

Note final
9,7 (nove e sete)

FLÁVIO HONDA
WAGNER SHIN NISHITANI

WAN
22/01/04

Simulação, Fabricação e Comando de MEMS Eletrotermomecânicos

Trabalho de Formatura apresentado à
Escola Politécnica da Universidade de
São Paulo para obtenção do título de
Bacharel em Engenharia

São Paulo
2003

**FLÁVIO HONDA
WAGNER SHIN NISHITANI**

**Simulação, Fabricação e Comando de MEMS
Eletrotermomecânicos**

Trabalho de Formatura apresentado à
Escola Politécnica da Universidade de
São Paulo para obtenção do título de
Bacharel em Engenharia

Área de Concentração:
Engenharia Mecatrônica

Orientador:
Prof. Dr. Emílio Carlos Nelli Silva

São Paulo
2003

Aos nossos pais, que têm investido esforços sem medida para
nossa formação.

AGRADECIMENTOS

A Deus, por nos permitir chegar até aqui.
A nossos pais, pelo apoio durante todo o período acadêmico.
Ao professor Emílio, pela contínua supervisão e encorajamento.
Ao Paulo, por compartilhar parte do seu tempo e de sua pesquisa em nosso favor.
Ao Laboratório Nacional de Luz Síncrotron (LNLS), por disponibilizar suas instalações e equipamentos.

RESUMO

O presente trabalho objetivou o estudo do ciclo completo de desenvolvimento de um MEMS (“Micro-Electro-Mechanical Systems”) eletrotermomecânico, incluindo a sua simulação por método dos elementos finitos e o desenvolvimento de seu sistema de comando de posicionamento, feito através de uma fonte de tensão controlada remotamente. Um MEMS é um mecanismo de acionamento elétrico que permite a manipulação de pequenos objetos, sejam eles mecânicos ou de origem biológica (células, por exemplo), possibilitando desde a teleoperação à automatização completa do funcionamento. Seus elementos mecânicos e atuadores são fabricados por meio de tecnologia de microfabricação. Em sua quase totalidade são mecanismos flexíveis, onde o movimento é dado pela flexibilidade da própria estrutura ao invés de serem usados pinos e juntas, muito difíceis de serem montados nessa escala. A atuação eletrotermomecânica é baseada na expansão térmica da estrutura resultante do calor gerado pela passagem de uma corrente elétrica. Foram feitas simulações dos MEMS eletrotermomecânicos pelo Método dos Elementos Finitos, utilizando um “software” comercial. A fabricação se deu utilizando um método de microfabricação de superfície (fotolitografia) e um comando por “joystick” foi implementado com sucesso.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LISTA DE SÍMBOLOS

1. INTRODUÇÃO	1
2. REVISÃO BIBLIOGRÁFICA.....	2
2.1. Metodologia de Projeto de MEMS	2
2.2. Tipos de Atuadores em Micromanipulação	4
2.3. O Atuador Eletrotermomecânico	7
2.4. Simulação em Método de Elementos Finitos.....	11
2.4.1. Fundamentos Teóricos	11
2.5. Fabricação de MEMS	14
2.6. Estratégia de Comando	24
2.6.1. Visão Geral	24
2.6.2. Interface entre o MATLAB e o “joystick”.....	27
2.6.2.1. “Data Acquisition Toolbox”.....	27
2.6.3. Interface entre o MATLAB e a fonte DC	31
2.6.3.1. A INTERFACE GPIB	31
2.6.3.2. “INSTRUMENT CONTROL TOOLBOX”.....	36
2.6.4. Interface com o usuário.....	46
3. MATERIAIS E MÉTODOS	50
3.1. Simulações de MEMS.....	50
3.1.1. Obtenção do Modelo para Simulação	50
3.1.2. Garra (projeto otimizado).....	51
3.1.3. Mesa XY (projeto intuitivo).....	53
3.2. Comando de MEMS	56
3.2.1. Construção de um Adaptador de “Joystick”	56
3.2.2. Algoritmo do “software” de comando	60
3.2.3. Construção da interface com o usuário	63
3.3. Experimento com a Mesa XY.....	64

3.4. Simulação Revisada da Mesa XY	67
4. RESULTADOS.....	69
4.1. Simulações de MEMS.....	69
4.1.1. Garra (projeto otimizado).....	69
4.1.2. Mesa XY (projeto intuitivo).....	73
4.2. Interface do Comando Remoto da Fonte DC.....	76
4.3. Experimento com a Mesa XY	81
4.4. Simulação Revisada da Mesa XY	84
5. DISCUSSÃO.....	88
5.1. Simulações de MEMS.....	88
5.1.1. Garra (projeto otimizado).....	88
5.1.2. Mesa XY (projeto intuitivo).....	89
5.2. Interface do Comando Remoto da Fonte DC.....	90
5.3. Experimento com a Mesa XY	91
5.4. Simulação Revisada da Mesa XY	93
6. CONCLUSÕES	94
7. ANEXO A – Código-Fonte do “software” de Comando em MATLAB.....	95
8. LISTA DE REFERÊNCIAS	100

LISTA DE FIGURAS

Fig. 2.1 - Atuador eletrotermomecânico simples.....	7
Fig. 2.2 - Atuador eletrotermomecânico ativo.....	8
Fig. 2.3 – Deposição de resiste	20
Fig. 2.4 – Processo de Fotolitografia	22
Fig. 2.5 – Visão geral do comando remoto da fonte DC	24
Fig. 2.6 – Interfaces de “hardware” e “software”	25
Fig. 2.7 – Conector de 24 vias e suas correspondentes linhas	32
Fig. 2.8 – Processo de transferência do protocolo “handshaking”	34
Fig. 2.9 – Fluxo de informações entre o MATLAB e o instrumento	38
Fig. 2.10 – “Layout Editor”	46
Fig. 2.11 – Inspetor de Propriedades	49
Fig. 3.1 - Perfil da garra obtida por projeto otimizado	51
Fig. 3.2 - Desenho CAD Bidimensional da Mesa XY	53
Fig. 3.3 - Modelo tridimensional da Mesa XY em ANSYS	54
Fig. 3.4 – Algoritmo do software de comando do microatuador	61
Fig. 3.5 – Objetos da interface gráfica desenvolvida.....	63
Fig. 3.6 – Microscópio com Câmera CCD acoplada	64
Fig. 3.7 - Substrato fixado ao microscópio e com as pontas de prova posicionadas .	65
Fig. 3.8 - Equipamento usado no comando da fonte DC por “joystick”	66
Fig. 3.9 - Modelo Revisado da Mesa XY em ANSYS	68
Fig. 3.10 - Modelo Revisado da Mesa XY em ANSYS (detalhe de um dos terminais)	68
Fig. 4.1 - Simulação da Garra: Deslocamento Total (tensão em U_x).....	69
Fig. 4.2 - Simulação da Garra: Distribuição de Temperatura (tensão em U_x).....	70
Fig. 4.3 - Simulação da Garra: Deslocamento Total (tensão em U_y).....	70
Fig. 4.4 - Simulação da Garra: Distribuição de Temperatura (tensão em U_y).....	71
Fig. 4.5 - Simulação da Garra: Deslocamento Total (tensão em U_g).....	71
Fig. 4.6 - Simulação da Garra: Distribuição de Temperatura (tensão em U_g).....	72
Fig. 4.7 - Simulação da Mesa XY: Distribuição de Tensão Elétrica	73
Fig. 4.8 - Simulação da Mesa XY: Densidade de Corrente.....	74
Fig. 4.9 - Simulação da Mesa XY: Distribuição de Temperatura.....	74

Fig. 4.10 - Simulação da Mesa XY: Deslocamento.....	75
Fig. 4.11 - Simulação da Mesa XY: Tensão Mecânica (Von Mises)	75
Fig. 4.12 – Interface com o usuário	76
Fig. 4.13 – Configurações da interface GPIB	77
Fig. 4.14 – Seleção do TIPO DE COMANDO e TENSÃO MÁXIMA	78
Fig. 4.15 – Mensagem de erro: Tensão Máxima Excedida	78
Fig. 4.16 – Tela de operação via “joystick”	79
Fig. 4.17 – Mensagem de erro: Operação via “joystick”	79
Fig. 4.18 – Confirmação de saída	80
Fig. 4.19 - Mecanismo da Mesa XY fabricado.....	81
Fig. 4.20 - Substrato de alumina que contém a Mesa XY fabricada	81
Fig. 4.21 - Equipamento com usuário operando o comando por “joystick”.....	83
Fig. 4.22 - Extremidade da Mesa XY cuja posição era controlada por “joystick”....	83
Fig. 4.23 - Simulação Revisada da Mesa XY: Distribuição de Tensão Elétrica	84
Fig. 4.24 - Simulação Revisada da Mesa XY: Densidade de Corrente	85
Fig. 4.25 - Simulação Revisada da Mesa XY: Distribuição de Temperatura	85
Fig. 4.26 - Simulação Revisada da Mesa XY: Deslocamento	86
Fig. 4.27 - Simulação Revisada da Mesa XY: Deslocamento (detalhe).....	86
Fig. 4.28 - Simulação Revisada da Mesa XY: Tensão Mecânica (Von Mises).....	87

LISTA DE TABELAS

Tab. 2.1 – Comparação entre as técnicas de deposição	17
Tab. 2.2 – Código-fonte do Adaptador de Demonstração	29
Tab. 2.3 – Adaptadores do Data Acquisition Toolbox	30
Tab. 2.4 – Propriedades configuráveis da interface GPIB.....	39
Tab. 2.5 – Propriedades específicas da interface GPIB.....	39
Tab. 2.6 – Adaptadores de Interface que o MATLAB suporta.....	40
Tab. 2.7 – Funções de escrita de dados.....	44
Tab. 2.8 – Funções de leitura de dados.....	45
Tab. 3.1 - Propriedades do níquel eletrodepositado (garra).....	52
Tab. 3.2 - Algumas propriedades do níquel eletrodepositado (mesa XY).....	55
Tab. 3.3 - Temperatura e resistividade elétrica do níquel eletrodepositado (mesa XY)	
.....	55
Tab. 3.4 - Temperatura e condutividade térmica do níquel eletrodepositado (mesa XY).....	55
Tab. 3.5 – Modificação dos nomes dos arquivos.....	56
Tab. 4.1 - Alguns resultados da simulação (garra)	72
Tab. 4.2 - Resultados experimentais (mesa XY)	82

LISTA DE ABREVIATURAS E SIGLAS

ANSYS	- Programa comercial
API	- Application Program Interface
ATN	- Attention
CAD	- Computer Aided Design
CCD	- Charge Coupled Device
CR	- Carriage Return
CVD	- Chemical vapor Deposition
DAV	- Data Valid
DC	- Direct Current
DUV	- Deep Ultraviolet
EOI	- End or Identify
EOS	- End of String
ETM	- Eletrotermomecânico
Fig.	- Figura
GPIB	- General-Purpose Interface Bus
GUI	- Graphical User Interface
GUIDE	- Graphical User Interface Development Environment
i.e.	- do latim “id est” (“isto é”)
IEEE	- Institute of Electrical and Electronics Engineers
IFC	- Interface Clear
LF	- Line feed
LNLS	- Laboratório Nacional de Luz Síncrotron
MATLAB	- Programa comercial
MEF	- Método de Elementos Finitos
MEMS	- Micro-Electro-Mechanical Systems
NDAC	- Not Data Accepted
NRFD	- Not Ready for Data
NTSC	- National Television Systems Committee
PCI	- Peripheral Component Interconnect
PR	- Photoresist
REN	- Remote Enable

SCPI	- Standard Commands for Programmable Instruments
SDK	- Software Development Kit
SMA	- Shape Memory Alloy
SP	- São Paulo
SRQ	- Service Request
Tab.	- Tabela
TCP/IP	- Transmission Control Protocol/Internet Protocol
UDP	- User Datagram Protocol
UUID	- Universal Unique Identification
UV	- Ultravioleta
VISA	- Virtual Instrumentation Software Architecture

LISTA DE SÍMBOLOS

α	- Coeficiente de expansão térmica
ΔT	- Diferença de temperatura
ε	- Matriz de deformações (MEF)
Γ	- Fronteira da região sob investigação (MEF)
Γ_{eE}	- Fronteira da região sob investigação com tensão elétrica conhecida (MEF)
Γ_{eT}	- Fronteira da região sob investigação com temperatura conhecida (MEF)
Γ_{eU}	- Fronteira da região sob investigação com deslocamento conhecido (MEF)
Γ_{nT}	- Fronteira da região sob investigação com temperatura desconhecida (MEF)
ρ	- Resistividade elétrica
σ	- Matriz de tensões (MEF)
Ω	- Área da região sob investigação (MEF)
A	- Área da seção transversal do atuador ETM simples
d_f	- Deslocamento livre (sem carga) do atuador ETM simples
E	- Módulo de elasticidade
f_{nT}	- Fluxo de calor na fronteira dependente da temperatura (MEF)
F_b	- Força máxima (blocking force) do atuador ETM simples
H	- Transporte de calor para fora do atuador ETM simples
I	- Corrente elétrica
I	- Matriz identidade
k_e	- Condutividade elétrica
k_t	- Condutividade térmica
l	- Comprimento
n	- Vensor normal à superfície
P	- Potência elétrica
R	- Resistência elétrica
t	- Espessura
T	- Temperatura
T_∞	- Temperatura ambiente
T_{espec}	- Temperatura conhecida, aplicada a uma área particular (MEF)

- T_{ij} - Terminal para aterramento elétrico (i sendo 1 ou 2, j sendo X ou Y) (mesa XY)
- u - Deslocamento (MEF)
- u_{espec} - Deslocamento conhecido, aplicado a uma área particular (MEF)
- U - Tensão elétrica
- U_x - Terminal para aplicação de tensão elétrica para mover em X (garra)
- U_y - Terminal para aplicação de tensão elétrica para mover em Y (garra)
- U_g - Terminal para aplicação de tensão elétrica para fechar garra (garra)
- V - Tensão elétrica
- V_{espec} - Tensão elétrica conhecida, aplicada a uma área particular (MEF)
- V_{ij} - Terminal para aplicação de tensão elétrica (i sendo 1 ou 2, j sendo X ou Y) (mesa XY)
- w - Largura
- X - Eixo ou Direção X
- Y - Eixo ou Direção Y
- Z - Eixo ou Direção Z

1. INTRODUÇÃO

A pesquisa em torno da manipulação de pequenos objetos, sejam eles mecânicos ou de origem biológica (células, por exemplo), tem sua importância cada vez mais evidenciada pela tendência mundial em se miniaturizar sistemas. A principal aplicação direta seria em um amplo espectro, variando da montagem de sistemas mecânicos de tamanho reduzido, a microcirurgias e pesquisas em biotecnologia. Para seu uso em larga escala, é necessária uma elevada precisão mesmo sob altas velocidades no posicionamento do dispositivo.

Os micromecanismos que possibilitam desde a teleoperação à automatização completa do funcionamento são os denominados MEMS (“Micro-Electro-Mechanical Systems”), cujos elementos mecânicos e atuadores são fabricados por meio de tecnologia de microfabricação. Em sua quase totalidade são mecanismos flexíveis, onde o movimento é dado pela flexibilidade da própria estrutura ao invés de serem usados pinos e juntas, muito difíceis de serem montados nessa escala. Um tipo bem simples de atuação é o denominado eletrotermomecânico, onde a base da atuação é a expansão térmica da estrutura resultante do calor gerado pela passagem de uma corrente elétrica.

O presente trabalho objetiva o estudo das técnicas de projeto e fabricação do microatuador eletrotermomecânico, incluindo a sua simulação por método dos elementos finitos e o desenvolvimento de seu sistema de comando de posicionamento, feito através de uma fonte de tensão controlada remotamente. Ao final desta atividade, deseja-se obter o protótipo do sistema completo (atuador e sistema de comando) mas sem a preocupação de fazê-lo capaz de processos automatizados.

2. REVISÃO BIBLIOGRÁFICA

2.1. Metodologia de Projeto de MEMS

O microatuador em questão terá uma atuação bidimensional, manipulando objetos em um plano.

