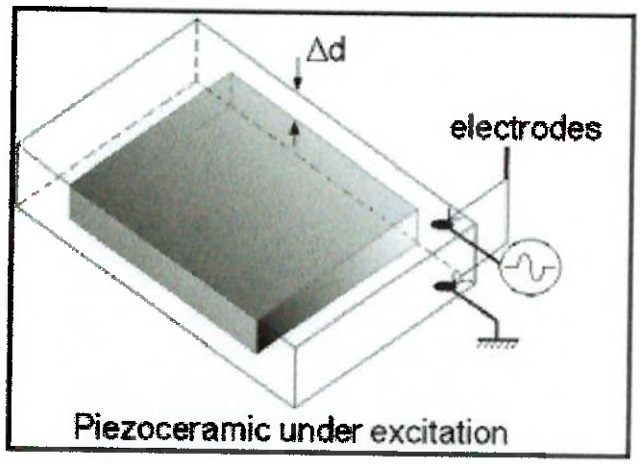

Fig. 3 – Cerâmica Piezelétrica sob excitação.

3.1.2 Dispositivos Flextensionais

Os dispositivos flextensionais visam amplificar e/ou direcionar o movimento e a força aplicada pela cerâmica piezelétrica de forma otimizada.

Para tanto são rodados programas e algoritmos de otimização topológica para obter a forma final do dispositivo.

Na figura seguinte está um exemplo dos passos para a obtenção do dispositivo.

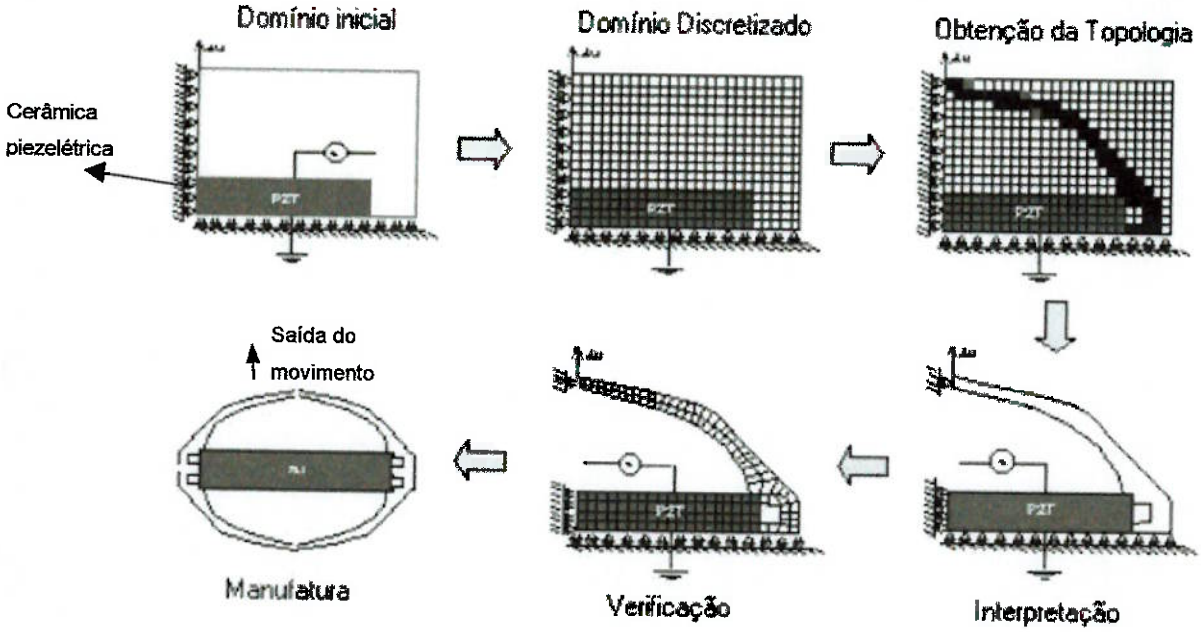

Fig. 4 –Otimização Topológica

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

3.2 Interferômetro de Michelson

O interferômetro, como o próprio nome já diz, é um instrumento destinado a medir interferências. O interferômetro foi desenvolvido por Albert Abraham Michelson (1852 - 1931) em 1881, que recebeu em 1907 o prêmio Nobel por sua descoberta. Com o aparelho original, Michelson foi capaz de medir comprimentos e variações de comprimentos de ondas com muita precisão.

O interferômetro permite a medida com precisão do tamanho de objetos muito pequenos, como o diâmetro de um fio de cabelo, o diâmetro de uma célula de sangue, etc. Michelson na época utilizou seu interferômetro para medir o comprimento da barra de irídio utilizada como padrão internacional para a unidade de metro. Ele obteve então que o comprimento desta barra era igual a 1.553.163,5 comprimentos de onda de luz monocromática vermelha. Em razão desta medida, Michelson recebeu em 1907 o Prêmio Nobel de física, sendo o primeiro americano a receber esta honra. Em 1961, finalmente a barra de irídio foi abandonada como padrão de medida e o metro foi redefinido em termos do comprimento de onda da luz. Em 1983, no entanto, os avanços tecnológicos da época demandavam uma precisão maior, e o padrão de medida foi novamente substituído pela distância que a luz percorre em 1/c segundos, onde c é a velocidade da luz em metros por segundo.

O interferômetro é um instrumento de medida versátil e eficiente. É usado principalmente em medidas de comprimento, mas também encontra grande uso na determinação de índices de refração e caminho ótico.

3.3 Interferometria Laser

3.3.1 Funcionamento

O interferômetro de Michelson é a forma fundamental da grande variedade de interferômetros de dois feixes. No esquema a seguir, a luz vem da fonte L, incide na

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

placa paralela P, sofre uma refração até incidir na outra superfície semi-espelhada, aonde irá se dividir em dois feixes, os quais irão atingir os espelhos A1 e A2 perpendicularmente.

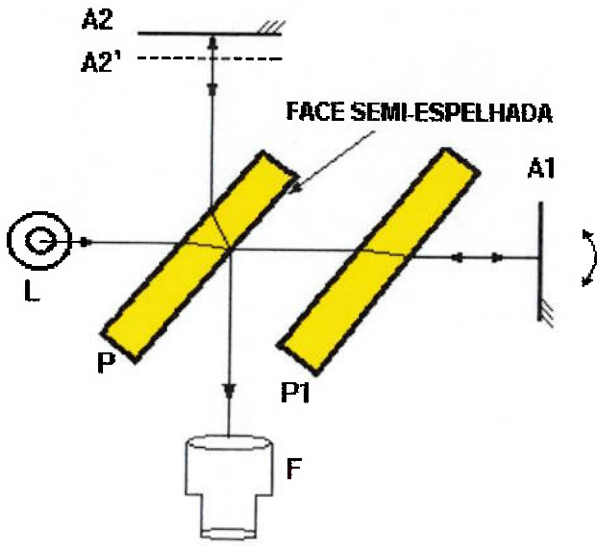

Fig. 5 – Interferômetro de Michelson – Placas paralelas.

Os retornos dos feixes irão atingir a face semi-espelhada da placa P, e as franjas de interferência podem ser vistas diretamente a olho nu, ou através de um telescópio F. Notar que a luz refletida por A2 passa através da placa P três vezes, enquanto que a luz refletida por A1 passa apenas uma vez. A placa compensadora P1 é idêntica na espessura e no paralelismo à placa P e sua inserção vai igualar os caminhos dos dois feixes.

Quando os espelhos estiverem a distâncias iguais e perpendiculares, o campo de interferência será uniforme. Quando as superfícies refletoras não estiverem perpendiculares, as franjas passam de circulares a linhas. Quanto maior a diferença entre as distâncias dos espelhos A1 e A2 à placa P, mais círculos concêntricos de interferência serão observados.

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

3.3.2 Modelo simplificado

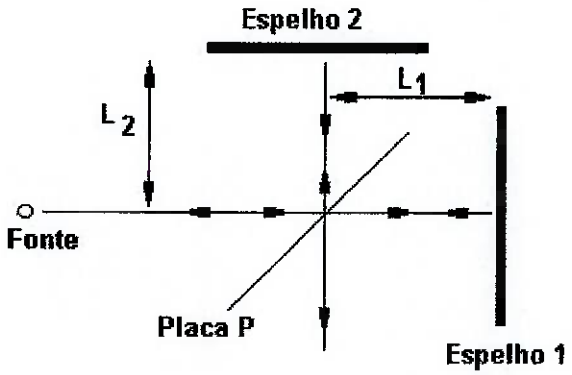

Fig. 6 – Interferômetro de Michelson – Deslocamentos.

No modelo simplificado da figura, a diferença de percursos no ar vale $2(L_1-L_2)$, e a diferença de fase entre os dois feixes recombinados é assim :

$$\Delta\Phi = \frac{2\pi}{\lambda} 2\,(L_1 - L_2) + \Delta\phi_0$$

Onde $\Delta\Phi$ é a diferença de fase introduzida pelo divisor de feixe (P). Como um feixe faz uma reflexão interna e o outro uma reflexão externa, $\Delta\Phi$ não é exatamente π por causa do filme refletor depositado.

Obtendo por deslocamento longitudinal L de um dos espelhos, a intensidade I na saída será:

$$I_s = I_1 + I_2 + 2\sqrt{I_1 + I_2}\,\cos\left[\frac{4\pi}{\lambda}(L_1 - L_2) + \frac{4\pi}{\lambda}\Delta L\right]$$

$$\text{e se } I_1 = I_2 = I \Rightarrow I_s = 2I\left[1 + \cos\left(\Delta\Phi_0 + \frac{4\pi}{\lambda}\Delta L\right)\right]$$

Assim toda vez que o deslocamento do espelho móvel atingir um valor múltiplo de l/2, o valor da intensidade se repete.

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

A intensidade transmitida no caso especial de caminhos ópticos iguais seria máxima para todos os comprimentos de onda, não fosse o filme semi-refletor requerer um suporte de vidro. Neste caso a reflexão acontece com uma defasagem devida à diferença de índices de refração dos meios vidro-ar, dado uma defasagem somente na interface ar-vidro e, conseqüentemente transmissão nula.