Devido ao seu tamanho (os maiores são da ordem de 1cm), será um mecanismo flexível, dada a dificuldade de se fabricar pinos e juntas nessas dimensões. Os mecanismos flexíveis possuem certas vantagens, como ausência de folga e atrito, porém estão restritos a deslocamentos limitados e podem levar a elevadas tensões mecânicas na estrutura.

Para a sua atuação, escolhe-se um dos vários tipos existentes, cada um com suas vantagens e desvantagens particulares, devendo o projetista optar por uma solução de compromisso.

Para projetar o desenho do microatuador como um mecanismo flexível, foi utilizada otimização topológica, que nos fornece uma solução otimizada do problema. Normalmente, a intuição nos leva a uma solução não otimizada por causa da tendência de utilizarmos formas e estruturas simples de serem tratadas teoricamente. Como um projeto ótimo pode consistir de formas nada usuais, um método numérico pode ser vantajoso, já que é indiferente à complexidade da modelagem teórica do resultado. A otimização topológica é um método para achar uma topologia, ou seja, uma forma estrutural, otimizada em relação a determinadas propriedades bem definidas.

Em um projeto envolvendo otimização topológica, seguimos uma seqüência de passos:

- *Definição do problema:*

Nesta etapa, as propriedades desejadas devem ser determinadas, definindo-se a função objetivo a ser otimizada. Outras características do projeto devem ser definidas também: o formato e o tamanho do domínio de projeto (espaço que pode ser ocupado pelo dispositivo), propriedades do material usado, condições de contorno, quantidade de material a ser usado, entre outras.

- *Otimização Topológica:*

É um procedimento iterativo composto por uma análise de elementos finitos seguida de uma redistribuição do material no domínio de projeto. Para isso, o domínio deve ser discretizado, sendo cada elemento capaz de assumir várias densidades do material utilizado. Uma otimização rápida pode ser obtida através de uma análise de sensibilidade da mudança da densidade dos elementos. O problema desse método está na possibilidade de obtermos uma solução não otimizada ao apontar um arranjo ótimo local. Uma vez obtida a solução supostamente ótima, a única maneira de detectar uma otimização local seria comparando-a com todas as outras possibilidades, o que é impossível devido ao tempo computacional necessário.

- *Projeto Otimizado:*

Com a solução ótima do domínio de projeto, encontrada pelo método de otimização, uma interpretação é necessária, transformando o domínio discretizado em um desenho do dispositivo passível de fabricação. As propriedades desejadas da topologia resultante são verificadas através de uma análise por elementos finitos.

Uma vez com o desenho do mecanismo pronto, passa-se à simulação do MEMS por método de elementos finitos (MEF). Concluindo as devidas análises do comportamento do mecanismo e tendo uma previsão das suas características, pode-se fabricá-lo sem maiores receios.

Com o mecanismo fabricado, a parte de comando deve ser implementada.

2.2. Tipos de Atuadores em Micromanipulação

A seleção do tipo do atuador não é uma tarefa fácil, já que há vários aspectos a serem considerados. Um atuador ideal deve ser facilmente incorporado ao método de otimização topológica, ter potência suficiente para atuar na estrutura flexível do sistema e ainda realizar trabalho sobre a carga desejada, requerer controles elétricos simples e ser de fácil fabricação. Dos vários mecanismos de atuação possíveis, nenhum deles é visivelmente superior para qualquer aplicação escolhida. A escolha do atuador envolve uma solução de compromisso entre as características desejadas.

No caso estudado, e que será usado para fabricação, foi escolhido o atuador eletrotermomecânico, baseado na expansão volumétrica do material do atuador devido ao calor. Outros tipos possíveis são o piezoelectrônico, capacitivo e SMA (“Shape Memory Alloy”).

Para esta escolha, foram analisadas basicamente quatro características:

- *Máxima Performance:*

São de principal interesse o máximo deslocamento, força e trabalho possíveis para um dado tipo de atuação. Uma alta densidade de trabalho (J/m^3) é desejável, já que, de outro modo, seriam necessários atuadores de dimensões elevadas para tarefas simples, impossibilitando a implementação de microatuadores.

Os eletrotermomecânicos, assim como outros baseados na atuação por elongamento (como o piezelétrico), possui grande força e pequeno deslocamento. Também apresenta uma densidade de trabalho relativamente grande.

Os capacitivos tem características contrárias, permitindo grandes deslocamentos, mas com menores forças, além de baixa densidade de trabalho.

Os SMA tem potências maiores que todos os anteriores, combinando deslocamentos, forças e densidade de trabalho elevados.

- *Fabricação:*

Aspectos relacionados à fabricação são as características práticas envolvidas nesta escolha.

Alguns atuadores, como o piezelétrico e SMA, são feitos de materiais mais incomuns, cujas propriedades dependem de uma composição química muito particular. Uma variação pequena poderia fazer com que os requisitos de performance não sejam atendidos.

Para os eletrotermomecânicos e capacitivos, são necessários apenas materiais comuns, o que facilita sua fabricação.

A fabricação e montagem dos piezoelétricos e capacitivos são relativamente difíceis em micromecanismos. Em particular, os capacitivos necessitam de um processo de fabricação com tolerância muito baixa , da ordem de $1\mu\text{m}$.

- *Versatilidade:*

Alguns mecanismos de atuação podem ter sérias limitações quanto ao seu uso.

Como exemplo, os materiais piezelétricos, usados em atuadores desse mesmo tipo, são quimicamente voláteis, o que limita o seu uso a ambientes não agressivos, que não causem sua deterioração. Os eletrotermomecânicos consomem muita potência e esquentam muito, o que torna seu emprego proibitivo em ambientes sensíveis termicamente. Apesar dessa desvantagem, seu comando elétrico é simples. Ambos tem o deslocamento linear em relação à entrada.

Os capacitivos são bem versáteis, apesar de ter um controle mais complicado: o comportamento não é linear em relação à entrada.

Os SMA são mais complicados de serem implementados, necessitando de um tratamento térmico e mecânico para desempenhar devidamente a sua função. Além disso, só atuam em um sentido de movimento, ao contrário dos demais, devendo ser reposicionado por meios externos à posição original para mais um ciclo de movimentação.

Em relação ao comportamento dinâmico, o piezoelétrico e capacitivo possuem uma resposta mais rápida se compararmos com o eletrotermomecânico e SMA, que dependem de efeitos térmicos para a atuação.

- *Adaptação ao método de otimização:*

Como o mecanismo de atuação deverá ser incluído no método de otimização topológica, a sua facilidade em ser representado pelo algoritmo deve ser levada em conta.

Por ser de natureza linear, o eletrotermomecânico é simples de ser incluso no método de otimização.

Assim, o atuador do tipo eletrotermomecânico é uma solução com bom compromisso entre as características citadas acima, não conflitando com nenhuma delas. Por requerer muita potência, este atuador não pode ser usado em diversas aplicações, mas é aceitável em laboratórios, onde podem estar disponíveis fontes de potência relativamente elevada.

2.3. O Atuador Eletrotermomecânico

O atuador eletrotermomecânico é baseado na expansão térmica do sólido que o compõe.

A expansão térmica é o efeito em que o material dilata sob aquecimento. Essa característica é apresentada por todos os materiais sólidos em diversos graus, sendo causada por um aumento na média das distâncias interatômicas. Como está relacionada ao espaçamento interatômico, é dependente das propriedades dos elementos e da estrutura cristalográfica.

A expansão térmica é caracterizada por um coeficiente α . Tanto o deslocamento livre (sem carga) d_f , como a força máxima do atuador F_b ("blocking force", força realizada sobre um objeto restringindo o movimento do atuador) variam linearmente com a diferença de temperatura ΔT para um atuador simples na forma de uma viga engastada em uma das extremidades e livre na outra:

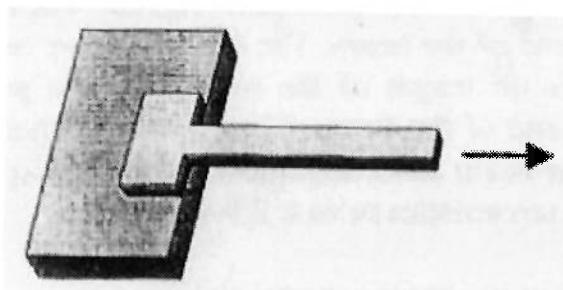


Fig. 2.1 - Atuador eletrotermomecânico simples

$$d_f = l \cdot \alpha \cdot \Delta T$$

$$F_b = EA \cdot \alpha \cdot \Delta T$$

onde l é o comprimento da viga

E é o módulo de elasticidade do material do atuador

A é a área da seção transversal do atuador

No caso do atuador que será usado, diz-se que é ativo, pois o atuador terá meios para se aquecer, sendo este o aquecimento por resistência elétrica (efeito Joule). A figura abaixo ilustra um atuador ativo, uma variação do visto anteriormente:

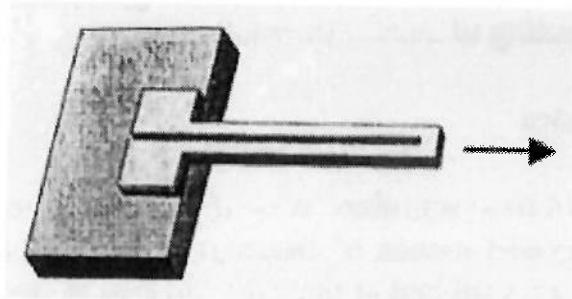


Fig. 2.2 - Atuador eletrotermomecânico ativo

A única diferença é um recorte longitudinal em parte da viga, o que causa um isolamento entre os dois lados e força uma corrente a passar por toda a viga assim que uma diferença de potencial for aplicada aos dois terminais. Para fendas muito estreitas na viga, as equações de deslocamento livre e força máxima ainda são válidas.

A resistência elétrica R do atuador é dada por:

$$R = \frac{4\rho l}{wt}$$

onde ρ é a resistividade do material do atuador
 l , w e t são o comprimento, largura e espessura.

Através da potência dissipada P:

$$P = RI^2 = \frac{U^2}{R}$$

onde I é a corrente que passa pelo atuador
 U é a tensão aplicada aos terminais

pode-se calcular a diferença de temperatura ΔT , responsável pelo deslocamento e força do dispositivo:

$$\Delta T = \frac{P}{H}$$

onde H é o transporte de calor para fora da estrutura (W/K)

O transporte de calor para fora da estrutura (a perda de calor) é a somatória dos efeitos de condução, convecção e radiação, sendo esta última desprezível, apesar do modelo de perda de calor preciso ser muito importante para a consistência do resultado do método de otimização topológica.

Pode-se verificar, então, que a temperatura pode ser controlada através da tensão U imposta aos terminais do atuador.

O material utilizado para a fabricação do atuador é um dos fatores que determinam suas características principais (deslocamento livre e força máxima), visto que são propriedades do material o coeficiente de expansão térmica α e o módulo de elasticidade E. Assim, como foi utilizado apenas um material, podemos separar a influência da forma do atuador, resultado da otimização, e do material escolhido na determinação da performance do atuador. Se fossem utilizados mais de um material, não seria possível fazer essa distinção. Para o caso dos atuadores térmicos, deseja-se um coeficiente de expansão térmica α e o módulo de elasticidade E altos, para que tenhamos o maior deslocamento e força possíveis.

Duas das possíveis alternativas seria o silício e o óxido de silício, dois materiais tradicionalmente usados em micro-usinagem, porém estes não teriam boa performance devido ao baixo coeficiente de expansão α .

O zinco também poderia ser usado pela boa performance (coeficiente de expansão α e módulo de elasticidade E altos), mas deve ser excluído por causa da facilidade de sua fluência (“creep”) sob baixas tensões mecânicas. Em um atuador térmico, a facilidade à fluência não é aceitável devido às altas tensões mecânicas e temperaturas envolvidas. Além disso, o zinco é um material altamente reativo quimicamente e, por oxidar-se rapidamente, suas propriedades elétricas, térmicas e mecânicas seriam alteradas inaceitavelmente.

O manganês possui também boa performance, próxima da do zinco, mas é tóxico e não muito usado em micromáquinas, dificultando aquisição de conhecimento acerca de seu uso.

O alumínio é um dos materiais mais usados em micromáquinas, tendo excelente ou boa performance em todos os aspectos. O principal problema é o baixo ponto de fusão (para o alumínio puro e algumas ligas, por volta de 600°C). Devido

aos requisitos particulares dos atuadores térmicos no caso de estudo, o seu processamento é difícil.

O níquel e o cobre possuem alta performance e são similares. Na preferência de material para a fabricação dos atuadores, estaria abaixo apenas do alumínio, cujo deslocamento é pouco maior e força, pouco menor. Entre o níquel e o cobre, a principal diferença é uma característica dinâmica, que não será considerada já que os atuadores serão projetados para regime quase estático. O material escolhido para a fabricação do microatuador é o níquel.

2.4. Simulação em Método de Elementos Finitos

2.4.1. Fundamentos Teóricos

Para a análise de um mecanismo eletrotermomecânico, tem-se o trabalho conjunto de três tipos de disciplinas: elétrica, térmica e mecânica. Por isso, uma análise completa deve ser feita em três etapas. A parte elétrica fornece a distribuição das correntes elétricas na estrutura; a térmica determina a temperatura de cada ponto do dispositivo devido ao aquecimento por efeito Joule e perda de calor por condução, convecção ou radiação; a mecânica define os deslocamentos do mecanismo devido à distribuição de temperatura.

Cada tipo de domínio pode ser resolvido separadamente (elétrico, térmico e mecânico, nessa ordem), a não ser que haja algum tipo de acoplamento (dependência) entre eles. Essa dependência pode ser uma propriedade do material que varia com a temperatura e afeta a análise elétrica, o que obrigaria a resolver simultaneamente a análise elétrica e a térmica. Esse tipo de análise pode ser feito de forma iterativa, que não será abordada neste trabalho.

Nos casos estudados, os mecanismos são modelados bidimensionalmente, atribuindo-se uma espessura uniforme t . A área da região de interesse (parte da estrutura) é indicada por Ω , e sua fronteira, por Γ .

As equações diferenciais dos três tipos de análise são:

- *Análise Elétrica*

As equações diferenciais correspondentes aos fenômenos elétricos são dadas por:

$$\nabla \cdot (k_e \nabla V) = 0 \quad \text{em } \Omega$$

$$V = V_{\text{espec}} \quad \text{em } \Gamma_{\text{eE}}$$

onde k_e é a condutividade elétrica do material
 V é a tensão elétrica (varia com a posição)
 V_{espec} é a tensão conhecida, aplicada a uma área particular
 Γ_{eE} é a área da fronteira com a tensão elétrica especificada

O resultado dessa análise é a distribuição da densidade de corrente $k_e \nabla V$, decorrente de uma tensão V_{espec} aplicada na fronteira e da condutividade elétrica k_e do material.

- *Análise Térmica*

As equações diferenciais correspondentes aos fenômenos térmicos são:

$$\nabla \cdot (k_t \nabla T) + \nabla^T V k_e \nabla V = 0 \quad \text{em } \Omega$$

$$T = T_{espec} \quad \text{em } \Gamma_{eT}$$

$$n \cdot (k_t \nabla T) = f_{nT} \quad \text{em } \Gamma_{nT}$$

onde k_t é a condutividade térmica do material
 T é a temperatura (varia com a posição)
 T_{espec} é a temperatura conhecida aplicada a uma área particular
 Γ_{eT} é a área da fronteira com a temperatura especificada
 f_{nT} é o fluxo de calor na fronteira dependente da temperatura,
 composto pela condução, convecção e radiação térmica
 n é o versor normal à superfície
 Γ_{nT} é a área da fronteira cuja temperatura é incógnita

O resultado dessa análise é a distribuição da temperatura T no dispositivo a partir do termo de aquecimento pelo efeito Joule $\nabla^T V k_e \nabla V$, da perda de calor por convecção e da temperatura T_{espec} imposta na fronteira.