A presença das lâminas de vidro traz também um sistema paralelo de reflexões na segunda face e conseqüentemente de franjas. A intensidade deste sistema secundário é fraca, e difícil de ser observada.

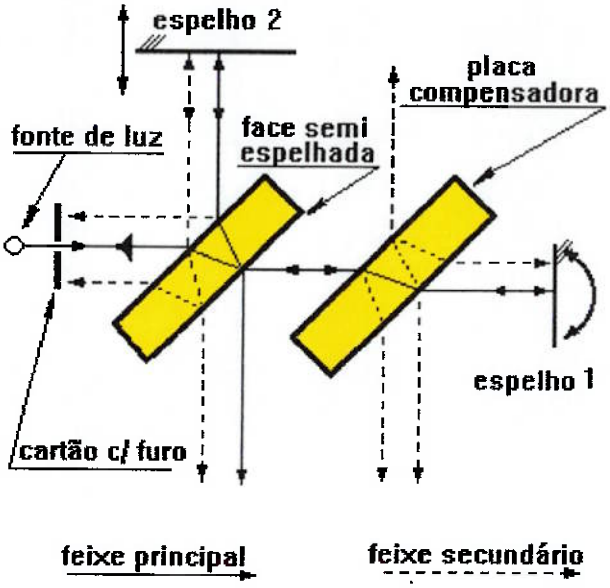

Fig. 7 – Interferômetro de Michelson – Sistema de feixes secundário.

Posicionando-se um foto-detector no lugar do telescópio F, pode-se medir a intensidade resultante dos dois feixes e conseqüentemente calcular a defasagem entre os dois feixes luminosos. Ou ainda, pode-se calcular o deslocamento nanométrico de um dos espelhos permanecendo com o outro parado, desde que esse deslocamento permaneça na faixa aproximadamente linear da intensidade luminosa, podendo-se medir com sucesso deslocamentos da ordem de $\lambda / 4$.

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

3.3.3 Montagem do Interferômetro

O Interferômetro de Michelson disponível e montado no laboratório de ótica está esquematizado a seguir:

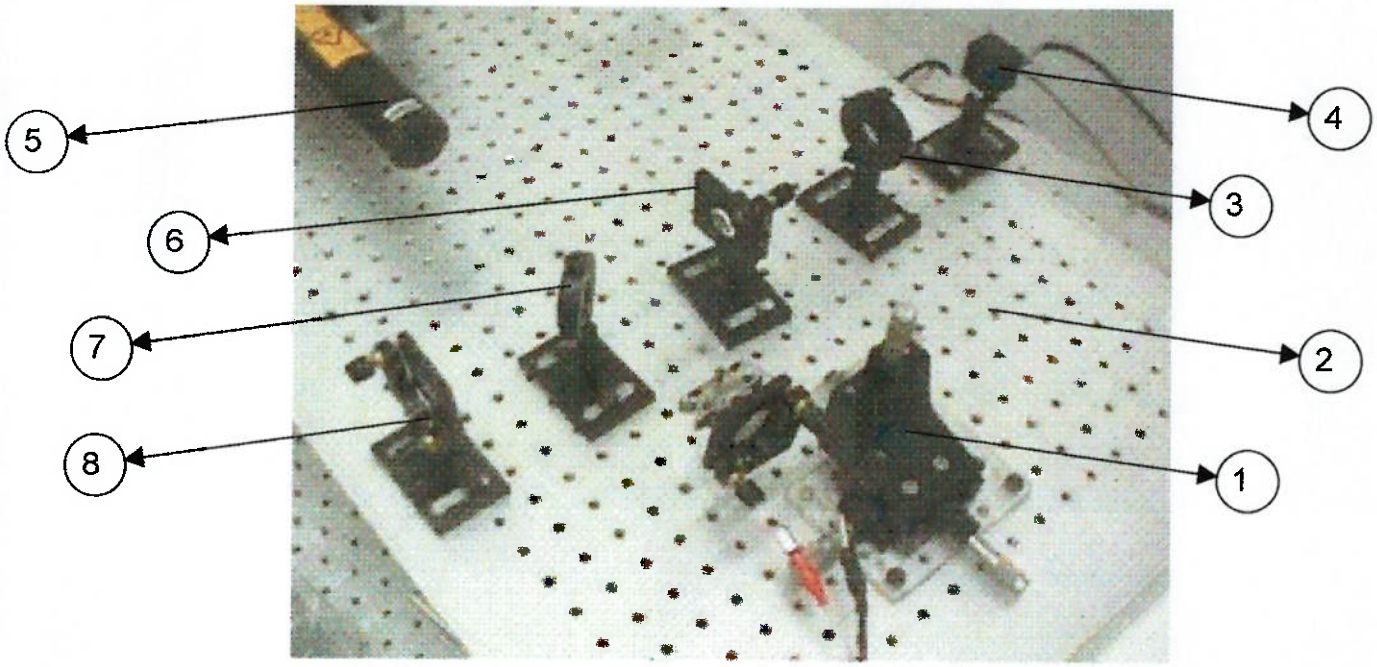

Fig. 8 – Interferômetro de Michelson – Configuração.

A constituição utilizada para o alinhamento e a operação do interferômetro, de acordo com a figura é:

1. Suporte da amostra com um parafuso nanométrico;
2. Base de apoio do sistema / Mesa ótica;
3. Vidro compensador de caminho ótico;
4. Foto-receptor;
5. Emissor laser He-Ne (potência nominal de saída 0,5 mW e comprimento de onda de aproximadamente 633 nm);
6. Lente convergente;
7. Divisor de feixe dielétrico com parafuso para fixação na base;

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

8. Espelho referência com controle de posicionamento e parafusos para fixação na base;

A base foi colocada em uma mesa ótica, que é um local estável e firme, destinada a minimizar a presença de vibrações ou perturbações que possam atrapalhar o experimento e alterar as medições.

Os espelhos devem ser alinhados de tal forma que o ângulo formado por eles seja aproximadamente 90°. Já o Divisor de Feixe deve ser posicionado da tal forma que os ângulos formados entre os espelho e o Divisor de Feixe seja aproximadamente 45°, quanto melhor o alinhamento, melhor o padrão de interferência gerado.

Para a medição dos deslocamentos dos atuadores piezelétricos, o anteparo é substituído por um foto-detector, que tem como finalidade captar as variações de intensidade luminosa geradas pela interferência da luz. O foto-detector é ligado num osciloscópio que se comunica com um PC, onde, com o auxilio do software MATLAB, serão calculados os deslocamentos medidos. Para a excitação dos transdutores utiliza-se um gerador de funções e um amplificador. Acoplado aos transdutores piezelétricos flextensionais encontra-se uma superfície refletora.

3.3.4 Recursos Necessários

Para o desenvolvimento deste trabalho estavam disponíveis no laboratório de ótica da Escola Politécnica todos os recursos necessários, incluindo o interferômetro, os atuadores piezelétricos, os equipamentos eletrônicos para geração e aquisição de sinal bem como o computador que será utilizado para analisar os sinais capturados.

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

3.4 O Método dos Elementos Finitos e Suas Aplicações

O Método de Elementos Finitos (MEF) é um método matemático/computacional para análise de problemas do contínuo. O método permite que a peça em estudo tenha forma geométrica, carregamento e condições de contorno quaisquer. Ocorre uma semelhança física entre o modelo FEA (Finite Element Analysis) e a situação física real, não sendo o modelo uma abstração matemática difícil de ser visualizada.

Inicialmente o MEF foi usado em cálculo estrutural (década de 60), hoje é largamente aplicado em problemas de campo (calor, fluidos, campo elétrico e magnético). Algumas das análises que podem serem executadas por softwares de elementos finitos:

- Estática linear de tensões e deformações (edifícios, pontes, torres, componentes mecânicos em geral, tubulações industriais,...).
- Dinâmica (modos de vibração e freqüências naturais).
- Não linear de tensões e deformações (conformação, grandes deformações,...).
- Térmica (transmissão de calor em regime permanente e transiente).
- Tensões devido ao carregamento térmico (tubulações industriais).
- Escoamento de fluídos (aerodinâmica e hidrodinâmica).
- Campos elétricos (condutores, isolantes, eletrodeposição e corrosão) e magnéticos.

É próprio da mente humana querer subdividir os sistemas em seus componentes individuais ou, mais propriamente, em seus elementos. Assim, surge quase que naturalmente a idéia de que, a partir do entendimento do comportamento de cada elemento, poder-se-á entender o funcionamento do conjunto, por mais complexo que possa parecer. Ou seja, compreender o todo, entendidas as partes. Em muitas situações práticas, a identificação dos componentes de um sistema, ou mais particularmente de uma estrutura, parece-nos uma tarefa quase que óbvia. Por

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

exemplo, para uma estrutura espacial metálica constituída unicamente por vigas, é natural identificar os componentes individuais de vigas, ou elementos, que conectados entre si nas juntas ou nós estruturais constituirão o conjunto estrutural.

Outra idéia bastante comum e que se torna fundamental na análise estrutural é a Idéia de rigidez. Todos possuem a idéia de rigidez desde as primeiras aplicações com os elementos elásticos (ou molas) da física básica. O conceito de mola equivalente (ou rigidez equivalente) a um conjunto de molas também faz parte do dia-a-dia do técnico. Assim ocorre também ao abordar-se a análise estrutural. A rigidez da estrutura depende da rigidez de cada um de seus elementos. Pode-se avaliar a rigidez da estrutura a partir da rigidez de cada um de seus elementos.

Eis então, de um modo simples, a primeira idéia do Método dos Elementos Finitos:

A estrutura ou de forma geral O Componente Mecânico é subdividido em um número finito de partes (os elementos) que são conectados entre si. A estrutura então pode ser representada como uma montagem de elementos que constitui um modelo matemático, também chamado de modelo estrutural ou idealização estrutural.

Diversos componentes podem ser representados dessa forma: a caixa estrutural completa de um veículo, componentes de um chassi, pára-choques, eixos, componentes de máquinas, carcaça de diferencial e no caso do presente trabalho os atuadores piezelétricos flextensionais.

O modelo de elementos finitos é composto por elementos conectados entre si por nós, formando a malha de elementos finitos, conforme a figura 9:

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

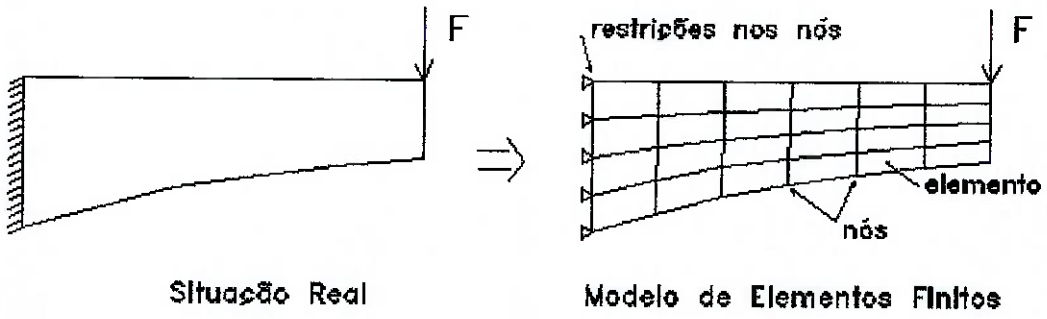

Fig. 9 – Modelo de Elementos Finitos.

Dois aspectos iniciais constituem as características principais do método dos elementos finitos:

- A subdivisão da estrutura em elementos, isto é, a malha de Elementos Finitos;
- A escolha do elemento apropriado para modelar uma dada situação física.

A escolha do tamanho adequado da malha não parece óbvia em uma estrutura contínua. E realmente não é. Depende do conhecimento das propriedades do elemento escolhido para representação do problema, que é a mais fundamental característica do método.

Do ponto de vista prático, os "softwares" de Elementos Finitos oferecem uma biblioteca de elementos do programa, contendo diversos elementos, cada qual tentando representar um diferente comportamento físico conhecido da Mecânica (placas, cascas, membranas, sólidos, vigas, etc.). Esse comportamento é descrito por intermédio de funções matemáticas que em última análise contabilizam a propriedade desejada daquele elemento individual. Dispondo da biblioteca de elementos, podemos avaliar a propriedade da estrutura inteira a partir da propriedade de cada elemento.

Tendo montado o modelo estrutural, pode-se determinar a configuração deformada da estrutura no computador, por intermédio dos deslocamentos dos nós, qualquer que

seja a forma da estrutura e o tipo de carregamento. Está-se então em condições de determinar o estado de tensões na estrutura e fazer a avaliação de seu comportamento mecânico. Pode-se ainda determinar, uma vez fornecidas as propriedades elétricas e piezelétricas do material, as distribuições de corrente, a impedância e as deformações devido às voltagens impostas em um componente piezelétrico.

Assim, o Método dos Elementos Finitos é uma ferramenta extremamente valiosa para ajudar as equipes de engenharia em uma das tarefas mais importantes no desenvolvimento de um produto, que é determinar o seu comportamento mecânico e garantir que não haverá falha em condições normais de operação, assim como o componente irá operar de acordo com o esperado nos requisitos de projeto.

Para executar uma análise que conduza a decisões adequadas, devem ser observados alguns pré-requisitos:

- Entendimento claro do problema físico a ser simulado;
- Conhecimento do comportamento desejado (critério de projeto);
- Propriedades dos materiais envolvidos;
- Características dos elementos finitos envolvidos na análise;
- Definição da região de interesse, definindo a extensão do modelo de análise;
- Condições de contorno - cargas e vínculos estruturais.

Assim, o modelo proposto deve representar trecho a trecho da forma mais acurada possível o que ocorre na estrutura real. Essa representação só poderá ser feita se o analista conhecer o comportamento dos elementos finitos disponíveis e identificar no objeto de análise esse comportamento, de sorte a utilizar o elemento adequado para cada aplicação. Em resumo, os programas de elementos finitos não são, sob hipótese alguma, ferramentas complexas, que independem do julgamento do analista, pois constituem um auxílio a ele, que deve conhecer os conceitos fundamentais do MEF, e o comportamento dos principais elementos da biblioteca do programa. Uma base

conceitual adequada é o melhor caminho para obter bons resultados nas aplicações práticas do dia-a-dia com os softwares de Elementos Finitos.

3.4.1 Exemplo de Aplicação

A seguir um exemplo de aplicação do método dos elementos finitos utilizando o programa ANSYS. No exemplo tem-se um braço que será engastado no furo da esquerda e no furo inferior será aplicada uma força vertical para baixo:

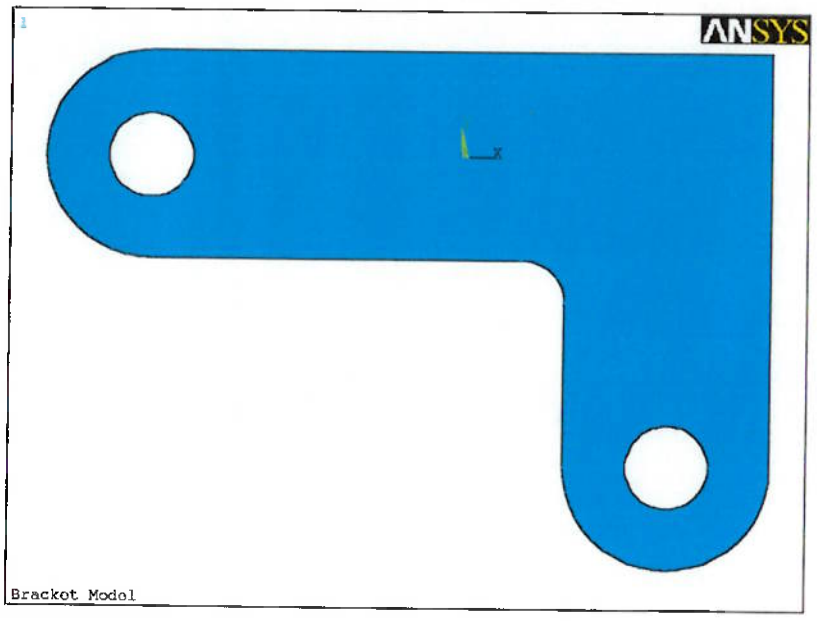

Fig. 10 – Modelo de Área Ansys.

Após a discretização da área e a criação dos elementos da malha:

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

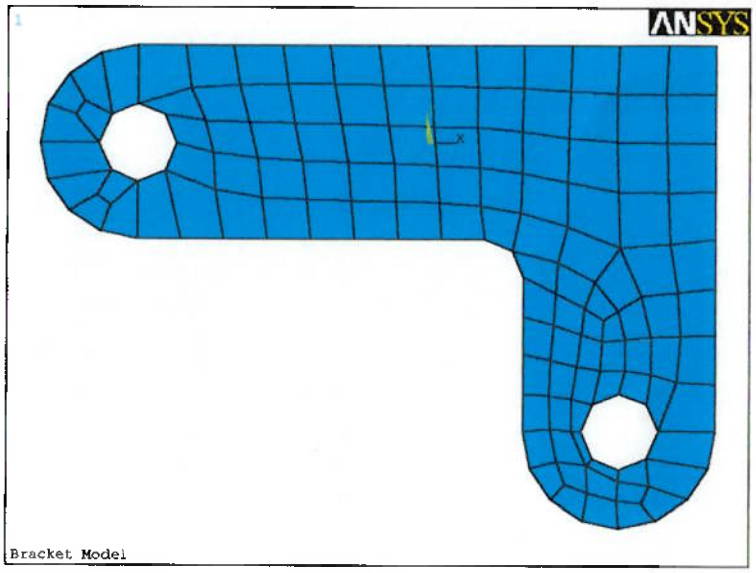

Fig. 11 – Modelo Discretizado.

Depois de aplicadas as condições de contorno (engastamento) e os esforços (força vertical) obtêm-se o resultado a seguir onde se encontra representada a forma deformada da estrutura e a distribuição das tensões na direção do eixo x.

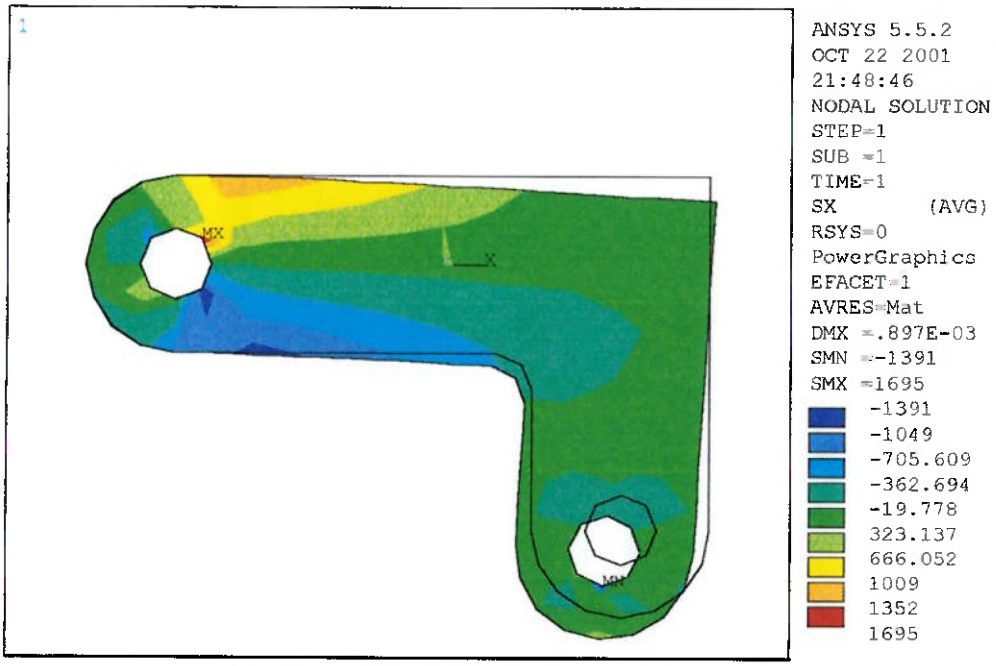

Fig. 12 – Modelo de Área Ansys.