- *Análise Estrutural*

As equações diferenciais correspondentes aos fenômenos térmicos são:

$$\nabla \cdot \sigma = 0$$

$$\sigma = E(\varepsilon - \alpha(T - T_\infty)I) \quad \text{em } \Omega \qquad u = u_{espec} \quad \text{em } \Gamma_{eU}$$

$$\varepsilon = (\nabla u + \nabla^T u)/2$$

onde σ é a matriz de tesões
 E é a matriz de elasticidade (relaciona deformações com tensões)
 ε é a matriz de deformações
 α é o coeficiente de expansão térmica
 T é a temperatura (varia com a posição)
 T_∞ é a temperatura ambiente

I é a matriz identidade

u é o deslocamento

u_{espec} é o deslocamento conhecido aplicado a uma área particular

Γ_{eU} é a área da fronteira com o deslocamento especificado

O resultado dessa análise são os deslocamentos u no mecanismo a partir da distribuição de temperatura T e dos deslocamentos u_{espec} impostos na fronteira.

Podemos perceber que não existem termos usuais de força, já que não existe uma aplicação de força mecânica sobre a estrutura. O que existe é um termo de deformação térmica (usando α e as temperaturas T e T_∞) relacionado às tensões σ pela matriz de elasticidade E .

Para cada um dos três tipos de análise, existem equações diferenciais que governam o fenômeno em questão. Mas só existem soluções analíticas destas para as mais simples geometrias e condições de contorno do mecanismo analisado. Para que seja possível uma solução facilmente aplicável a um mecanismo geral, é necessário utilizar métodos numéricos, como um programa de elementos finitos.

2.5. Fabricação de MEMS

Os MEMS são fabricados a partir de processos de microfabricação, divididos basicamente em dois tipos: microfabricação em volume (“bulk micromachining”) e microfabricação de superfície (“surface micromachining”). Ambos os processos se iniciam a partir de um substrato, como o silício, por exemplo, e serão descritos brevemente a seguir. O presente estudo concentra-se no processo de microfabricação de superfície.

- A microfabricação DE SUPERFÍCIE mantém o substrato inalterado, e adiciona/remove camadas de filmes com o padrão desejado sobre a superfície do substrato. Essa técnica fornece uma grande flexibilidade na obtenção de estruturas planas XY (altura e largura), com baixo controle sobre a dimensão Z (espessura).
- O processo de microfabricação EM VOLUME envolve a remoção de material do próprio substrato de silício a fim de se obter um produto final desejado; Esse processo permite um controle maior sobre a espessura (dimensão Z) com restrição da flexibilidade nas dimensões X e Y.

A microfabricação DE SUPERFÍCIE envolve deposição, remoção e criação de padrões de filmes finos em um substrato. O substrato geralmente é uma lâmina de silício, por exemplo, um disco circular, de 500 a 1000 μm de espessura e de 3 a 12 polegadas de diâmetro. Para os MEMS, o substrato geralmente tem função apenas de apoio estrutural ou suporte mecânico.

Os processos utilizados nas técnicas de microfabricação podem ser enumerados nos seguintes:

- Deposição de filme
- Oxidação
- Ataque químico ou Remoção de material ou corrosão (“Etching”)
- Litografia

- Difusão
- Implantação de íon

A deposição e oxidação são técnicas aditivas, enquanto o ataque químico é subtrativo. O aspecto mais importante da microfabricação moderna é a litografia, técnica na qual se obtêm os padrões desejados nos filmes depositados. A difusão e a implantação de íons são meios utilizados para introduzir dopantes nos filmes depositados ou no substrato.

DEPOSIÇÃO DE FILMES

A fim de se depositar filmes para a microfabricação de superfície há quatro técnicas principais que podem ser utilizadas: evaporação, “sputtering”, deposição química de vapor (CVD, do inglês “Chemical Vapor Deposition”) e epitaxial. Todos podem depositar filmes com a espessura variando aproximadamente de $0,1\mu\text{m}$ a $4\mu\text{m}$.

- Evaporação: como o próprio nome diz, a técnica de evaporação envolve a deposição do material evaporando-o sobre um substrato. Para se fazer isso, um pedaço de material a ser depositado, denominado “charge”, é aquecido até à sua temperatura de evaporação. O substrato alvo, no qual será feita a deposição, é colocado de tal maneira que sua face esteja diretamente alinhado à vista do “charge”. Esse processo é realizado sob vácuo ($\sim 10^4\text{ Pa}$) a fim de que o filme depositado resultante contenha poucos contaminantes. Ao se aquecer o suficiente, o “charge” começará a evaporar-se. O material se dispersa radialmente do “charge” e uma porção dele se aloja no substrato.
- “Sputtering”: nessa técnica, ocorre o bombardeio de um “charge” com íons de alta energia (tipicamente Ar^+). Os íons retiram átomos da superfície do “charge”. O substrato alvo é posicionado de tal maneira que os átomos desprendidos se alojem em sua superfície.
- Deposição química de vapor: ocorre a deposição de filmes através de reações químicas. Os substratos são posicionados dentro de um tubo de um forno circular e são aquecidos com bobinas resistivas (o aquecimento é utilizado

para aumentar as taxas das reações químicas). Gases reagentes são lançados dentro do tubo e reagem na superfície do substrato, depositando um sólido reagente. Se as condições apropriadas são satisfeitas, e os gases são escolhidos corretamente, o material desejado será depositado. Obviamente, os materiais que podem ser depositados são limitados pela química.

- Epitaxial: constitui uma técnica de crescimento de cristais. Geralmente, o substrato é um material monocristalino, como, por exemplo, um monocristal de silício. Nessa técnica, este cristal único é estendido. O epitaxial pode ser realizado apenas em materiais monocristalinos e apresenta condições e processos complicados.

Comparação entre as técnicas de deposição

Os fatores utilizados para comparar as técnicas de deposição são:

- conformidade: refere-se à capacidade de o material se depositar em topologias distintas;
- uniformidade: refere-se à variação na espessura da camada depositada ao longo do substrato;
- contaminação: refere-se à quantidade de materiais indesejados que são introduzidos na camada depositada;
- materiais que a técnica pode depositar; e
- tensão residual: tensão entre a camada depositada e o substrato e/ou outros filmes finos. A tensão residual pode ser resultado de: diferença nos coeficientes de expansão térmica entre os vários filmes e o substrato, dopantes e gradientes de contaminação, discordância dos contornos de grão (i.e., nas interfaces filme-filme ou filme-susbrato).

A Tab. 2.1 abaixo mostra a comparação entre as técnicas de deposição:

Tab. 2.1 – Comparação entre as técnicas de deposição

Técnica de deposição	Conformidade e Uniformidade	Contaminação	Tensão residual	Materiais que podem ser depositados
Evaporação	Pobre	Pobre	Nenhum	Qualquer material que pode ser fundido em uma temperatura razoável
Sputtering	Pobre	Pobre	Muito baixa	Qualquer material
CVD	Excelente	Boa	Alta	Limitado pela química
Epitaxial	Excelente	Boa	Nenhuma	Camadas monocristalinas

OXIDAÇÃO

A oxidação é outra técnica aditiva. A oxidação é realizada colocando-se os substratos em um ambiente oxidante (p. ex., uma mistura de O₂ e H₂O para substratos de silício) e aquecimento. Esse processo pode ser utilizado para depositar até 2 μm de dióxido de silício.

Além desta técnica de oxidação usual, também podem ser utilizados óxidos como, por exemplo, o SiO₂ misturados com um agente (algum tipo de solvente volátil) que aumenta a viscosidade do óxido fá-lo agir como um líquido de elevada viscosidade. Este líquido de viscosidade elevada é depositado sobre o substrato, que é girado a altas velocidades para produzir uma camada uniforme de óxido líquido. O substrato e o óxido são aquecidos e o solvente evapora-se, deixando apenas a camada de óxido. A camada de óxido produzida é pobre em comparação com a obtida a partir oxidação usual (aquecimento do substrato em um ambiente oxidante); a interface não é tão robusta, e a camada depositada geralmente possui mais contaminantes do que a oxidação comum.

Os óxidos são muito importantes para MEMS porque eles são freqüentemente utilizados como camadas de sacrifício. Uma camada de sacrifício é aquela depositada para o único propósito de criar-se uma geometria desejada. Ela é posteriormente removida.

ATAQUE QUÍMICO

O ataque químico é uma técnica utilizada para a remoção controlada de material do substrato e seus filmes finos depositados. Há dois tipos de ataque químico: úmido e seco.

O ataque químico úmido utiliza reagentes químicos líquidos. O substrato é colocado dentro de um reagente e agitado (para se ter uma boa uniformidade da taxa de remoção) por um certo período de tempo. Quando o ataque químico é completado, os substratos são removidos. As taxas de ataque químico são bem conhecidas para muitos reagentes padrões e os materiais com os quais eles devem entrar em contato.

O ataque químico seco tipicamente utiliza íons energéticos para bombardear o substrato, do qual é retirado material em um processo similar ao “sputtering”. As taxas de ataque químico seco são menos acuradas.

Os dois aspectos mais importantes do processo de ataque químico são: seletividade e anisotropia. A seletividade de um reagente refere-se à quanto seletivo este é para atacar um material em relação a outro. Por exemplo, o HF (ácido fluorídrico), um reagente de ataque químico úmido, é extremamente seletivo entre o silício e o óxido; ele irá remover centenas de micron de óxido antes de remover um micron de silício.

A outra característica importante dos reagentes é sua anisotropia. Um reagente completamente anisotrópico irá remover material em apenas uma direção. Um reagente isotrópico irá provocar a remoção em todas as direções com a mesma taxa.

Os processos de remoção individuais irão ser diferentes um do outro em relação à seletividade e anisotropia. Porém, geralmente é verdadeiro dizer-se que o ataque químico úmido é muito seletivo e quase completamente isotrópico. Por outro lado, o ataque químico seco é quase sempre não-seletivo, porém muito anisotrópico.

Tipicamente, deseja-se alta seletividade e alta anisotropia. Há técnicas de ataque químico que incorporam aspectos de ambos, úmido e seco. Essas técnicas são moderadamente seletivas e moderadamente anisotrópicas. Quase sempre, existe uma solução de compromisso entre seletividade e anisotropia.

LITOGRAFIA

A litografia une os processos aditivo e subtrativo a fim de criar os padrões desejados nas camadas depositadas, por isso constitui o passo mais importante de todo o processo. Porém, também é o mais complicado. Dessa forma, muitos processos de fabricação são descritos pelo número de passos de litografia que eles envolvem. Quanto mais litografias, ou máscaras, ou passos, mais difícil é o processo de fabricação.

Há muitos tipos de litografia, incluindo, dentre outros, a foto (ultravioleta), a ultravioleta profunda (DUV) e por raios-X. A fotolitografia é a mais barata e, portanto, a mais comum na indústria. Porém, os outros três oferecem uma melhor resolução que a fotolitografia. A fotolitografia é o processo abordado neste trabalho, mas todas as técnicas de litografia usam os mesmos passos essenciais. Eles diferem apenas no método de exposição.

A litografia pode ser dividida nos seguintes passos:

1. Deposição do Resiste: O substrato é fixado em uma base por sucção através de vácuo e girado em alta rotação. Em seguida, um composto chamado fotoresiste (PR, do inglês “Photoresist”) é adicionado ao substrato em rotação, que é coberto por uma camada altamente uniforme de PR. O fotoresiste é um fluido de viscosidade elevada formado por um composto orgânico sensível à luz e um solvente volátil. O solvente é utilizado para obter-se a viscosidade apropriada. Há dois tipos de resisto: positivo e negativo. Quando exposto a certa radiação, o composto orgânico no resisto positivo muda a forma e se torna mais solúvel. O resisto negativo se torna menos solúvel quando exposto. O resisto positivo é o mais popular para a fotolitografia porque fornece uma resolução melhor. A Fig. 2.3 abaixo ilustra o processo de deposição de resisto.

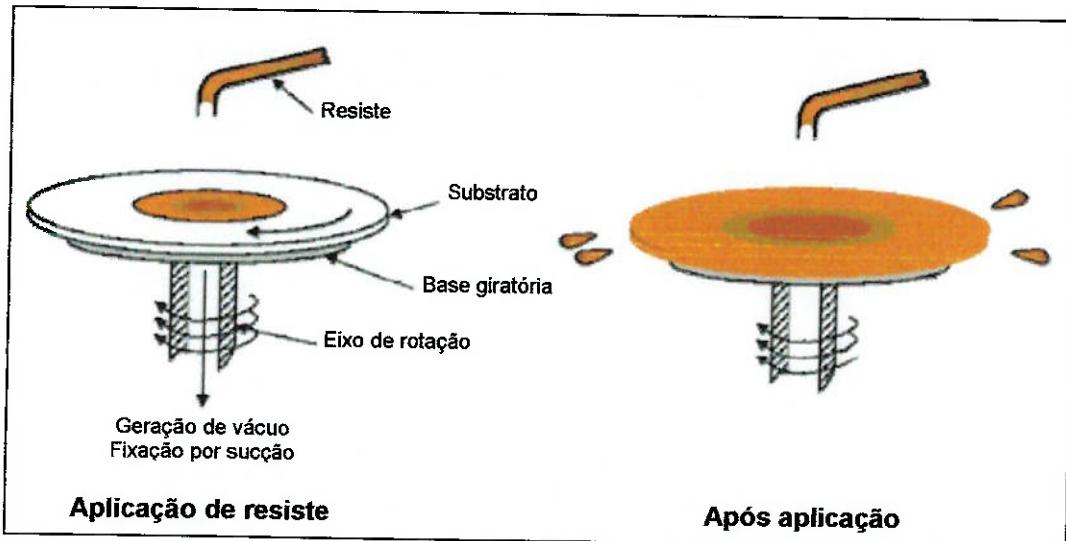


Fig. 2.3 – Deposição de resist

2. Aquecimento do resist em baixa temperatura: Os substratos com resist são então aquecidos à baixa temperatura por poucos minutos para promover a adesão entre as duas superfícies.
3. Alinhamento da máscara ao substrato: Uma máscara contém o padrão que se deseja transferir ao substrato, ou seja, o projeto no plano XY. Ele é composto de um substrato com finalidade de suporte mecânico que é transparente à exposição de radiação (p. ex., vidro é a camada de suporte para a fotolitografia UV) e uma camada de material com o padrão a ser transferido, que é opaco em relação à exposição de radiação (p. ex., em uma fotolitografia, uma liga de cobalto é aplicada ao vidro). O padrão na máscara é obtido por outro processo litográfico. Uma vez que a máscara está produzida, ela é posicionada sobre o substrato. Geralmente, para se obter uma resolução melhor, um sistema de redução de 10 para 1 é colocado entre a máscara e o substrato (i.e., a imagem transferida ao substrato é 10 vezes menor que aquela na máscara). Podem ser utilizados métodos simples para posicionar a máscara em contato físico com o substrato. A máscara é então alinhada manualmente com o substrato (esta é uma tarefa lenta e representa o maior custo na manufatura em termos de tempo e dinheiro).

4. Exposição do resiste: Uma vez que a máscara e o substrato estão alinhados, uma quantidade controlada de radiação expõe o resiste através da máscara. Apenas as partes de resiste que estão sob a porção transparente da máscara são expostos.
5. Revelação do resiste. Depois da exposição, coloca-se o resiste em contato com um solvente. No caso de resiste positivo, apenas as partes expostas são removidas, enquanto que, nos resistes negativos, as porções não-expostas são removidas.

Através do procedimento descrito acima, um padrão é transferido ao resiste, que em seguida, é utilizado como uma máscara para a deposição do material que constitui o MEMS, ou seja, após a revelação do resiste, é feito o preenchimento dos locais em que o resiste foi sensibilizado (no caso de resiste positivo) e removido. Após a deposição, utilizando solventes químicos, o resiste não-sensibilizado é dissolvido, restando apenas a estrutura desejada. Para se conseguir estruturas mais complexas, esse procedimento é repetido várias vezes, conforme mostra a Fig. 2.4 para o caso de um micromecanismo de níquel.