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

4 Medições, Resultados e Análises.

4.1 Procedimentos para medições

Para se efetuar as medições com o interferômetro devem-se verificar os seguintes passos:

- Posicionar o laser, os espelhos, o foto-receptor e a amostra de acordo com a configuração de Michelson;
- Ligar a fonte do laser;
- Alinhar todos os componentes do interferômetro até obter um padrão de interferência consistente:

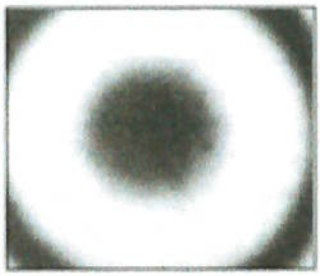

Fig. 13 – Padrão de Interferência.

- Ligar o computador, o gerador de funções, o osciloscópio e o foto-receptor;
- Ajustar o gerador para freqüência e amplitude desejadas;
- Captar os sinais desejados no osciloscópio;
- Transferir os dados do osciloscópio para o PC através da interface GPIB utilizando o software apropriado;
- Tratar os dados no MATLAB;

4.1.1 Calibração

A calibração é um dos passos importantes antes de se efetuar as medições dos sinais propriamente ditos.

O sistema interferométrico montado no laboratório é um sistema muito delicado, sendo que vários fatores podem influenciar nas medidas, como, por exemplo, ruídos do ambiente, iluminação e outros fatores.

Além disto, quando se vai medir deslocamentos de ordem de grandeza menor que a do comprimento de onda do laser, é necessário conhecer qual a voltagem fornecida pelo foto-receptor a cada vez que ocorre o pico de interferência construtiva e o vale de interferência destrutiva quando se desloca a amostra de um comprimento de ordem maior que o comprimento de onda (Cpp, na figura).

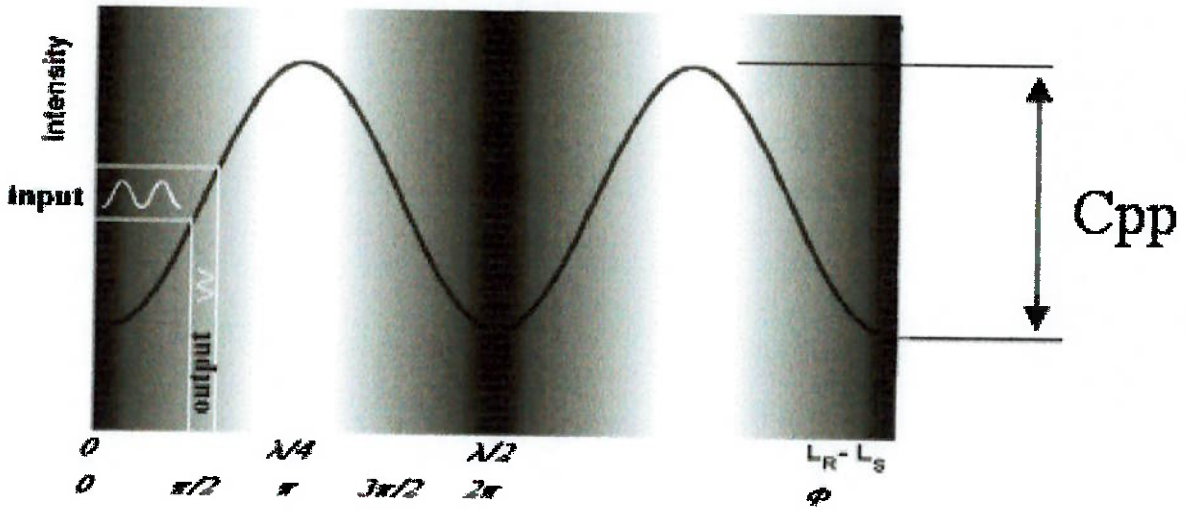

Fig. 14 – Medição de deslocamentos.

Para efetuar a calibração, desliga-se os fios que ligam o gerador de onda ao atuador e move-se o parafuso micrométrico do suporte de forma a movimentar o atuador de deslocamentos maiores que um comprimento de onda.

4.1.2 Medições de Deslocamento

Nas medidas de deslocamento é necessário recalibrar o sistema a cada três ou quatro medidas, de acordo com as condições do ambiente.

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

Uma vez feita a calibração, liga-se novamente os fios do gerador de funções no atuador.

Para obter a saída do foto-receptor é necessário esperar que o sinal no osciloscópio estabilize, devido aos ruídos de baixa freqüência no sistema que influenciam na amplitude do sinal de saída.

Uma vez estabilizado o sinal, obtém-se a saída do foto-receptor e com esse dado e com os dados da calibração calcula-se o deslocamento pico a pico da amostra:

$$D = \frac{Spp}{Cpp} \times \left(\lambda / 4\right)$$

Onde:

- Cpp: Sinal de Calibração
- Spp: Sinal de Atuação
- λ: Comprimento de onda do laser.

4.1.3 Medições de Defasagem

Nessas medições visa-se calcular a defasagem entre a voltagem de entrada no atuador e a voltagem de saída do foto-receptor.

Nas medições de defasagem não é necessária a calibração do sistema, uma vez que os ruídos de baixa freqüência e as condições do ambiente influenciam somente a amplitude do sinal, não influenciando na freqüência e na fase do sinal de saída.

A defasagem é calculada da seguinte forma:

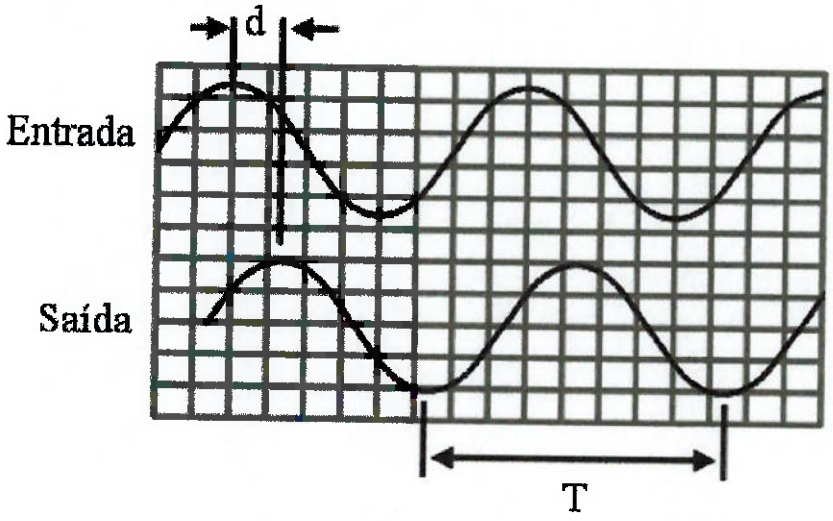

Fig. 15 – Medição de Defasagens.

$$\varphi = \frac{d}{T} \times (360)$$

Esse cálculo pode se tornar muito extenso e mecânico devido à grande quantidade de pontos envolvidos. Para tanto foi gerado um programa no MATLAB de forma a automatizar o processo.

4.1.4 Medições de Impedância

As medições de impedância servem para se observar onde ocorrem as ressonâncias dos atuadores e foram realizadas no laboratório de ultra-som da escola.

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

4.2 Gráficos e Resultados

A seguir um exemplo de sinal obtido pelo osciloscópio:

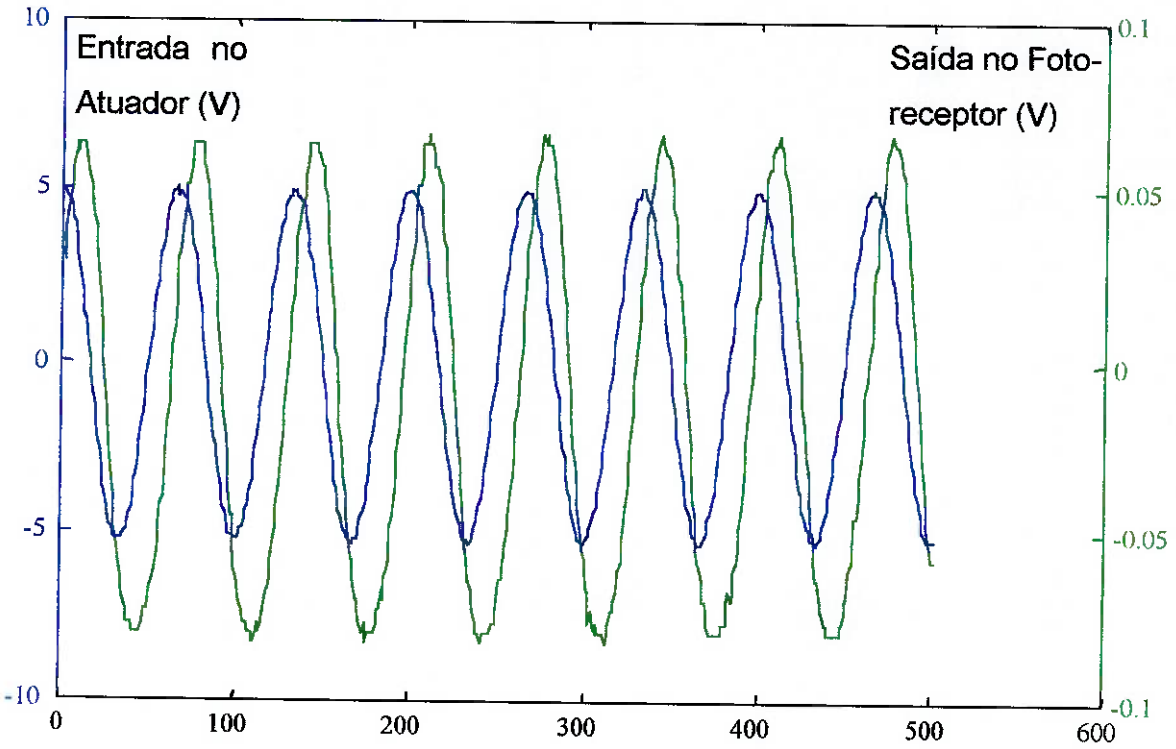

Fig. 16 – Exemplo de sinais captados no osciloscópio.

4.2.1 Deslocamentos

Foram calculados os deslocamentos em função da voltagem para duas freqüências de excitação: 10 e 15 kHz.

Posteriormente calculou-se os deslocamentos em função da freqüência para uma voltagem de excitação de 10 Vpp.

Os gráficos obtidos para os deslocamentos em função da voltagem aplicada para o atuador f1a1025 são os seguintes:

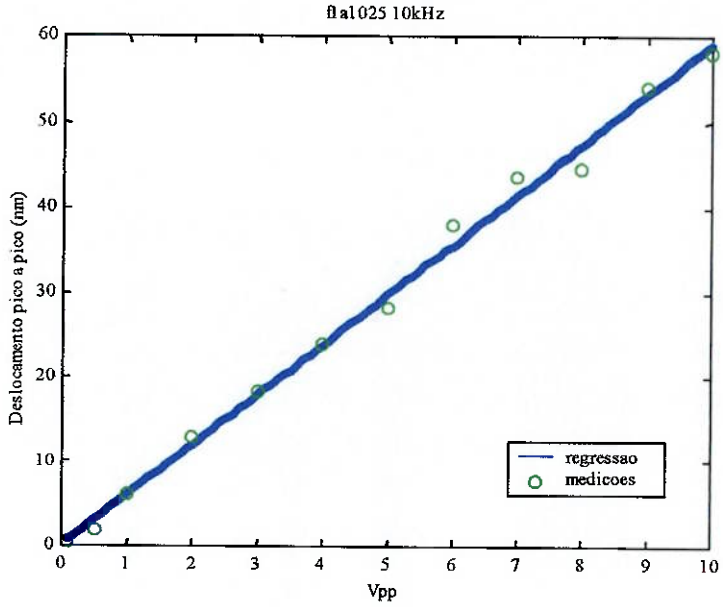

Fig. 17 – Voltagem x Deslocamento a 10 khz.

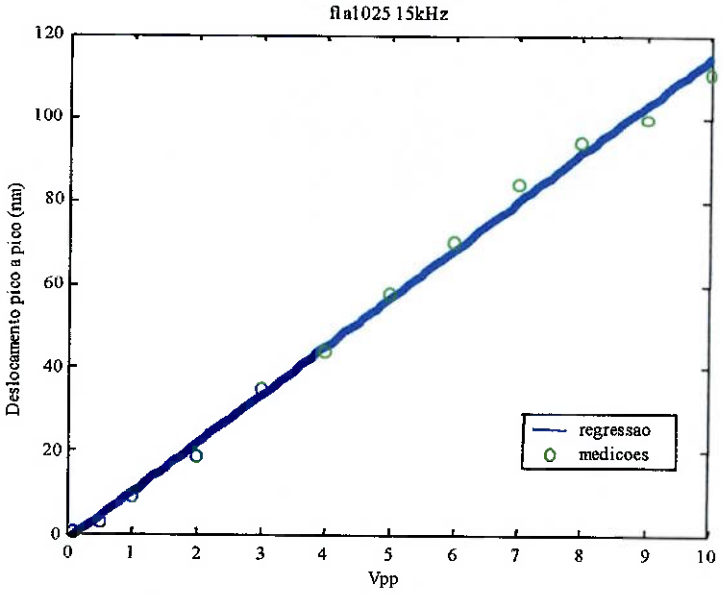

Fig. 18 – Voltagem x Deslocamento a 15 khz.

Percebe-se o comportamento linear do atuador em relação a voltagem de entrada.

4.2.2 Defasagem

Os dados de defasagem medidos para o mesmo atuador f1a1025 estão no gráfico a seguir:

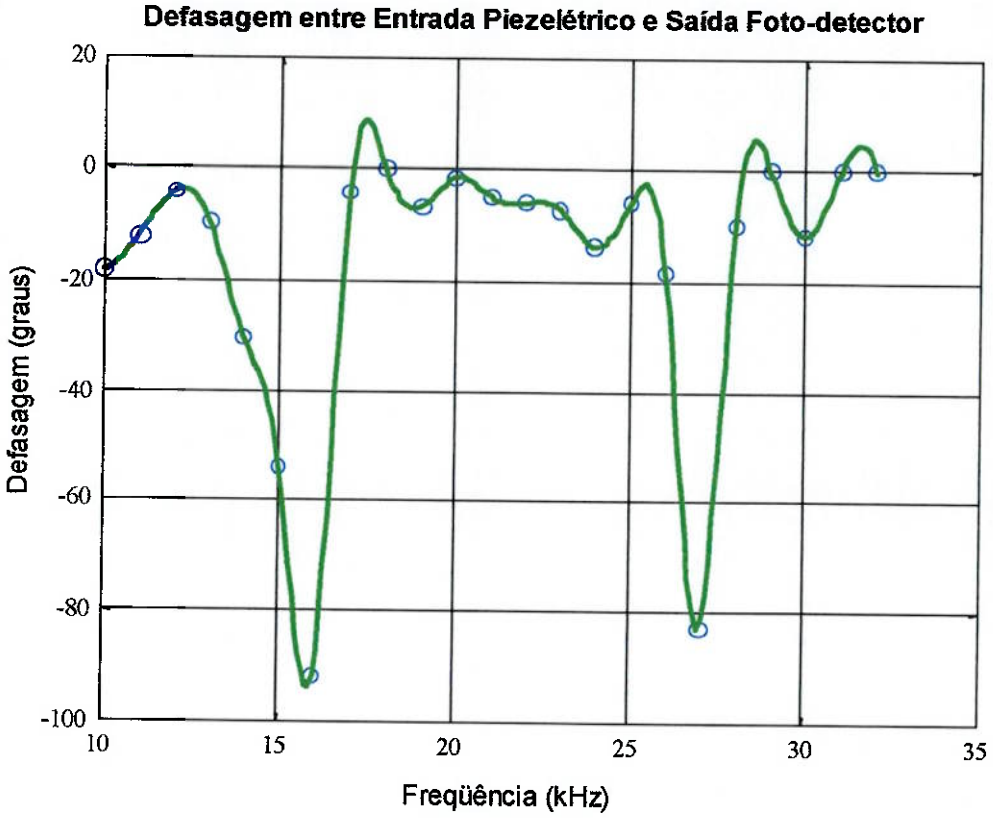

Fig. 19 – Fase entre o sinal de entrada e o de saída.

Assim, experimentalmente obteve-se os seguintes pontos de ressonância:

F_1 = 16,5 khz

F_2 = 27,5 khz

4.2.3 Impedância

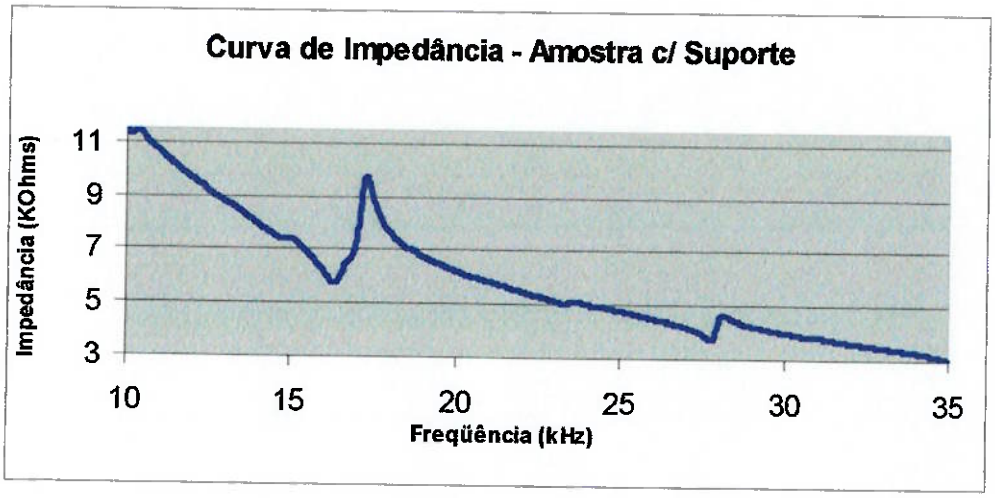

Fig. 20 – Curva de Impedância – Amostra c/ Suporte.

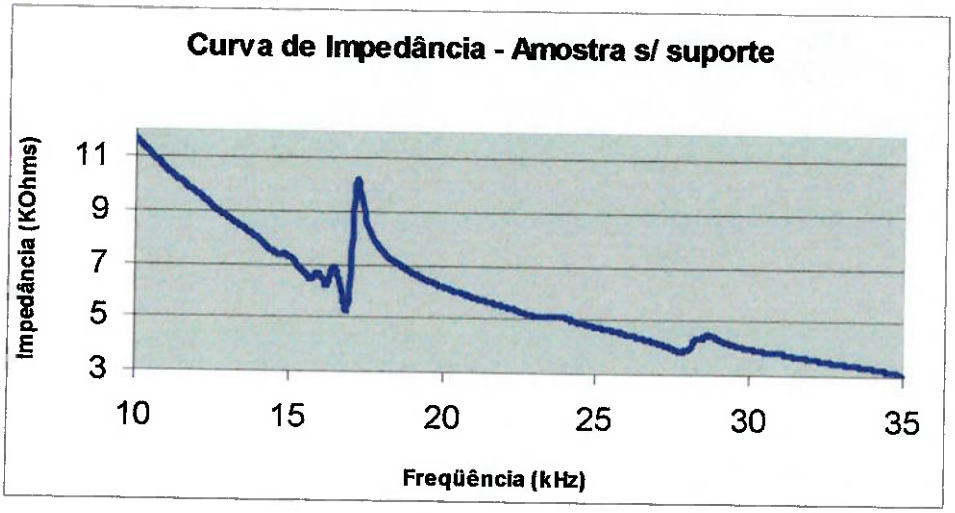

Fig. 21 – Curva de Impedância – Amostra s/ Suporte.