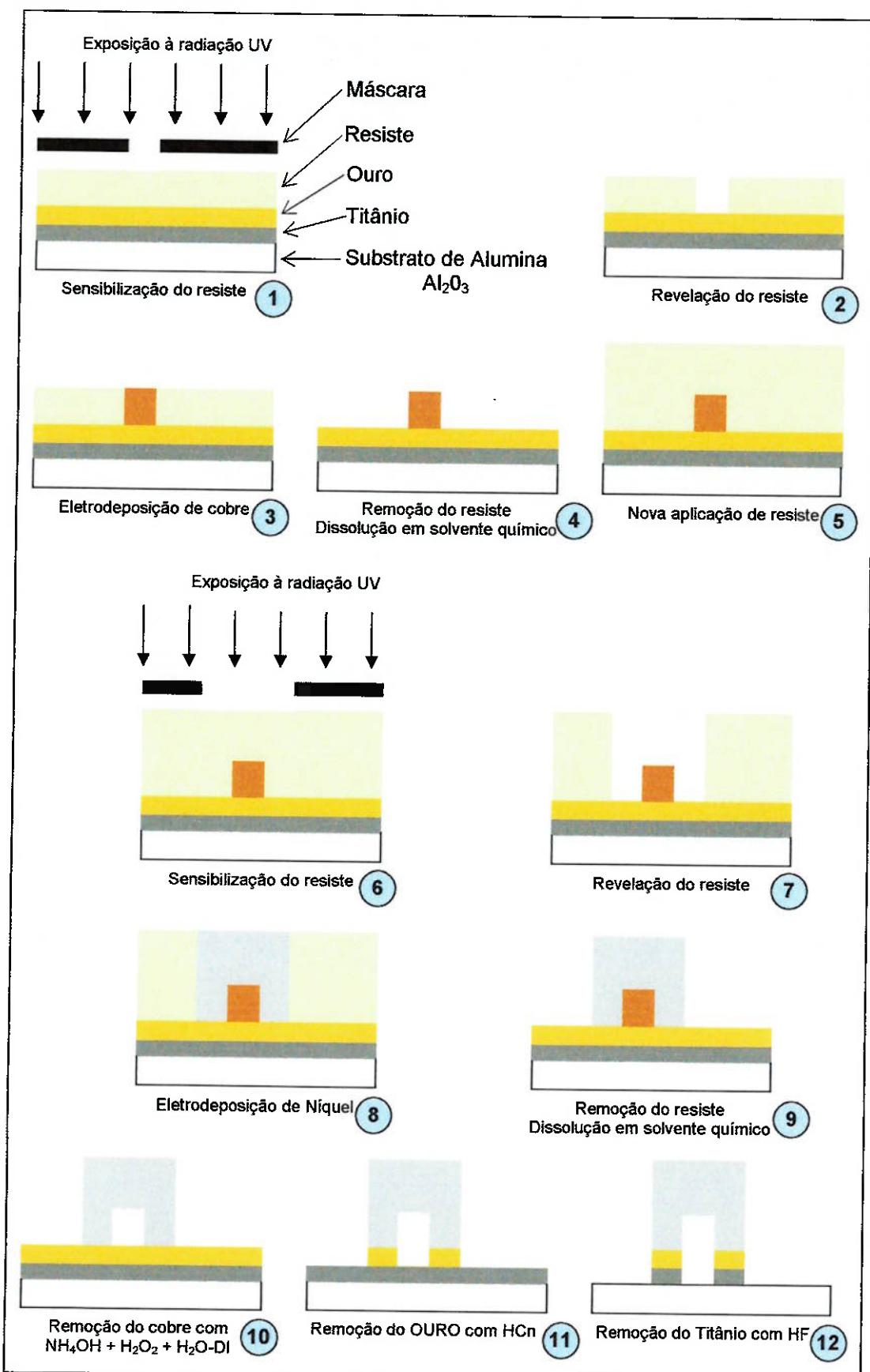


Fig. 2.4 – Processo de Fotolitografia

DIFUSÃO E IMPLANTAÇÃO DE ÍON

A difusão e a implantação de íon são técnicas para introduzir dopantes (o fósforo, o boro e o arsênio são os mais comuns) em um substrato para que os semicondutores se tornem mais condutivos. A difusão funciona colocando-se um filme fino altamente dopado em cima (ou embaixo) de um substrato não-dopado. Este conjunto é então colocado em um forno e aquecido por um determinado período de tempo. Gradientes de concentração irão guiar os dopantes para as camadas não-dopadas.

A implantação de íon é semelhante ao “sputtering”. O dopante é convertido em um íon e acelerado em direção ao substrato. O dopante move-se tão rápido que ele simplesmente colide no substrato e ali permanece. Um recozimento é necessário após a implantação de íon para reparar danos causados pela implantação. Isto é, o substrato é aquecido a uma alta temperatura temporariamente. Isso permite que os átomos se rearranjem, e tornando os dopantes, partes ativas da matriz cristalina.

2.6. Estratégia de Comando

2.6.1. Visão Geral

O comando do MEMS eletrotermomecânico é realizado por referência de tensão, fornecida por uma fonte de tensão e corrente contínuas (fonte DC), e pode ser feito essencialmente através de duas maneiras:

- Comando direto, através do ajuste da tensão desejada no próprio painel da fonte;
- Comando remoto, que utiliza uma interface com o usuário para ajustar a tensão desejada na fonte.

Neste trabalho será abordada apenas a estratégia de comando remoto, visto que a operação direta é intuitiva e trivial.

A Fig. 2.5 apresenta a visão geral da estratégia utilizada para o comando remoto da fonte DC e o fluxo de informações necessárias para a implementação do comando.

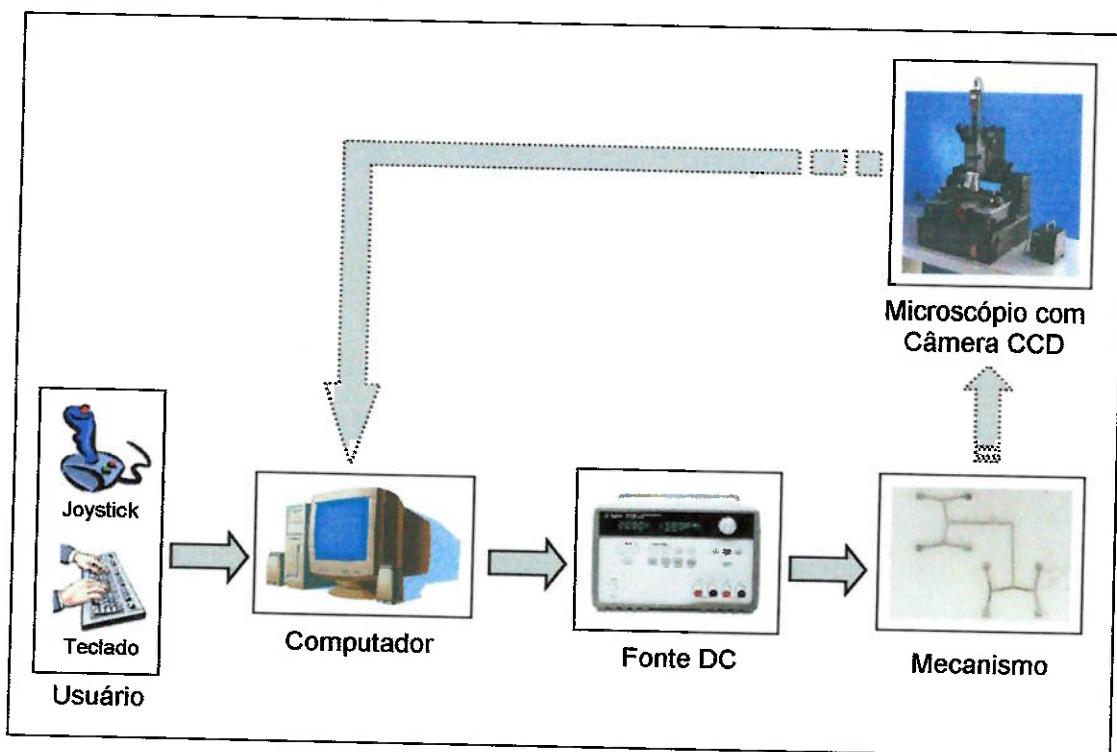


Fig. 2.5 – Visão geral do comando remoto da fonte DC

O usuário interage com o sistema para obter o movimento desejado do microatuador através de um teclado ou de um “joystick”, fornecendo o valor da tensão necessária para realizar tal movimento no primeiro caso, ou simplesmente o deslocamento desejado no último caso. Essa informação é enviada a um computador que realiza o processamento dos dados que serão enviados à fonte DC. A fonte DC fornece a tensão de alimentação do microatuador. O movimento realizado é observado através de um microscópio cuja imagem é transmitida a um computador (que não necessariamente é o mesmo que realiza o processamento dos dados) para documentação.

As interfaces de software e hardware entre os componentes do sistema de comando remoto são apresentadas Fig. 2.6.

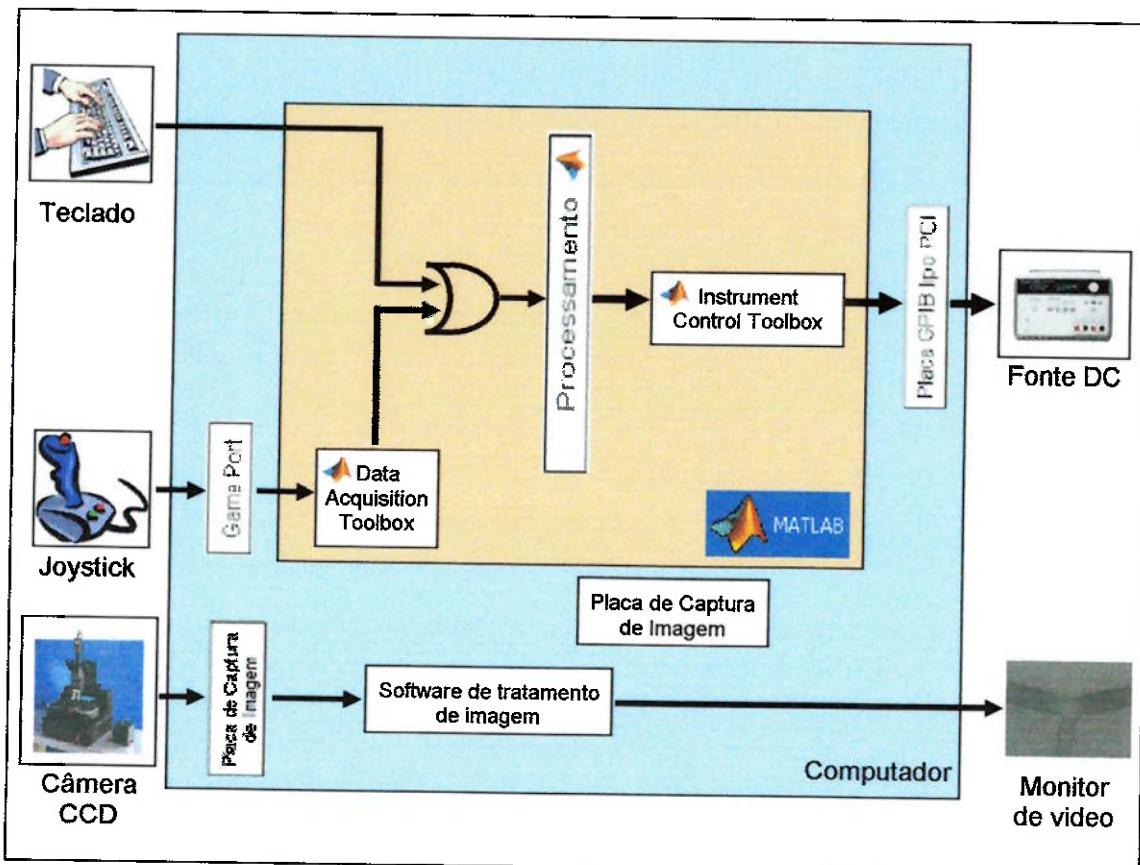


Fig. 2.6 – Interfaces de “hardware” e “software”

A plataforma de software utilizada para o processamento dos dados é o *MATLAB, The Language of Technical Computing*, versão 6.5 Release 13, fornecido pela *The MathWorks, INC.*

O MATLAB pode receber as informações fornecidas pelo usuário através de duas interfaces:

- Teclado, para o qual o MATLAB já possui métodos para a leitura de dados;
- Joystick, conectado ao “Game Port” (entrada analógica) do computador. Os dados são enviados ao MATLAB através do “Data Acquisition Toolbox”, que possui funcionalidade de comunicação com dispositivos de aquisição de dados.

O MATLAB então realiza o processamento dos dados, o que significa traduzir o deslocamento desejado pelo usuário em um valor de tensão a ser enviado à fonte DC para o comando do microatuador. A comunicação entre o computador e a fonte DC é realizada através da interface GPIB, e os dados processados no MATLAB são disponibilizados nessa interface através do “Instrument Control Toolbox”.

O movimento do microatuador é observado em um microscópio, que possui uma câmera CCD (“Charge Couple Device”) acoplada a ele. As imagens obtidas são enviadas a um computador através de uma placa de captura de imagens, e o tratamento das imagens – obtenção de fotos e filmes – é realizado utilizando-se um “software” comercial.

Nas seções seguintes serão melhor abordadas as interfaces o MATLAB e o “joystick” através do “Data Acquisiton Toolbox” e entre o MATLAB e a fonte DC através do “Instrument Control Toolbox”.

2.6.2. Interface entre o MATLAB e o “joystick”

2.6.2.1.“Data Acquisition Toolbox”

VISÃO GERAL

Uma maneira de o MATLAB comunicar-se com o meio externo é através do “Data Acquisiton Toolbox”, que fornece um conjunto de ferramentas para o controle e a comunicação com “hardwares” de aquisição de dados compatíveis com o computador. Há um nível de “software”, denominado “adaptador”, que constitui a interface entre o “Data Acquisition Toolbox” e o dispositivo.

O “Data Acquisition Toolbox” inclui adaptadores para os seguintes fabricantes:

- Advantech
- Agilent Technologies
- IOTech
- Keithley Instruments
- Mesurement Computing
- National Instruments
- United Eletronic Industries
- Windows soundcards devices

O “hardware” de aquisição de dados deste trabalho é o “Game Port” ao qual será conectado o “joystick” para o comando da referência de tensão da fonte DC. O adaptador para este hardware foi construído utilizando o “Data Acquisition Toolbox Adaptor Kit”, que será abordado abaixo, e o Microsoft API como interface de hardware entre o “joystick” e o MATLAB.

DATA ACQUISITION TOOLBOX ADAPTOR KIT

O “Data Acquisition Toolbox Adaptor Kit” consiste de três componentes principais:

- A documentação do “toolbox”;
- O código-fonte de um adaptador de demonstração, localizado no diretório (MATLABROOT) \toolbox\daq\daqadaptor, onde MATLABROOT é a raiz do diretório de instalação do MATLAB. Este diretório contém dois subdiretórios: **AdaptorKit** e **Demo**.

O subdiretório **AdaptorKit** contém arquivos que são comuns a todos os adaptadores, e que são normalmente incluídos na subpasta **include**, conforme será visto adiante. O subdiretório **Demo** contém arquivos que são específicos para um adaptador em particular – neste caso, o adaptador de demonstração. A lista de arquivos de ambos os subdiretórios é dada na Tab. 2.2.

- O código-fonte completo para as bibliotecas dos adaptadores inclusos no “Data Acquisiton Toolbox”. Todos os arquivos dos códigos-fonte estão localizados na pasta (MATLABROOT) \toolbox\daq\daq\src, onde MATLABROOT é a raiz do diretório da instalação do MATLAB. Esta pasta contém as subpastas listadas na Tab. 2.3.

Tab. 2.2 – Código-fonte do Adaptador de Demonstração

Subpasta	Arquivo	Descrição
AdaptorKit	AdaptorKit.h	Contém definições para classes de dispositivos não-específicos e “templates” que são utilizados para criar todos os adaptadores. As classes definidas fornecem suporte para sincronização por software (“software clocking”), armazenamento (“buffering”) e disparo (“triggering”).
	AdaptorKit.cpp	Define as funções para as classes contidas no AdaptorKit.h. Contém GUIDs (“Globally Unique Identifiers”) para a máquina. Define temporizadores de alta e baixa resoluções utilizando métodos do Windows Multimedia.
	daqmex.idl	Arquivo de definição da interface utilizado para definir as interfaces seriais (COM) da máquina de aquisição de dados (daqmex).
	daqmex.h	Construído a partir do daqmex.idl pelo compilador MIDL do Microsoft IDL.
	DaqmexStructs.h	Define a maioria das estruturas utilizadas pelas bibliotecas do adaptador, e pela máquina de aquisição de dados
Demo	SArrayAccess.h	Define classes e “templates” utilizadas para criar e gerenciar matrizes e vetores seguros.
	demo.dsp	Arquivo de projeto para a construção do adaptador de demonstração.
	demo.def	Arquivo de definição para a construção do demo.dll.
	demo.cpp	Define o ponto de entrada no demo.dll
	demo.rc	“Resource script file” gerado pelo Microsoft Developer Studio.
	demo.idl	Arquivo de definição de interface para o adaptador de demonstração. Todas as interfaces específicas desse adaptador são definidas aqui.
	resource.h	Arquivo gerado pelo Microsoft Developer Studio. Contém as definições para as constantes utilizadas pelo programa do adaptador de demonstração.
	demoin.h	Define a classe Cdemoin, que implementa a interface da entrada analógica ImwInput. Essa interface fornece a sincronização por software.
	demoin.cpp	Define as funções para a classe Cdemoin, que é definida em demoin.h.
	demoadapt.h	Define a classe Cdemoadapt, que implementa a interface ImwDemoadapt. Essa interface declara os métodos que são comuns à todo o adaptador.
	demoadapt.cpp	Define as funções para a classe Cdemoadapt, definida no demoadapt.h.

StdAfx.h	Define algumas instruções para o compilador, e internamente, inclui arquivos de cabeçalhos padrões do sistema.
StdAfx.cpp	Internamente, inclui cabeçalhos padrões do sistema. Ambos StdAfx.cpp e StdAfx.h fornecem uma melhor organização das seções de cabeçalho dos arquivos no projeto.

Tab. 2.3 – Adaptadores do Data Acquisition Toolbox

Nome da Pasta	Descrição
computerboards	Contém o código-fonte completo para construir a biblioteca do adaptador para os dispositivos da ComputerBoards (Mesurement Computig Corp.). O nome do adaptador é cbl e o nome da biblioteca do adaptador é mwcbl.dll .
hpe1432	Contém o código-fonte completo para construir a biblioteca para os dispositivos da Agilent Technologies E1432/33/34. O nome do adaptador é hpe1432 e o nome da biblioteca do adaptador é mwhpe1432.dll .
mwnidaq	Contém o código-fonte completo para construir a biblioteca do adaptador para os dispositivos da National Instruments compatíveis com o driver NI-DAQ. O nome do adaptador é nidaq e o nome da biblioteca do adaptador é mwnidaq.dll .
winsound	Contém o código-fonte completo para construir a biblioteca do adaptador para uma placa de som genérica, que utiliza o “driver” Windows Waveform Audio. O nome do adaptador é winsound e o nome da biblioteca do adaptador é mwwinsound.dll .
keithley	Contém o código-fonte completo para construir a biblioteca do adaptador para os dispositivos da Keithley Instruments. O nome do adaptador é keithley e o nome da biblioteca do adaptador é mwkeithley.dll .
include	Contém arquivos comuns para construir todas as bibliotecas dos adaptadores. Esta pasta é praticamente idêntica à pasta AdaptorKit , do adaptador de demonstração. Porém, contém os três arquivos adicionais: <ul style="list-style-type: none"> • daqtbxver.h – Arquivo de controle de versão • thread.h – Contém definições da classe <i>thread</i> e das classes necessárias para iniciar e manter <i>threads</i> seguras (como mutex, semáforos) • cirbuf.h – Define a classe que implementa o “buffer” circular

2.6.3. Interface entre o MATLAB e a fonte DC

A comunicação entre o computador e a fonte DC é realizada através de um barramento GPIB, utilizando-se uma placa GPIB tipo PCI, conectada à placa-mãe do computador. Os dados processados no MATLAB são disponibilizados nesse barramento através do “Instrument Control Toolbox”. Esses dois serão os assuntos abordados nas seções que se seguem.

2.6.3.1.A INTERFACE GPIB

VISÃO GERAL

O GPIB (“General Purpose Interface Bus”) é uma interface definida pelo IEEE (“Institute of Electrical and Electronics Engineers”) – IEEE 488 – para a transferência de dados entre computadores e instrumentos. Ele é amplamente utilizada para permitir que instrumentos de teste sejam controlados remotamente, embora também seja usada em muitas outras aplicações.

As principais características do GPIB são:

- Podem ser conectados até 15 dispositivos no barramento
- Comprimento máximo do barramento de 20m
- Distância máxima entre dispositivos de 2m
- É enviado 1 byte de informação digital de forma paralela por vez
- Taxa máxima de transmissão de dados de 1 Mbyte/s

Os equipamentos no barramento se enquadram em uma de três categorias: controladores, faladores ou ouvidores, embora qualquer equipamento possa cumprir mais de uma função. O controlador possui o papel mais importante no barramento e, geralmente, é um computador, que sinaliza aos instrumentos as várias funções que deverão realizar. Ele também assegura que não ocorram conflitos no barramento. Se dois faladores tentassem conversar ao mesmo tempo, os dados poderiam se corromper, e a operação de todo o sistema estaria seriamente comprometida.

ESTRUTURA DO BARRAMENTO

O barramento tem um total de dezesseis linhas ativas. Estas linhas podem ser organizadas em três categorias. Oito linhas são usadas para a **transferência de dados**, três são usadas para o **controle da transferência de dados** (protocolo “**handshake**”), e as outras cinco são usadas para o **gerenciamento geral do barramento**, carregando informações de status e de controle.

A Fig. 2.7 ilustra as linhas do barramento no conector padrão de 24 vias.

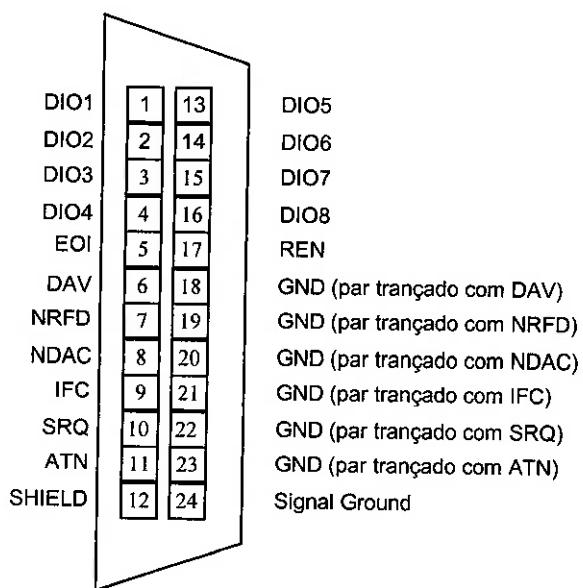


Fig. 2.7 – Conector de 24 vias e suas correspondentes linhas

As linhas são organizadas da seguinte maneira:

- **Linhas de dados:** 8 linhas, DIO1 – DIO8, usadas para a transferência de dados e comandos, 1 byte por vez. Note que os números começam com 1, que corresponde ao bit 0 (2^0) no byte de dados, de acordo com a convenção usual.
- **Linhas do protocolo “handshake”:** 3 linhas, usadas para controlar a transferência de informações nas linhas de dados:

DAV	“Data Valid”
NDAC	“Not Data Accepted”
NRFD	“Not Ready for Data”

- **Linhas de controle:** 5 linhas, para o controle geral das atividades dos instrumentos e do barramento:

ATN	“Attention”
IFC	“Interface Clear”
REN	“Remote Enable”
SRQ	“Service Request”
EOI	“End or Identify”

- **Linhas de terra:** 8 linhas, para blindagem e retorno de sinal

OPERAÇÃO

A operação do barramento é baseada no protocolo “handshaking”. As três linhas DAV, NDAC e NRFD controlam esse protocolo, na seqüência “Nós estamos prontos – Aqui estão dos dados – Nós os recebemos” para a transferência de informações através do barramento.

Todos os ouvidores no barramento usam a linha NRFD para indicar que estão prontos para receber os dados. Se um ouvidor mantém esta linha em “low”, isso impede que qualquer transferência de dados seja iniciada. Apenas quando todos os instrumentos liberarem a linha NRFD, e esta estiver em “high”, o próximo estágio pode ser iniciado.

Os dados são colocados nas linhas de dados pelo falador e a linha DAV é levada para “low”, indicando que os dados no barramento são válidos. Isso sinaliza para todos os ouvidores que eles são capazes de ler os dados disponíveis nas linhas de dados. Durante esta operação, a linha NDAC é mantida em “low” por todos os ouvidores ativos, i.e. aqueles que têm sido instruídos para receberem os dados. Somente quando eles tiverem lido os dados é que cada dispositivo irá liberar a linha NDAC. Quando o último dispositivo liberar esta linha seu nível irá subir para “high” e o falador saberá que todos os dados foram recebidos e o próximo byte de dados pode ser transferido.

A Fig. 2.8 ilustra o processo de transferência do protocolo *handshaking*.

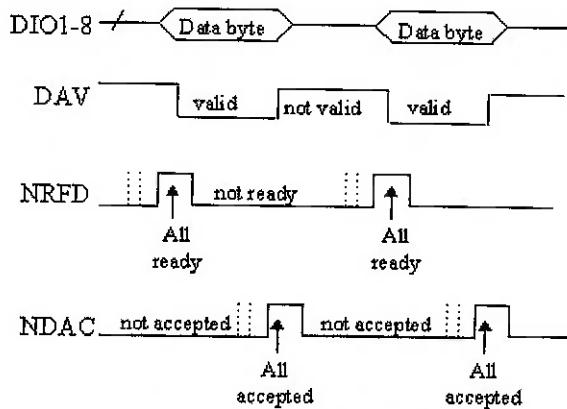


Fig. 2.8 – Processo de transferência do protocolo “handshaking”

Ao se transferir os dados dessa maneira (assíncrona), eles são colocados no barramento numa taxa adequada para o falador, e são mantidos até que o ouvidor mais devagar os tenha recebido. Dessa forma, sempre se consegue uma taxa ótima de transferência de dados, e não há problemas de especificação e de interface associados com velocidades nas quais os dados devem ser transferidos.

GERENCIAMENTO DO BARRAMENTO

Adicionalmente ao processo de transferência do protocolo “handshaking”, o barramento também incorpora um grande número de facilidades de gerenciamento do barramento que são muito úteis, permitindo que os instrumentos sejam controlados de uma maneira muito flexível.

Cada instrumento tem seu próprio endereço. Isso permite que o controlador e o falador conversem com um dispositivo específico. É imperativo, obviamente, que cada dispositivo tenha um endereço próprio e único no barramento. Se não, podem ocorrer conflitos e a operação do sistema irá falhar.

Geralmente, tipos diferentes de instrumentos têm seus próprios valores padrões configurados na fábrica, durante a manufatura. Entretanto, em todos os casos deve ser possível mudar-se o endereço. Mesmo que apenas 15 dispositivos são permitidos no barramento a cada vez, valores de endereços até 31 são permitidos. Isso permite uma certa flexibilidade na definição de endereços específicos para certos tipos de equipamentos dentro de uma organização.

Uma das linhas mais importantes é a linha ATN, através da qual, o controlador é capaz de sinalizar se os dados a serem colocados na linha de dados são informações de controle, ou dados propriamente ditos. Quando esta linha é colocada em “low”, diz-se que o barramento está em **modo de comando**, e os comandos do barramento podem ser colocados no mesmo. Um dos comandos mais comuns é dar o endereço de um instrumento para o qual os dados serão enviados. Além disso, quando a linha é colocada em “low”, qualquer falador desiste do controle da linha DAV e a transmissão de dados é encerrada. Outro resultado é que todos os ouvidores, quer ativos ou não, irão escutar os dados de controle sendo transmitidos.

Quando a linha ATN está em “high”, diz-se que o barramento está no **modo de dados** e ocorre a transferência de dados entre os instrumentos. No modo de comando, nem todas as linhas de dados são utilizadas. O bit 7 é ignorado, enquanto que os bits 5 e 6 indicam o tipo de informação a ser transmitida. Pode ser um comando do barramento, um endereço de falador ou um endereço de ouvidor.

Há outros quatro controles que são amplamente utilizados no barramento: as linhas IFC, SRQ, REN, e EOI.

A linha IFC é usada pelo controlador para dar um “reset” no barramento e colocá-lo num estado quiescente (temporariamente inativo). Qualquer falador ou ouvidor que está ativo é desativado e o controle retorna ao controlador. Essa linha não é utilizada num curso de uma operação normal. Porém, pode ser usada quando o sistema precisa de uma re-configuração ou no ligamento inicial, quando o barramento pode estar em um estado aleatório.

O REN é uma função usada para configurar a operação de todos os instrumentos conectados em remoto, não permitindo configurações manuais a partir do painel frontal.

O EOI é uma função opcional utilizada para indicar o fim de uma transferência de dados de vários bytes. Como uma alternativa, o último byte pode ser um caractere especial, como “carriage return” (CR) ou “line-feed” (LF). Isso pode ser interpretado pelo dispositivo controlado, de acordo com a configuração de cada software.

O SRQ é uma função de interrupção. Quando um dispositivo coloca a linha SRQ em low, indica que se deseja interromper as atividades correntes de tal maneira

que uma atenção pode ser dada para um evento particular. Um processo chamado “polling” realiza isso. Essencialmente, ele apenas solicita a informação do status dos dispositivos no barramento.

POLLING

Há duas maneiras através das quais os instrumentos no barramento podem realizar o “polling”: paralelo e serial.

O “polling” paralelo pode operar apenas com até 8 instrumentos. Isso ocorre porque cada um dos dispositivos irá retornar um bit de status por uma das 8 linhas de dados. Para o “polling” paralelo, o controlador coloca as linhas ATN e EOI em “low”, e cada instrumento responde transmitindo um mensagem de status de 1 bit.

O “polling” serial é mais flexível, porém é mais demorado. Neste caso, o controlador envia a cada instrumento, um comando de habilitação (“enable”) de “polling” serial. Este é um dos comandos do barramento que pode ser enviado quando a linha ATN está mantida em “low”. Quando um instrumento recebe esse sinal, ele retorna uma informação de status de 8 bits. Quando o controlador recebe os dados de status ele envia um sinal de comando para desabilitar (“disable”) o “polling” serial e retorna o barramento e os instrumentos a seu modo de dados normal.

2.6.3.2.“INSTRUMENT CONTROL TOOLBOX”

VISÃO GERAL

O “Instrument Control Toolbox” consiste de um conjunto de funções .m – funções “ponto m” – construídas no ambiente do MATLAB com as seguintes características:

- Uma estrutura para comunicação com instrumentos que suportam interface GPIB (IEEE-488), o padrão VISA, os protocolos TCP/IP ou UDP, e as interfaces seriais (RS-232, RS-422, e RS-485).

- Funções para transferência de dados entre o MATLAB e o instrumento:
 - Os dados podem ser binários (numéricos) ou texto.
 - Dados de texto podem ser quaisquer comandos utilizados pelo instrumento, como, por exemplo, os comandos da linguagem SCPI (“Standard Commands for Programmable Instruments”).
 - A transferência pode ser síncrona e bloquear a linha de comando do MATLAB, ou assíncrona e não bloquear a linha de comando do MATLAB.
- Comunicação baseada em eventos.
- Funções para armazenar dados e informações de eventos em um arquivo texto.
- Ferramentas que facilitam o controle do instrumento em um ambiente gráfico fácil de usar.

As informações são transferidas entre o MATLAB e o instrumento através de dois componentes: as funções .m – funções “ponto m” são os arquivos com a extensão do MATLAB – e os adaptadores de drivers de interface (“Interface driver adapter”) A Fig. 2.9 ilustra como as informações fluem de componente para componente.