Observa-se uma pequena variação de sinal que depende da fixação do atuador, ou seja, caso se aperte o atuador muito no suporte o sinal será ligeiramente distorcido.

Os picos de ressonância do gráfico de defasagem correspondem aos da curva de impedância.

4.2.4 Comparações com simulações realizadas no ANSYS

Nas simulações realizadas no Ansys foi utilizado um modelo correspondente a ¼ do atuador f1a1025 e está representado na figura:

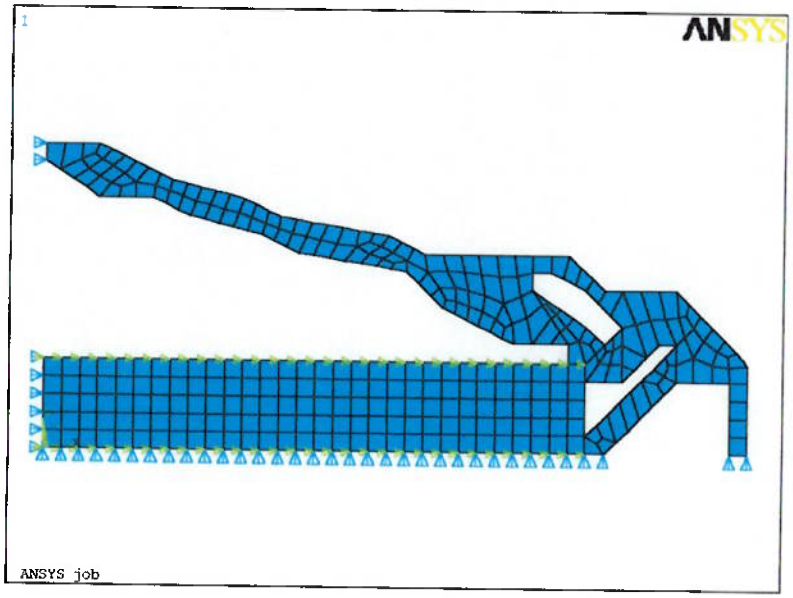

Fig. 22 – Modelo Ansys.

As propriedades da cerâmica piezelétrica e do alumínio utilizadas para as simulações se encontram na tabela a seguir:

PZT-5A	
Constantes elásticas	(10^{10} N.m^{-2})
C_{11}^E	12.00
C_{12}^E	7.51
C_{13}^E	7.52
C_{33}^E	11.10
C_{44}^E	2.10
C_{66}^E	2.24
Constantes piezelétricas	(C.m^{-2})
e_{31}	-5.4
e_{33}	15.8
e_{51}	12.3
Constantes dielétricas	(10^{-9}C^2.N^{-1}.m^{-2})
ε_{11}	8.107
ε_{11}	7.345
Densidade	7800 kg.m^{-3}
Amortecimento	5 x 10^{-8}

Alumínio	
Módulo de Young	71 x 10^9 N.m^{-2}
Densidade	2800 kg.m^{-3}
Amortecimento	4 x 10^{-7}
Coeficiente de Poisson	0.33

Tabela 1: Propriedades dos Materiais PZT-5A e alumínio usadas nas simulações com ANSYS.

Forma deformada do atuador ao se aplicar 10 Vpp:

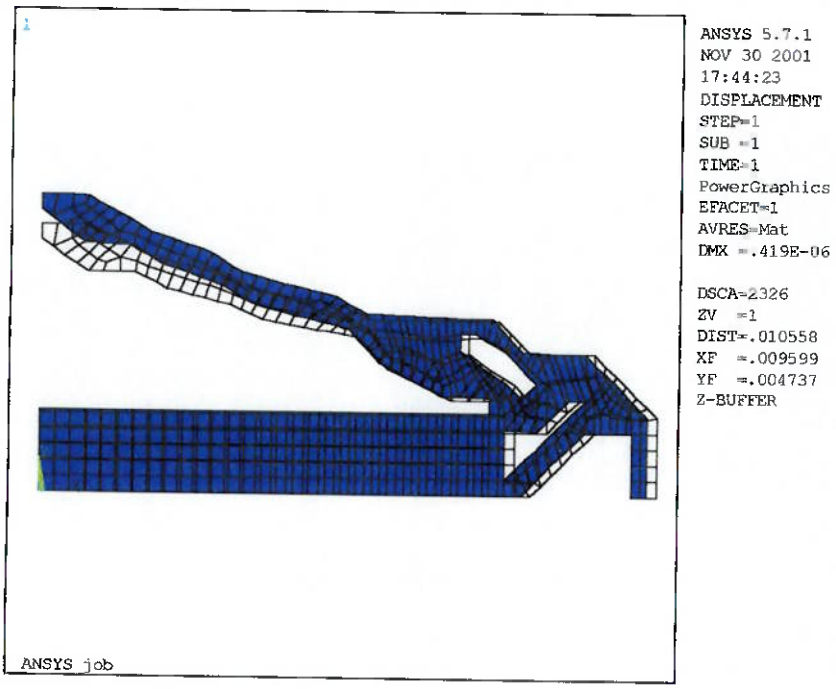

Fig. 23 – Forma deformada – 10 Vpp.

Medidas de deslocamento em frequências de 10 a 50 khz:

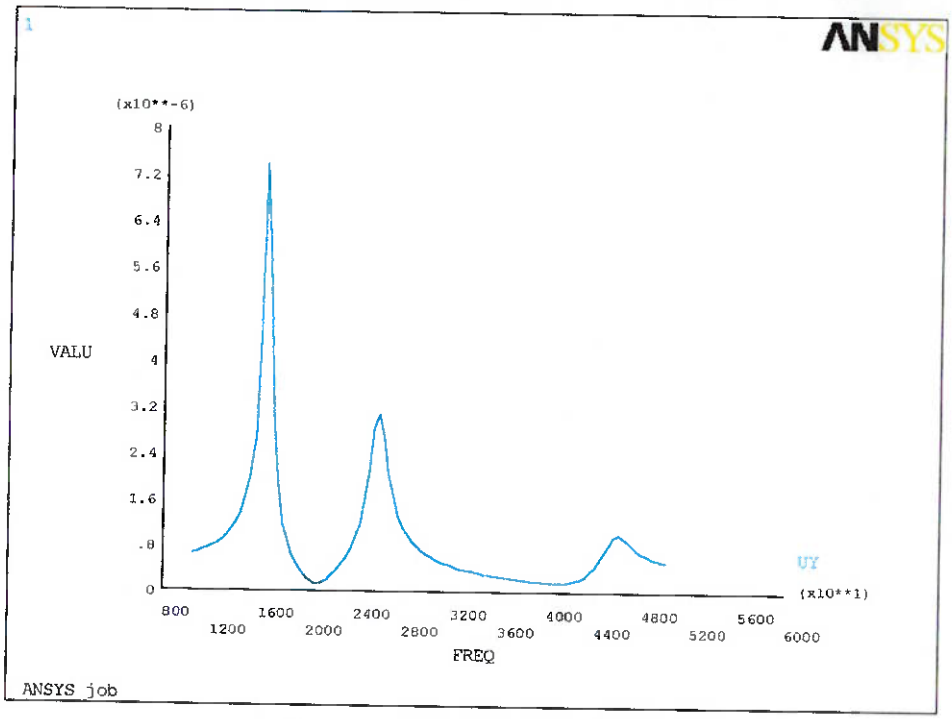

Fig. 24 – Varredura em freqüência.

Comparação com os resultados experimentais:

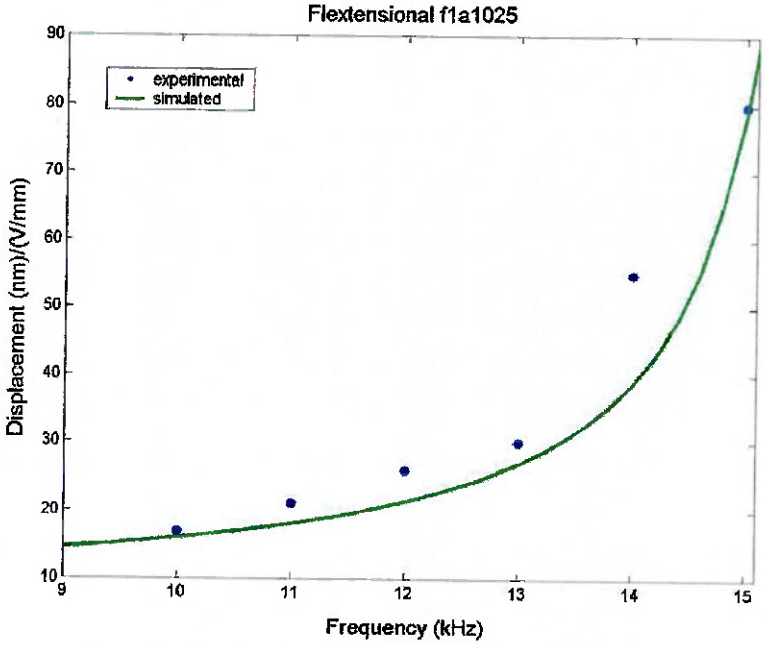

Fig. 25 – Comparação - Freqüência x Deslocamento.