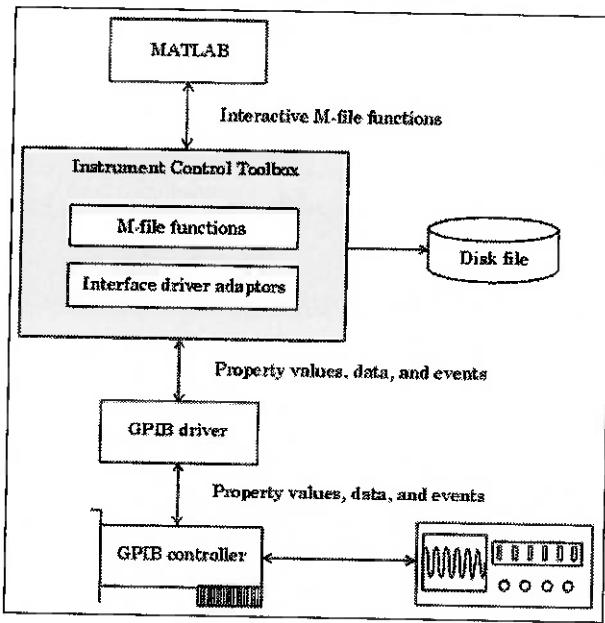


Fig. 2.9 – Fluxo de informações entre o MATLAB e o instrumento

As informações consistem de:

- **Valores de propriedades** (“property values”): definem o comportamento da aplicação do instrumento de controle. De maneira geral, pode-se pensar em propriedade como uma característica do “toolbox” ou do instrumento que pode ser configurada para adequar-se às necessidades. As Tab. 2.4 e 2.5 apresentam as propriedades configuráveis e específicas da interface GPIB.
- **Dados** (“data”): pode-se escrever dados no instrumento e ler dados do instrumento. Os dados podem ser binários (numéricos) ou em forma de textos.
- **Eventos** (“events”): ocorre um evento depois que uma condição é satisfeita e pode resultar em um ou mais “callbacks”. Os eventos só podem ser gerados se as propriedades associadas forem configuradas. Um evento pode ser utilizado, por exemplo, para mostrar a ocorrência de uma falha.

Tab. 2.4 – Propriedades configuráveis da interface GPIB

ByteOrder: [{littleEndian} bigEndian]
BytesAvailableFcn
BytesAvailableFcnCount
BytesAvailableFcnMode: [{eosCharCode} byte]
ErrorFcn
InputBufferSize
Name
OutputBufferSize
OutputEmptyFcn
RecordDetail: [{compact} verbose]
RecordMode: [{overwrite} append index]
RecordName
Tag
Timeout
TimerFcn
TimerPeriod
UserData

Tab. 2.5 – Propriedades específicas da interface GPIB

BoardIndex
CompareBits
EOIMode: [{on} off]
EOSCharCode
EOSCharCode
EOSMode: [{none} read write read&write]
PrimaryAddress
SecondaryAddress

- **Arquivos de funções .m:** para se realizar qualquer tarefa na aplicação do instrumento de controle, deve-se chamar uma função .m do ambiente do MATLAB que, dentre outras funções, permite:
 - Criar objetos de instrumentos, que fornecem uma porta de entrada para as capacidades do instrumento e permitem controlar o comportamento da aplicação.
 - Conectar o objeto ao instrumento.
 - Configurar os valores das propriedades.
 - Escrever dados no instrumento e ler dados do instrumento.
 - Avaliar o estado da aplicação e examinar os recursos de hardware.
- **Adaptador de “driver” de interface** ou simplesmente **adaptador**: constitui a ligação entre o “toolbox” e o “driver” da interface (placa de GPIB) e sua principal função é passar informações entre esses dois. Os “drivers” de interface são fornecidos pelo fabricante do instrumento. Os adaptadores que o MATLAB suporta estão listados na Tab. 2.6.

Tab. 2.6 – Adaptadores de Interface que o MATLAB suporta

Nome da função	Adaptador
Agilent Techonologies	Agilent
Computer Boards	
Measurement Computing Co	cbi
Capital Equipment Co	cec
IOTech	iotech
Keithley	keithley
National Instrumens	ni

COMUNICAÇÃO COM O INSTRUMENTO

A comunicação com o instrumento pelo barramento GPIB, através do “Instrument Control Toolbox”, é realizada através das seguintes etapas:

1. Criar um objeto de instrumento
2. Conectar o objeto ao instrumento
3. Configurar os valores das propriedades
4. Escrita e leitura de dados
5. Desconectar

CRIAÇÃO DE UM OBJETO

(Funções de criação de objeto, ou construtores de objeto)

Objetos de instrumento são componentes do “toolbox” utilizados para acessar o instrumento. Eles fornecem uma porta de entrada para as funcionalidades do instrumento e permitem o controle do seu comportamento de aplicação.

Um objeto GPIB é criado com a função `gpib`.

```
OBJ = GPIB('VENDOR', BOARDINDEX, PRIMARYADDRESS);
```

Essa função tem como parâmetros de entrada o nome do adaptador ‘VENDOR’, o índice da placa GPIB BOARDINDEX e o endereço primário do instrumento PRIMMARYADDRESS.

CONECTANDO O OBJETO AO INSTRUMENTO

Antes que se possa utilizar o objeto do instrumento para escrever ou ler dados, é necessário que se conecte o objeto ao instrumento, cujo endereço ou porta foi definido pelo construtor.

Pode-se utilizar a função `fopen` para conectar um objeto ao instrumento.

```
FOPEN (OBJ);
```

CONFIGURAÇÃO DAS PROPRIEDADES / RETORNO DE INFORMAÇÕES DA PROPRIEDADE

Pode-se estabelecer o comportamento desejado do objeto de instrumento através da configuração dos valores de propriedades.

Pode-se configurar valores de propriedades utilizando a função `set`, a notação de ponto (“dot notation”), ou especificando os pares nome da propriedade/valores da propriedade no instante da criação do objeto.

função set:

```
set(OBJ, 'P1', 'V1');
```

“dot notation”:

```
g.P1 = 'V1';
```

criação do objeto:

```
g = gpib('VENDOR', BOARDINDEX, PRMARYADDRESS, 'P1', 'V1');
```

Para se obter informações das propriedades e seus respectivos valores atuais, pode-se utilizar o comando `get` ou a notação de ponto. No caso de o objeto já ter sido criado, a função `set` retorna, para um arquivo ou para a linha de comando, todas as funções configuráveis.

Os objetos de instrumento possuem dois tipos de propriedade: **propriedades básicas**, que podem ser utilizadas objetos de qualquer tipo de interface; e **propriedades específicas do objeto**, que podem ser utilizadas apenas por alguns tipos de objetos.

CONFIGURAÇÃO DO ENDEREÇO GPIB

Cada objeto GPIB é associado com um controlador e com um instrumento. O endereço GPIB consiste do índice da placa do controlador GPIB (BOARDINDEX) e dos endereços primário e (opcionalmente) secundário do instrumento

(PRIMARYADDRESS e SECONDARYADDRESS, respectivamente). O fabricante Agilent Technologies utiliza o termo “Unidade Lógica” (“logical unit”) para se referir ao índice da placa.

ESCRITA E LEITURA DE DADOS

A comunicação com o instrumento envolve a escrita e a leitura de dados.

Existem três pontos a serem considerados:

- Qual é o processo através do qual os dados fluem do MATLAB para o instrumento e do instrumento para o MATLAB?

O “Instrument Control toolbox” gerencia automaticamente os dados transferidos entre o MATLAB e o instrumento.

- Os dados a serem transferidos são binários (numéricos) ou texto (ASCII)?

Para muitos instrumentos, escrever dados no formato de texto significa escrever “strings” de comandos que alteram configurações do instrumento, preparam o instrumento para retornar dados ou informações de estado, e assim por diante. Escrever dados binários significa escrever valores numéricos no instrumento, como dados de calibração e de forma de onda.

- A função de escrita ou leitura irá bloquear o acesso à linha de comando do MATLAB?

É possível controlar-se o acesso à linha de comando do MATLAB especificando se a operação de escrita ou leitura é síncrona ou assíncrona. Uma operação síncrona bloqueia o acesso à linha de comando até que a função de escrita ou leitura termine sua execução. Uma operação assíncrona não bloqueia o acesso à linha de comando e é possível executar-se comandos adicionais enquanto a função de escrita ou leitura está sendo executada no “background”.

OPERAÇÕES DE ESCRITA

A Tab. 2.7 mostra as funções associadas à escrita de dados:

Tab. 2.7 – Funções de escrita de dados

Nome da função	Descrição
<code>Binclockwrite</code>	Escreve blocos de dados binários no instrumento
<code>Fprintf</code>	Escreve dados de texto no instrumento
<code>Fwrite</code>	Escreve dados binários no instrumento
<code>Stopasync</code>	Pára operações de leitura e escrita simultâneas

Uma operação de escrita, utilizando `fprintf` (escrita de texto) e `fwrite` (escrita de dados binários) é completada quando uma das duas seguintes condições for satisfeita:

- Os dados especificados são escritos;
- O tempo especificado na propriedade “Timeout” se esgota.
- Um instrumento determina se operação de escrita está completa baseada nos valores das propriedades `EOSMode`, `EOIMode` e `EOSCharCode`. Se a propriedade `EOSMode` é configurada para escrita ou escrita&leitura, cada ocorrência de “\n” no comando de texto é substituída pelo caractere de Fim-de-String (“End-of-String EOS”) especificado pelo valor de `EOSCharCode`. Portanto, quando se utiliza o formato do `fprintf` padrão, ou seja, “%s\n” todos os comandos de texto escritos no instrumento serão finalizados com aquele valor. O valor default de `EOSCharCode` é LF (“line feed caracter”). O EOS requerido por cada instrumento é descrito em sua documentação.

Se o `EOIMode` está “on”, então a linha Fim (“End”) ou Identidade (“Identity”) é afirmada quando o último byte é escrito no instrumento. O último byte pode ser parte de um dado binário ou de um dado com formato de texto. Se `EOSMode` está configurado para escrita ou escrita&leitura, então o último byte escrito é o valor de `EOSCharCode` e a linha EOI é afirmada quando o instrumento receber aquele byte.

OPERAÇÕES DE LEITURA

A Tab. 2.8 mostra as funções associadas à leitura de dados:

Tab. 2.8 – Funções de leitura de dados

Nome da função	Descrição
<code>Binblockread</code>	Leitura de bloco de dados binários
<code>Fgetl</code>	Leitura de uma linha de texto, descartando o terminador
<code>Fgets</code>	Leitura de uma linha de texto, incluindo o terminador
<code>Fread</code>	Leitura de dados binários
<code>Fscanf</code>	Leitura de dados, formatando-os em texto
<code>Readasync</code>	Leitura assíncrona de dados
<code>Scanstr</code>	Leitura de dados, formatando-os em texto, e verificação
<code>Stopasync</code>	Pára operações de leitura e escrita de dados

Uma operação de leitura com `fgetl`, `fgets`, `fread`, `fscanf` ou `readasync`, completa-se de uma das seguintes condições for satisfeita:

- A linha EOI é afirmada
- A terminação especificada pela propriedade `EOSCharCode` está pronta. Isso pode ocorrer apenas quando a propriedade `EOSMode` é configurada para leitura ou escrita&leitura.
- O tempo especificado na propriedade “Timeout” se esgota
- O número especificado de valores é lido (apenas para `fread`, `fscanf` e `readasync`)
- O “buffer” de entrada está cheio (se o número de valores não é especificado).

Além disso, pode-se interromper uma operação de leitura assíncrona a qualquer momento com a função `stopasync`.

DESCONECTAR

Quando o objeto criado não for mais necessário, deve-se desconectá-lo do instrumento, e remove-lo da memória e do workspace através das seguintes funções:

```

fclose (OBJ);
delete (OBJ);
clear OBJ;

```

2.6.4. Interface com o usuário

Para uma operação amigável e fácil de usar, a interface com o usuário foi feita de forma gráfica, que, no MATLAB, pode ser obtida através do guide (“Graphical Users Interface Development Environment”).

O guide fornece um conjunto de ferramentas para a criação de GUIs (“Graphical User Interfaces”). Essas ferramentas simplificam de maneira considerável o processo de arranjo dos objetos e a programação da interface.

A Fig. 2.10 mostra o “Layout Editor” do guide que constitui o painel de controle para todas as suas ferramentas.

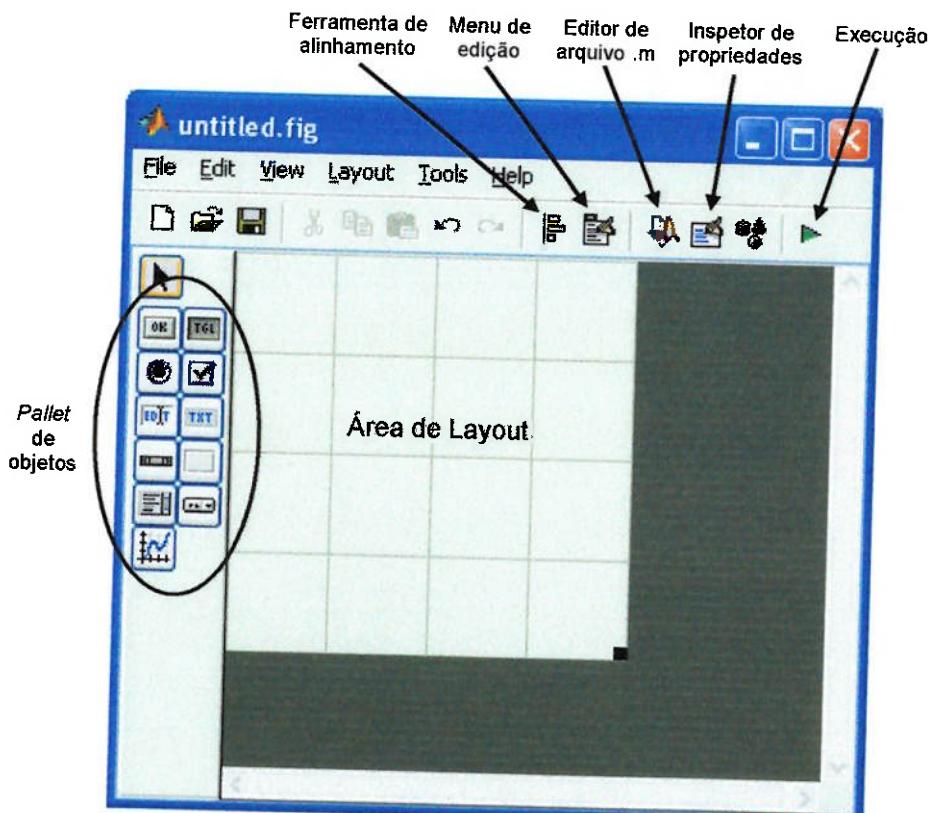


Fig. 2.10 – “Layout Editor”

Os objetos disponíveis para a interface são:

- “Push Buttons”: geram uma ação quando são clicados.
- “Toggle Buttons”: geram uma ação e indicam um estado binário (ligado ou desligado, por exemplo); quando são clicados, permanecem pressionados e a rotina de “callback” é executada quando o botão do mouse é liberado; um clique subsequente no mouse, retorna o “toggle button” para o estado anterior e uma outra rotina de “callback” é executada.
- “Check Boxes”: geram uma ação quando clicados e seus estados são: “checado” ou “não-checado”; são utilizados quando determina-se estado através de um conjunto de escolhas independentes.
- “Radio Buttons”: são similares aos “check boxes”, mas as escolhas são mutuamente exclusivas dentre si, ou seja, apenas uma escolha fica selecionada em um dado tempo; são ativados por um clique do mouse sobre o objeto e o display indica o estado da escolha.
- “Edit Text”: campos que permitem o usuário entrar ou modificar “Strings” de texto.
- “Static Text”: mostra linhas de texto, podendo ser utilizados como identificação de outros objetos; não têm rotinas de “callbacks” associadas.
- “Sliders”: aceitam entradas numéricas dentro de uma faixa específica e pré-configurada de valores; a seleção é realizada pelo movimento de uma barra de deslizamento, cuja posição indica o valor desejado.
- “Frames”: são caixas que englobam regiões de uma janela de figura; facilitam o entendimento da interface porque agrupam visualmente objetos relacionados; não têm rotinas de “callbacks” associados.
- “List Boxes”: mostram uma lista de ítems, permitindo o usuário selecionar um ou mais deles.
- “Pop-Up Menus”: mostra uma lista de opções quando clicado na seta à direita do menu.

- “Enabling or Disabling Controls”: permite controlar se um objeto deve responder a um clique do mouse; possui três estados:
 - “On” – o controle é operacional.
 - “Off” - o controle está desativado e aparece em cinza.
 - “Inactive” – o controle está desativado, mas não aparece em cinza.
- “Axes”: permitem a interface mostrar gráficos ou imagens.
- “Figure”: são janelas que contêm a interface desenvolvida no “Layout Editor”.

O arranjo dos objetos na interface é feita clicando-se sobre os objetos no “*pallet*”, à esquerda do “Layout Editor”, e arrastando-os para a área de layout. Eventualmente pode-se ajustar o alinhamento dos objetos utilizando a “Ferramenta de Alinhamento” conforme mostra a Fig. 2.10. As propriedades dos objetos, tais como tipo, nome, conteúdo, nome da rotina de “callback”, cor, posição na janela, dentre outras, também são ajustadas através de um duplo-clique sobre os mesmos ou pelo “Inspetor de Propriedades” (Fig. 2.10). Para cada tipo de objeto, uma janela semelhante à mostrada na Fig. 2.11 será aberta permitindo a edição de suas propriedades.

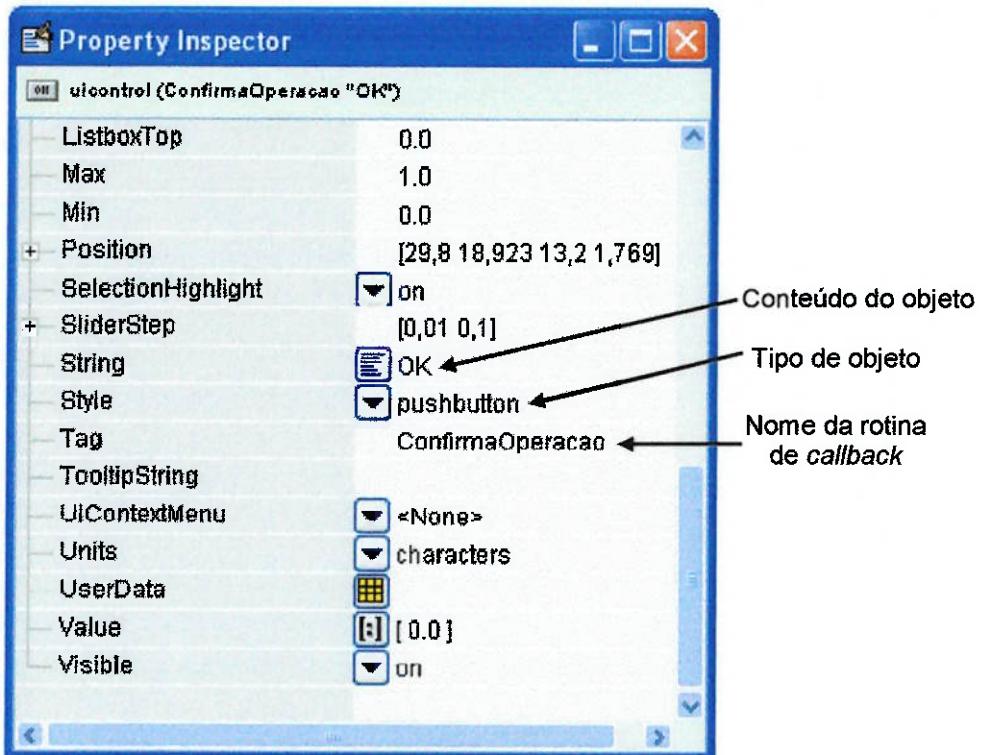


Fig. 2.11 – Inspetor de Propriedades

Após estabelecidos todos os objetos na janela e configuradas as suas propriedades, o “Editor de Arquivo .m” (Fig. 2.10) irá salvar a interface gráfica em um arquivo com extensão “.fig” e abrir o arquivo “.m” para a programação das funcionalidades da interface.

3. MATERIAIS E MÉTODOS

3.1. Simulações de MEMS

Inicialmente, tendo-se escolhido o MEMS eletrotermomecânico como objeto de estudo, fez-se as simulações dos mecanismos passíveis de fabricação usando o ANSYS, um programa que utiliza o método de elementos finitos, versão 6.1. As características mais importantes decorrentes das tensões elétricas impostas são: a corrente total fornecida (que não pode passar de um máximo permitido pela fonte DC), a distribuição de temperatura (que determina o deslocamento e não pode passar de um certo valor máximo, acima do qual a estrutura seria danificada) e os deslocamentos da estrutura. Também é interessante conhecer as tensões mecânicas máximas para verificar a viabilidade do deslocamento do mecanismo flexível.

3.1.1. Obtenção do Modelo para Simulação

Para a simulação de um mecanismo, deve-se ter um modelo matemático. Se houver um modelo CAD disponível, torna-se fácil obter um modelo para o programa de simulação que usa o método de elementos finitos, como o ANSYS, usado no presente trabalho. Assim, se o modelo CAD não existir, este pode ser obtido seguindo os passos abaixo, assumindo que exista o seu perfil bidimensional na literatura.

Uma figura com o perfil do mecanismo é digitalizada, sendo tratada em um programa de vetorização, transformando os contornos em vetores. Usando um programa de CAD, os vetores são transformados nas faces que compõem o perfil bidimensional do dispositivo, sendo em seguida extrudado. Os terminais elétricos e outras faces que ficarão engastadas no substrato terão uma altura um pouco maior na extrusão, modelando a diferença de espessura entre estes elementos e as partes móveis. Com o modelo tridimensional pronto, basta importá-lo para o programa de simulação por elementos finitos e iniciar as análises.

3.1.2. Garra (projeto otimizado)

Um dos possíveis mecanismos que poderíamos fabricar é uma garra com 3 graus de liberdade: movimentação no plano (direções X e Y) e fechamento da garra. Para esse mecanismo, foi feita uma série de simulações de modo a termos uma previsão do seu funcionamento. O modelo de garra utilizado foi obtido de Jonsmann (1999).

No mecanismo, existem terminais elétricos onde são impostas tensões elétricas, de modo que a corrente resultante aqueça a estrutura por efeito Joule e ocorra a expansão térmica do material. As características mais importantes decorrentes das tensões elétricas impostas são: a corrente total fornecida (que não pode passar de um máximo permitido pela fonte DC), a distribuição de temperatura (que determina o deslocamento e não pode passar de um certo valor máximo) e os deslocamentos da estrutura. Também é interessante conhecer as tensões mecânicas máximas para verificar a viabilidade do deslocamento do mecanismo flexível.

O terminal U_{gnd} é sempre conectado ao valor de referência (terra). Nos terminais U_x , U_y e U_g , temos os locais onde podemos impor as tensões elétricas para obter a movimentação horizontal X, vertical Y (no plano de trabalho) e atuação da garra (fechamento).

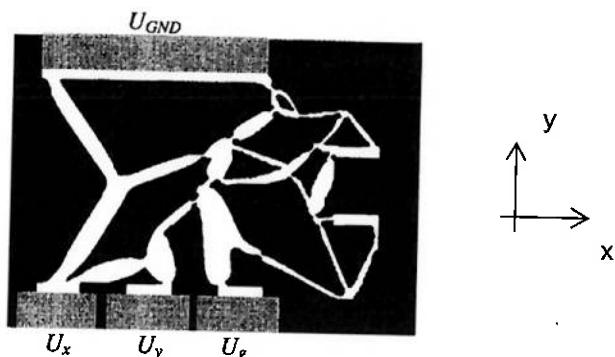


Fig. 3.1 - Perfil da garra obtida por projeto otimizado

Foram, então, impostas tensões de 0,5V em U_x , U_y e U_g separadamente para ver os efeitos isolados.

O tamanho da garra foi consequência da metodologia de fabricação que foi empregada. Para a confecção das máscaras usadas no processo, foi empregado um fotolito comum, cuja resolução permitiria dimensões mínimas de 50 µm. Assim, a menor largura no perfil do mecanismo flexível não pode ser inferior a esse valor, o que levou a aumentá-lo proporcionalmente nas duas dimensões até que o valor mínimo de largura fosse atingido. Assim, a parte suspensa (não fixada no substrato) do mecanismo simulado e passível de fabricação tem 11,6mm de largura por 7,4mm de altura e 70µm de espessura.

Nas áreas dos terminais em contato com o substrato, foram impostas temperaturas de 293K (20°C) e foram consideradas engastadas. Nas áreas das partes móveis, foi definido uma convecção com coeficiente de película de 5000W/m²K e temperatura ambiente de 293K (20°C).

Usando o programa ANSYS, fez-se as simulações usando o elemento SOLID98 e análise dos tipos estrutural, térmica e elétrica, todas estáticas (sem considerar transiente). Foram considerados pequenos deslocamentos.

As propriedades do material do mecanismo foram obtidas de experimentos desenvolvidos no Laboratório Nacional de Luz Síncrotron (LNLS), em Campinas (SP):

Tab. 3.1 - Propriedades do níquel eletrodepositado (garra)

Densidade	8890 kg/m ³
Módulo de Elasticidade	188 Gpa
Coeficiente de Poisson	0,31
Coeficiente de Expansão Térmica	13,4 µm/m·°C
Condutividade Térmica	90,7 W/mK
Resistividade Elétrica	$7,2 \times 10^{-7} \Omega \cdot \text{cm}$

3.1.3. Mesa XY (projeto intuitivo)

Uma alternativa para a garra obtida por otimização topológica é a mesa XY, obtida por projeto intuitivo. Possui duas estruturas similares para a atuação no sentido X e Y. A atuação, assim como no caso da garra, é baseada no efeito de expansão térmica do material da estrutura quando esta é aquecida por efeito Joule.

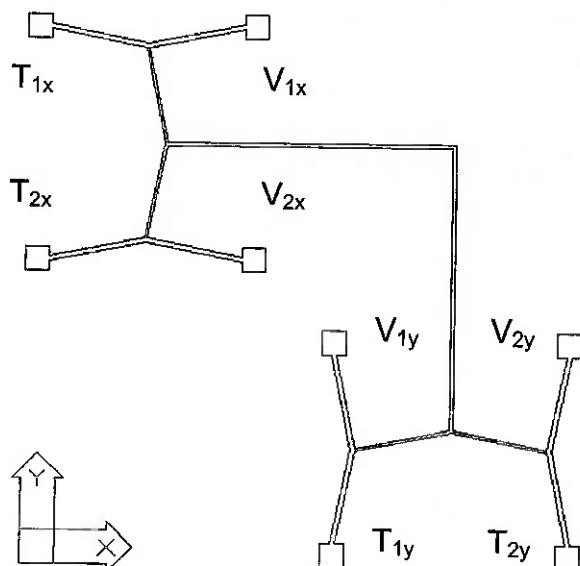


Fig. 3.2 - Desenho CAD Bidimensional da Mesa XY

Nos pares de terminais $V_{ix}-T_{ix}$ e $V_{iy}-T_{iy}$, vistos na Fig. 3.1.3.1 acima, são aplicadas as diferenças de potencial elétrico para induzir corrente entre eles, de modo a aquecer os e expandi-los. A combinação das duas atuações simultâneas de uma das estruturas, seja X ou Y, gera um deslocamento perpendicular amplificado da haste mais longa na direção desejada.

A mesa XY mede 2,7mm de largura por 2,5mm de altura (área delimitada pelos terminais), sem contar a haste (3mm de comprimento). O mecanismo possui espessura de 45 μ m e é preso ao substrato de alumina pelos terminais, com 58 μ m de espessura, o que deixa uma distância de 13 μ m entre o mecanismo e o substrato.

Para a simulação do mecanismo, foi levada em consideração a simetria do mecanismo, o que leva a separá-lo em duas partes iguais e analisar apenas uma delas. Assim, a modelagem em ANSYS ficou sendo:

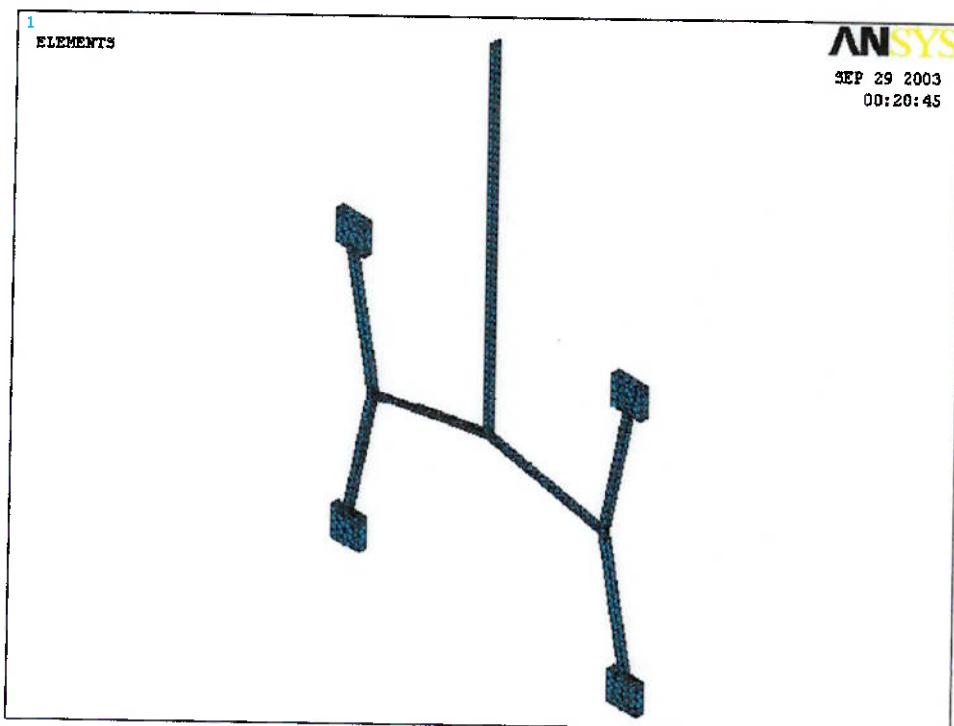


Fig. 3.3 - Modelo tridimensional da Mesa XY em ANSYS

Como no caso da garra, foi usado o elemento SOLID98 e análise estrutural, elétrica e térmica, todas estáticas, sem considerar transientes.

As propriedades do níquel consideradas para a simulação foram:

Tab. 3.2 - Algumas propriedades do níquel eletrodepositado (mesa XY)

Densidade	8890 kg/m ³
Módulo de Elasticidade	188 GPa
Coeficiente de Poisson	0,31
Coeficiente de Expansão Térmica	$1,34 \times 10^{-5}$ /K

Tab. 3.3 - Temperatura e resistividade elétrica do níquel eletrodepositado (mesa XY)

Temperatura (K)	Resistividade ($\Omega \cdot m$)
293	$1,71 \times 10^{-6}$
298	$9,31 \times 10^{-7}$
500	$9,49 \times 10^{-7}$

Tab. 3.4 - Temperatura e condutividade térmica do níquel eletrodepositado (mesa XY)

Temperatura (K)	Condutividade Térmica (W/m·K)
250	97,5
300	90,7
350	85,0
400	80,2
500	72,2
600	65,6
800	67,9
1000	71,8
1200	76,2
1400	80,4

Estas novas propriedades para o níquel foram retiradas de novos experimentos conduzidos no Laboratório Nacional de Luz Síncrotron (LNLS) em Campinas (SP) com mecanismos projetados especialmente para essas caracterizações.

Como condições de contorno, foram definidos deslocamentos nulos nos três eixos (X, Y e Z) e temperatura de 293K para as áreas do mecanismo fixos no substrato. Em todas as áreas expostas ao ambiente, foi imposto um coeficiente de convecção de 33000W/m²·K (com temperatura ambiente de 293K). As únicas áreas em que não há convecção são as de contato com o substrato e entre as duas estruturas na ponta da haste (na linha de simetria). Para gerar a atuação da estrutura, foi imposta uma tensão elétrica de 2V nos terminais V, sendo o “terra” ligado aos T. As tensões

elétricas e os “terrás” foram impostos em nós no meio dos terminais, para simular o uso das pontas de prova durante a atuação pela fonte controlada.

3.2. Comando de MEMS

Esta seção apresenta os métodos utilizados para a construção do adaptador de joystick e a interface gráfica com o usuário, e apresenta também o algoritmo do software de comando do microatuador.