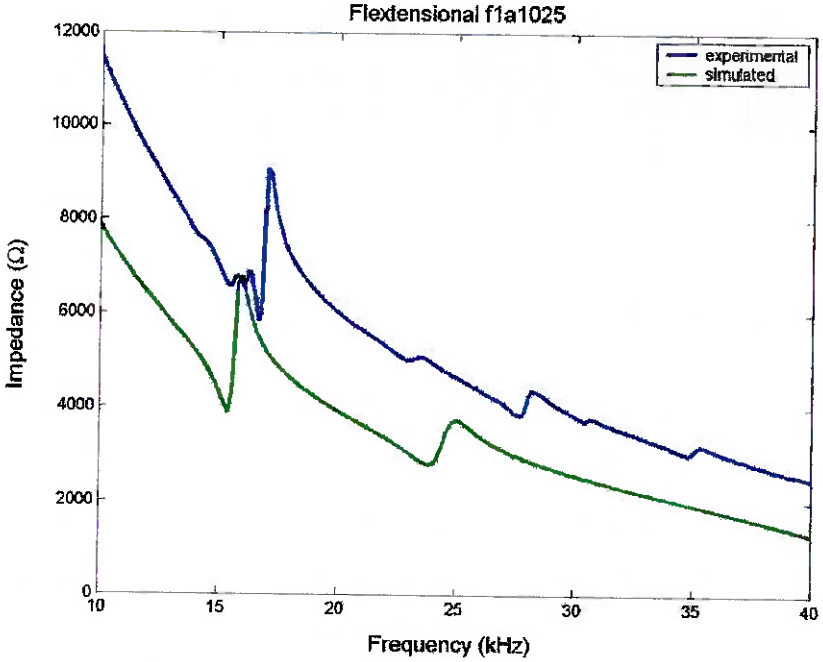

Fig. 26 – Comparação - Freqüência x Impedância.

Como observado há uma pequena discrepância entre os resultados experimentais e os simulados, provavelmente devido às constantes utilizadas para simulação.

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

5 Softwares

5.1 Aquisição de dados do osciloscópio

A aquisição de dados do osciloscópio até o primeiro semestre estava sendo feita através de uma interface GPIB que operava no ambiente DOS.

Assim, como foi descrito no item 4.1, após toda o arranjo e calibração do interferômetro era necessário ir para um ambiente DOS, acessar o programa de operação da GPIB e adquirir o sinal de apenas um canal de cada vez (ou de entrada ou de saída do foto-receptor). O programa gerava um arquivo texto com o sinal adquirido, o qual era aberto pelo MATLAB para começar o tratamento de dados.

Como esse procedimento descrito é extremamente trabalhoso, decidiu-se aperfeiçoá-lo, de forma que fosse necessário apenas um comando de dentro do MATLAB para fazer a aquisição de sinais de vários canais simultaneamente.

Para tanto, foi adquirida uma nova interface GPIB para o computador do laboratório. Com essa nova placa de aquisição tornou-se possível gerar programas em linguagem C que a acessam. Como também é possível gerar programas em C que rodam de dentro do MATLAB, foi possível atingir o objetivo.

Nos itens seguintes há uma descrição mais detalhada para a geração destes programas.

Neste projeto foram desenvolvidos dois programas para possibilitar a comunicação. Um deles foi desenvolvido em Linguagem C (mas acessado de dentro do MATLAB) para possibilitar o acesso à placa e o envio ou recebimento de strings de um instrumento conectado à placa. O segundo foi desenvolvido na linguagem do MATLAB utilizando o primeiro programa para enviar comandos exclusivamente para o osciloscópio e para receber dados do mesmo. Estes programas encontram-se nos anexos I e II respectivamente.

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

5.1.1 Acesso à GPIB

A placa adquirida, Agilent HPIB modelo 82350A, possui bibliotecas de funções para acesso aos instrumentos que estão conectados à mesma. Com essas bibliotecas é possível gerar programas em linguagem C ou Visual Basic que se comunicam com os instrumentos. As bibliotecas disponíveis são: *Agilent VISA (Visual instrument Software Architecture)* e *Agilent SICL (Standard Instrument Control Library).*

No caso de programas em C, deve ser inclusa no início do programa a declaração: *#include "visa.h"* ou *#include "sicl.h"* e ao decorrer do programa utilizam-se as funções de acordo com os manuais de instrução da placa [4].

A seguir um programa simples que se conecta a um instrumento para descobrir a identificação do mesmo:

```
/*id.c
Este programa de exemplo pergunta a um dispositivo conectado à GPIB
por sua string de identificação e imprime o resultado. */

#include <visa.h>
#include <stdio.h>

void main () {
ViSession defaultRM, vi;
char buf [256] = {0};

/* Inicia a seção com o dispositivo no endereço 7 */
viOpenDefaultRM(&defaultRM);
viOpen(defaultRM, "GPIB0::7::INSTR",VI_NULL,VI_NULL, &vi);

/* Envia a string ID? Para o dispositivo */
viPrintf(vi, "ID?\n");

/* Lê o resultado */
viScanf(vi, "%t", buf);

/* Imprime o resultado */
printf("String de identificação: %s\n", buf);
```

```
/* Encerra a seção */
viClose(vi);viClose(defaultRM);}
```

Observações:

- O endereço do dispositivo depende do que está configurado através do software IOConfig que vem com a placa. No caso do laboratório, o osciloscópio é o dispositivo número 7.
- A string que é enviada para o instrumento depende do instrumento utilizado, no item 5.1.3 serão descritas algumas strings pertinentes ao uso com o osciloscópio.

5.1.2 Integração do MATLAB com a linguagem C

É possível criar programas em C que, ao ser utilizados de dentro do MATLAB, se comportam como funções comuns do MATLAB. Estes programas são conhecidos como MEX-Files e podem ser mais bem estudados pela referência [5].

Segue-se um exemplo de um programa em C que utiliza uma variável numérica do MATLAB, a multiplica por 2 e retorna o resultado para o MATLAB.

```
/* biblioteca com funções para manipulação dos dados do MATLAB: */
#include "mex.h"

void timestwo(double y[], double x[]) {
   y[0] = 2 * x[0];
   return; }

/* função principal do programa, correspondente ao main() do C */
void mexFunction(
   int nlhs, mxArray *plhs[],
   int nrhs, const mxArray *prhs[] )
{
   double *y;
   double *x;
   unsigned int m, n;

   /* Checagem do número de argumentos */
```

```
    if (nrhs != 1)
          mexErrMsgTxt("É permitido apenas um argumento de entrada.");
    else if (nlhs != 1)
          mexErrMsgTxt("É permitido apenas um argumento de saída.");

    /* Checagem das dimensões e do timpo do argumento. */
    m = mxGetM(prhs[0]);
    n = mxGetN(prhs[0]);
    if ( !mxIsDouble(prhs[0]) || mxIsComplex(prhs[0]) || !(m==1&&n==1) )
          mexErrMsgTxt("A entrada x deve ser um escalar ");

    /* Cria a matrix do argumento de saída. */
    plhs[0] = mxCreateDoubleMatrix(m, n, mxREAL);

    /* Designa um ponteiro para cada entrada e saída.
    y = mxGetPr(plhs[0]);
    x = mxGetPr(prhs[0]);

    /* Chama a rotina para multiplicação. */
    timestwo(y, x);
}
```

Para executar o programa anterior no MATLAB deve-ser digitar no prompt:
» y = timestwo(x)

5.1.3 Comunicação com o osciloscópio

Para se comunicar com o osciloscópio é necessário enviar strings através da GPIB. Por exemplo, caso se deseje não mostrar o canal 2, deve-se enviar a seguinte string: "blank channel 2". Todos os comandos que o osciloscópio reconhece podem ser encontrados no manual de programação [6].

A seguir alguns dos comandos pertinentes a este projeto:

- digitize channel 1 | 2 | 3 | 4

Armazena as informações dos canais informados na memória.

- waveform source memory 5 | 6 | 7 | 8

  Define a memória da qual os dados serão transferidos.

- waveform format ascii

  Define o formato de transferência de dados.

- waveform data?

  Adquire os pontos referentes à memória informada.

- local

  Coloca o osciloscópio de volta à operação local.

- eoi on

  Informa que o osciloscópio deve enviar um caractere de termino de linha após cada dado enviado.

- acquire type average

  Informa que o tipo de aquisições deve ser com médias.

5.2 Cálculo de defasagem no MATLAB

Como descrito no item 4.1.4 o cálculo da defasagem entre o sinal de entrada e saída é bastante trabalhoso, para tanto foi desenvolvido um programa no MATLAB que utiliza os vetores com os dados retornados pelo programa descrito no anexo II e faz automaticamente o cálculo da defasagem. Este programa se encontra descrito a seguir:

<phase.m>

```
function fase = phase(a,b)
% Cálculo da diferença de fase entre dois sinais senoidais a e b
%
% Sintaxe:
% Fase = phase(a,b)
%

%Normalização

a = -1 + 2*(a-min(a))/(max(a)-min(a));
b = -1 + 2*(b-min(b))/(max(b)-min(b));

%Diferenciação

aa=diff(a);
aa = -1 + 2*(aa-min(aa))/(max(aa)-min(aa));
bb=diff(b);
bb = -1 + 2*(bb-min(bb))/(max(bb)-min(bb));


for I=1:length(aa),
   c(I)=atan2((b(I+1)+b(I))/2,bb(I))-atan2((a(I+1)+a(I))/2,aa(I));
end

Fase_=sort(c);
fase = mean(Fase_)*180/pi;
```

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

6 Conclusão

Durante a primeira parte do trabalho encontrou-se algumas dificuldades quanto à estabilização do Interferômetro e quanto a ruídos de baixa freqüência nos cálculos dos deslocamentos. Estes problemas tornam a medição muito onerosa, sendo necessário dispender um grande tempo em cada medição. De forma a corrigir este problema está sendo desenvolvido pelo colaborador *Doutorando Gilder Nader* um atuador realimentado onde será posicionado o espelho de referência, anulando-se os ruídos de baixa freqüência.

No cálculo de defasagens a dificuldade encontrada foi devido à quantidade de cálculos a serem feitas para se obter a defasagem, para tanto se desenvolveu um programa em MATLAB destinado a melhorar o processo.

Ainda, como toda a aquisição em si estava onerosa, aperfeiçoou-se o processo de aquisição através da geração de uma função que pode ser utilizada de dentro do MATLAB para efetuar a aquisição dos dados. Este programa desenvolvido é útil não só neste trabalho, mas também em qualquer outro que envolva uma placa de aquisição de dados GPIB.

7 Referências Bibliográficas

[1] Hecht, Eugene, *Optics*, Adelphi University, Addison-Wesley Publishing Company.

[2] Halliday and Resnick, *Física IV,* Livros técnicos e Científicos.

[3] Tolansky, S., *An Introduction to Interferometry*, Longman, Green and Co. Ltd.

[4] Agilent Technologies, Inc, *Agilent Technologies VISA User's Guide*, Edition 4.

[5] The Mathworks, Inc, *Application Program Interface Guide*, Version 5.

[6] Hewlett-Packard S/A, *HP54112 Programing Reference Manual.*

[7] http://www.ifi.unicamp.br/~accosta/nota3.htm, site desenvolvido pelo Prof. Dr. A. C . Costa, docente do instituto de física da Unicamp.

[8] http://www.mcca.ep.usp.br/~lus, site do Laboratório de Sensores e Atuadores da Escola Politécnica da USP.

Sites introdutórios sobre MEF:

- http://www3.sympatico.ca/peter_budgell/FEA_intro.html
- http://www3.sympatico.ca/peter_budgell/ANSYS_tips.html
- http://students.fct.unl.pt/users/masr/
- http://www.pce.com.br/intro_fea.htm
- http://www.ime.eb.br/~savi/MEF_Apresentacao

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

ANEXO I

Neste anexo encontra-se a rotina desenvolvida em C integrada com o MATLAB para enviar e receber dados de uma interface GPIB.

Após compilado, utilizando o compilador do MATLAB ou o Visual C, é gerado o arquivo hpib.dll que deve ser colocado no diretório de trabalho do MATLAB. A utilização deste arquivo de dentro do MATLAB é a seguinte:

» [A, B] = hpib(*'operação'*, *'comando'*, *'s'*)

Onde:

'operação' deve ser *'send'* caso se deseje *enviar* comandos para o instrumento, ou *'get'* caso se deseje *capturar* dados do instrumento.

'comando' deve conter o comando que se deseja enviar (por ex, 'blank channel 2') ou o dado que se deseja capturar (por ex, 'yref?').

O terceiro argumento: *'s'* somente deve ser informado caso se deseje que o programa retorne caracteres e não números ao MATLAB, ou então quando a resposta do osciloscópio contiver caracteres, nos demais casos deve ser omitido.

A seguir o arquivo hpib.c que contém o código fonte do programa:

```
#include "mex.h"      // Funções para tratamento das variáveis do MATLAB
#include "visa.h"     // Funções para comunicação com a GPIB
#include <string.h>   // Funções para tratamento de strings

#define ARRAY_LENGTH 501    // tamanho do vetor retornado ao MATLAB

void mexFunction(
   int nlhs, mxArray *plhs[],
   int nrhs, const mxArray *prhs[] )
```


```
{
   char *operation, *command; // string para o primeiro e segundo argumento
   unsigned int length;       // armazena o tamanho de uma string
   int status;
   ViSession defaultRM, vi;   // para comunicação com a GPIB

   int doloop;
   int i,j;
   double *data;
   double *res;
   char **sdata;

   // Tamanho da string operation
   length = (mxGetM(prhs[0]) * mxGetN(prhs[0])) + 1;

   // Alocação de memória para a string operation
   operation = mxCalloc(length, sizeof(char));
   status = mxGetString(prhs[0], operation, length);
   if (status)
         mexErrMsgTxt("Could not convert operation string data.");

   /* Abre uma sessão de comunicação com a GPIB no endereço 7 */
   viOpenDefaultRM (&defaultRM);
   viOpen (defaultRM, "GPIB0::7::INSTR", VI_NULL,VI_NULL, &vi);

   // Captura da string command a ser enviada para o dispositivo
   if (nrhs > 1) {
         length = (mxGetM(prhs[1]) * mxGetN(prhs[1])) + 1;

         // Alocação de memória para a string command
         command = mxCalloc(length, sizeof(char));
         status = mxGetString(prhs[1], command, length);
         if (status)
               mexErrMsgTxt("Could not convert command string data.");
   }

   // Operação de envio de dados para o osciloscópio
   if(!strcmp(operation, "send"))
         viPrintf (vi, "%s\n",command);
```

```
// Operação de recebimento de dados do osciloscópio
else if (!strcmp(operation, "get"))
{
	viPrintf (vi, "%s\n",command);

	/* Caso sejam apenas dois os argumentos do comando, os dados serão
	enviados como números */
	if (nrhs==2)
	{
		data = mxCalloc(ARRAY_LENGTH, sizeof(double));

		i = 0;
		doloop = 1;
		do {
			viScanf (vi, "%lf", &data[i]);
			/* O osciloscópio envia o numero 1.000e38 quando não há
			dados para serem enviados */
			if ( data[i]>=100000000000 )
				doloop = 0;
			else
				i++;
		} while (doloop && i<ARRAY_LENGTH);

		if (!i)
			mexErrMsgTxt("The device returned no data!");
		else
		{
			plhs[0] = mxCreateDoubleMatrix(1, i, mxREAL);
			res = mxGetPr(plhs[0]);
			for(j=0; j<i; j++)
				*(res+j) = data[j];
		}
	}

	// Caso sejam três os argumentos do comando, os dados serão enviados como
	caracteres.
	else if(nrhs==3 )
	{
		sdata = mxCalloc(ARRAY_LENGTH, sizeof(char*));
		for (i=0;i<ARRAY_LENGTH;i++)
```


```
                sdata[i] = mxCalloc(20, sizeof(char));

            i = 0;
            doloop = 1;
            do {
                viScanf (vi, "%s", sdata[i]);
                if (!strcmp(sdata[i],"1.00000E+38"))
                    doloop=0;
                else if (strcmp(sdata[i],""))
                    i++;
            } while (doloop && i<501);

            if (!i)
                mexErrMsgTxt("The device returned no data!");
            else
                plhs[0] = mxCreateCharMatrixFromStrings(i, sdata);
        }
    }
}
```

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos

ANEXO II

Neste anexo encontra-se a rotina desenvolvida em linguagem de programação do MATLAB para acesso aos dados do osciloscópio utilizando-se o programa descrito no anexo anterior.

O arquivo osc.m deve ser colocado no diretório de trabalho do MATLAB juntamente com o arquivo hpib.dll.

A sintaxe de utilização de dentro do MATLAB está descrita a seguir:

» [A, B] = osc(CanalA, CanalB);

Onde:

- CanalA, CanalB => Número dos canais a serem capturados: 1, 2, 3 ou 4
- A, B => Vetores com os canais capturados.

Caso esteja havendo problemas ao receber os dados, por ex. tudo zerado,deve ser utilizado o seguinte comando para inicializar o osciloscópio:

» osc('init');

A seguir o arquivo osc.m que contém o código fonte do programa:

```
function [chanA, chanB] = osc(A,B)
% Captura os sinais de dois canais simultaneamente do osciloscopio
%  pela HPIB
%
% Sintaxe:
% [A, B] = osc(CanalA, CanalB);
%
% onde:
% CanalA, CanalB  ->  Numero dos canais a serem capturados: 1, 2, 3 ou 4
```

```
% A, B            ->  Vetores com os canais capturados
%
%
% Caso esteja havendo problemas ao receber os dados, por ex. tudo zerado,
%  utilize o seguinte comando para inicializar o osciloscopio:
%
% osc('init');
%
% OBS: É necessário que o arquivo hpib.dll esteja no mesmo diretório que o osc.m
% ou disponível no PATH do matlab.

if A == 'init'
   hpib('send','eoi on')
   hpib('send','acq type average reso off')
   hpib('send','head off')
else

% Digitalizacao dos canais desejados.
hpib('send',['digitize channel ' num2str(A) ', ' num2str(B)])

% visualizacao da captura
% O canal 1 e capturado na memoria 5, o 2 na 6 ...
% 1 -> 5
% 2 -> 6
% 3 -> 7
% 4 -> 8
memA = A + 4;
memB = B + 4;
%hpib('send',['view mem' num2str(memA) ' view mem' num2str(memB)])

% Captura dos dados da memoria A
hpib('send',['wav src mem ' num2str(memA)])
yref = hpib('get','yref?'); yref = yref(1,1);
yinc = hpib('get','yinc?'); yinc = yinc(1,1);
yorg = hpib('get','yorg?'); yorg = yorg(1,1);
hpib('send','wav form ascii')
chanA = hpib('get','data?');

% Normalizacao dos dados capturados
chanA = (chanA - yref)*yinc + yorg;
```


```
% Captura dos dados referente ao tempo
xref = hpib('get','xref?'); xref = xref(1,1);
xinc = hpib('get','xinc?'); xinc = xinc(1,1);
xorg = hpib('get','xorg?'); xorg = xorg(1,1);
t=[1:501];
t = (t - xref)*xinc + xorg;

% Captura dos dados da memoria B
hpib('send',['wav src mem ' num2str(memB)])
yref = hpib('get','yref?'); yref = yref(1,1);
yinc = hpib('get','yinc?'); yinc = yinc(1,1);
yorg = hpib('get','yorg?'); yorg = yorg(1,1);
hpib('send','wav form ascii')
chanB = hpib('get','data?');

% Normalizacao dos dados capturados
chanB = (chanB - yref)*yinc + yorg;

% Plotagem dos resultados
subplot(2,2,1)
plot(t,chanA)
subplot(2,2,2)
plot(t,chanB)
subplot(2,2,3)
plot(t,chanA,t,chanB)
subplot(2,2,4)
plot(chanA,chanB)


end

% Retorno do osciloscopio a operacao local
hpib('send','local')
```