3.2.1. Construção de um Adaptador de “Joystick”

Este tópico tem como objetivo apresentar os passos para a construção do Joy.dll, ou seja, a biblioteca do adaptador para o “Game Port” ao qual será conectado o “joystick”. Isso será realizado modificando-se os arquivos do adaptador de demonstração do “Data Acquisition Toolbox Adaptor Kit” (subpasta Demo) para criar a interface com o “joystick”, seguindo os passos abaixo.

Passo 1 – Nomear o novo adaptador

Para distinguir os arquivos do novo adaptador dos arquivos do adaptador de demonstração, os prefixos ‘**demo**’ nos arquivos e classes listadas na Tab. 2.2 devem ser renomeados utilizando-se o prefixo ‘**joy**’. A Tab. 3.5 lista os arquivos e as modificações que devem ser realizadas.

Tab. 3.5 – Modificação dos nomes dos arquivos

Adaptador de Demonstração	Alterações para o novo adaptador
Demoadapt.cpp	Joyadapt.cpp
Demoin.cpp	Joyin.cpp
Demo.cpp	Joy.cpp
Demo.def	Joy.def
Demo.idl	Joy.idl
Demo.rc	Joy.rc
Demoadapt.h	Joyadapt.h
Demoin.h	Joyin.h

Passo 2: Susbtuir os UUIDs

Este passo descreve as modificações fundamentais que permitem o adaptador receber os dados do “joystick”. Os quatro UUIDs (“Universal Unique Identifications”) no arquivo joy.idl devem ser substituídos manualmente por outros, gerados pelo GUIDGEN.EXE (Microsoft Visual C++ 6.0 Tool). Todas as seções que devem ser modificadas estão marcadas com comentários ‘TO_DO’, como mostra o seguinte exemplo de código (o nome do arquivo é seguido pelo número da linha entre parênteses).

Joy.idl(14) :

```
//TO_DO: Select new uuid for your own analog input interface.  
Use GUIDgen.
```

Joy.idl(25) :

```
//TO_DO: Select new uuid for your own adaptor type library.  
Use GUIDgen.
```

Joy.idl(36) :

```
//TO_DO: Select new uuid for your own adaptor class.  
Use GUIDgen.
```

Joy.idl(46) :

```
//TO_DO: Select new uuid for your own analog input class.  
Use GUIDgen.
```

Estas modificações permitem às classes, interface e biblioteca serem identificadas unicamente. Por exemplo, o UUID original da interface `IDemoIn` no primeiro ‘TO-DO’:

(CE932328-3BD9-11D4-A584-00902757EA8D)

deve ser modificado para o seguinte UUID da interface `IJoyIn`:

(61ABE1E1-AEAD-11d4-BAAF-00D0B7BAD1D3).

Passo 3: Fornecer informações de hardware do dispositivo

Para fornecer as informações de hardware corretas para o novo adaptador, as informações no método `AdaptorInfo` devem ser modificadas e as informações do método `OpenDevice` devem ser verificadas.

O método `AdaptorInfo` na classe `CJoyadapt` é modificada para retornar o número correto de “joysticks” conectados. O `Cdemoadapt` original possui algumas linhas marcadas com ‘bogus’ no primeiro ‘TO_DO’ (linha 106 do `Joyadapt.cpp` e `Demoadapt.cpp`) que precisam ser modificadas para retornar a informação atual sobre o “hardware”. Uma chamada à plataforma SDK API específica da Microsoft é adicionada para retornar esta informação.

```

if((wNumDevs=joyGetNumDevs())==0)
    return E_FAIL; //ERR_NODRIVER;

bDev1Attached=(joyGetPos(JOYSTICKID1,&joyinfo)!=JOYERR_UNPLUGGED);
bDev2Attached=((wNumDevs==2)&&(joyGetPos(JOYSTICKID2,&joyinfo)!=
JOYERR_UNPLUGGED));

if(bDev1Attached||bDev2Attached) // decide which joystick to use
    wDeviceID=bDev1Attached?JOYSTICKID1:JOYSTICKID2;
else
    return E_FAIL; // ERR_NODEVICE;
short len=wNumDevs; // Got it from HW

```

No segundo ‘TO_DO’ (linha 143 do `Demoadapt.cpp` e 160 do `Joyadapt.cpp`) no mesmo método `AdaptorInfo`, o código do adaptador de demonstração possui uma variável `ai` com o valor `true`. Esta variável indica que o “Game Port” é uma entrada analógica. Não é necessário modificar esta seção no `AdaptorInfo` porque a porta do “joystick” é utilizada como uma entrada analógica.

```
Bool ai=true; // HW is analog input
```

O método `OpenDevice` na mesma classe `Cjoydapt` tem a seção da saída analógica e as seções das entradas e saídas digitais ignoradas, sendo consideradas como um comentário do programa, e não é necessário fazer nenhuma modificação porque o “joystick” está sendo usado apenas como entrada analógica.

Passo 4: Inicializar variáveis “private”, adicionar uma chamada de “hardware”

No método `Open`, as informações sobre o joystick conectado e verificações de erros devem ser adicionadas. O número pré-definido de “joysticks” conectados é 1 (`wNumDevs=1`). Depois, a variável `private m_uIJoyDevInUse`, no método `Open`, deve ser inicializada para ser utilizada posteriormente pelo método `GetSingleValue` como um argumento para a chamada API `joyGetPosEx` do “joystick”. O método `GetSingleValue` adiciona a chamada API para receber a posição do “joystick” e também normaliza os dados para um campo de números simétricos.

```
m_uJoyNumOfDev = wNumDevs;
...
JOYINFOEX stJI;
stJI.dwSize = sizeof(stJI);
stJI.dwFlags = JOY_RETURNALL;
if (joyGetPosEx( m_uIdJoyDevInUse, (LPJOYINFOEX) &stJI ) != JOYERR_NOERROR )
    return E_FAIL;
```

Não há necessidade de se inicializar o hardware para o “Game Port” ou para o “joystick”. Portanto, o código do construtor `Cjoyin` permanece inalterado como no construtor original `Cdemo1` onde a inicialização para o “hardware” foi feita. Pode-se, no entanto, inicializar o “buffer” no construtor se se deseja armazenar os pontos de dados.

Passo 5 – Verificar os nomes dos arquivos nas macros

O nome da classe e os identificadores de programa na macro `DECLARE_REGISTRY` em `JoyAdapt.h` devem ter o prefixo ‘**joy**’ apropriado, ao invés de ‘**demo**’. Essas modificações foram realizadas no Passo 1 quando todas as classes, arquivos e identificadores com o nome ‘**demo**’ foram substituídas por ‘**joy**’.

```
DECLARE_REGISTRY(CJoyadapt, _T("Joy.Joyadapt.1"),
_T("Joy.Joyadapt"), IDS_PROJNAME, THREADFLAGS_BOTH )
```

Passo 6: Compilar e construir o adaptador

O projeto deve ser compilado e construído para produzir o adaptador **Joy.dll**, utilizando, por exemplo, o Visual C++.

Passo 7: Instalar o adaptador

O adaptador pode ser instalado no sistema de duas maneiras:

- `Regsvr32.exe ($pathname) \Joy.dll`
onde `$pathname` é o caminho do diretório onde se encontra o adaptador
- `daqregister('Joy.dll')`

3.2.2. Algoritmo do “software” de comando

O algoritmo para o software de comando do microatuador é mostrado na Fig. 3.4.

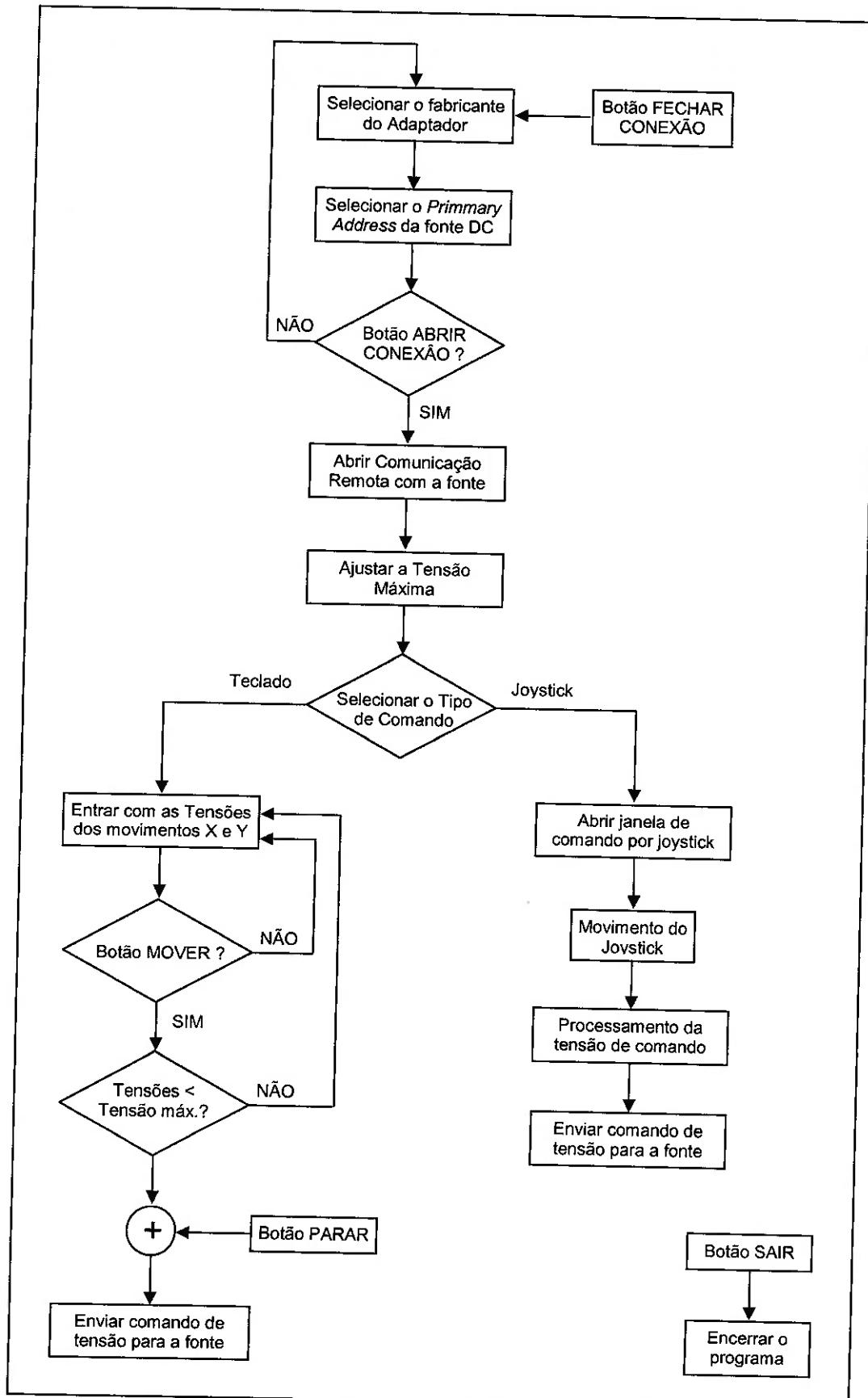


Fig. 3.4 – Algoritmo do software de comando do microatuador

O fluxo de dados é iniciado com a configuração do barramento GPIB, definindo-se o fabricante da placa GPIB (tipo PCI) conectada ao computador, o que define o adaptador a ser utilizado no objeto de instrumento criado pelo MATLAB; e definindo-se o “Primary Address” da fonte DC no barramento, cujo valor deve ser igual ao ajustado manualmente na mesma. A comunicação remota com a fonte DC é aberta clicando-se no botão “ABRIR CONEXÃO”. A partir desse momento, todo o comando da fonte DC é realizado pelo computador.

Em seguida, deve-se estabelecer o valor de tensão máxima à qual o microatuador pode ser submetido. Esse valor de tensão serve para dois propósitos: em primeiro lugar, previne entradas errôneas de tensão para os movimentos através do teclado; e em segundo lugar, o movimento comandado pelo “joystick” é associado a porcentagens dessa tensão máxima.

Finalmente, deve-se selecionar o tipo de comando desejado, teclado ou “joystick”. No primeiro caso, o usuário deve entrar com os valores de tensão para os movimentos nas duas direções (X e Y) e clicar no botão “MOVER” para enviar os dados à fonte DC, o que será realizado se os valores forem menores ou iguais à tensão máxima ajustada. O botão “PARAR” deve enviar tensão nula para os dois movimentos. No caso do comando através do “joystick”, uma tela que mostra o espaço de trabalho do microatuador deve aparecer e os dados devem ser enviados à fonte quando o “joystick” é movimentado.

O código-fonte implementado para esse algoritmo encontra-se no Anexo A.

3.2.3. Construção da interface com o usuário

A interface com o usuário foi obtida utilizando-se a ferramenta guide, conforme mencionado na seção 2.6.4. A Fig. 3.5 mostra os objetos que compõem a interface desenvolvida.

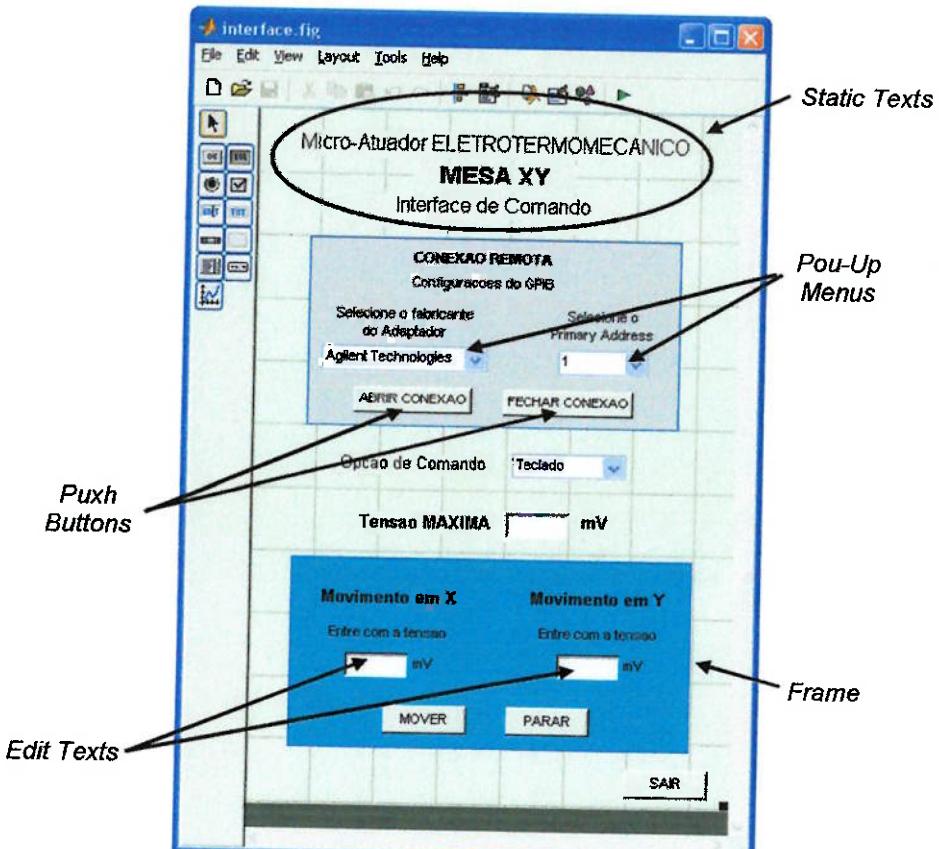


Fig. 3.5 – Objetos da interface gráfica desenvolvida

3.3. Experimento com a Mesa XY

Os experimentos com os mecanismos fabricados foram realizados para verificar os dados obtidos através das simulações e para o teste do programa de comando da fonte DC. A mesa XY foi fabricada conforme o processo descrito na seção 2.5.

A fabricação e testes foram feitos no Laboratório Nacional de Luz Síncrotron (LNLS) em Campinas (SP), dado que em seu Laboratório de Microfabricação existem todos os meios para a obtenção dos deslocamentos realizados pela mesa XY, usando-se uma câmera SONY CCD NTSC acoplada à estação microprovadora Microtech-CASCADE RH-4100, com aumentos de 100 \times /200 \times /400 \times (Fig. 3.6).

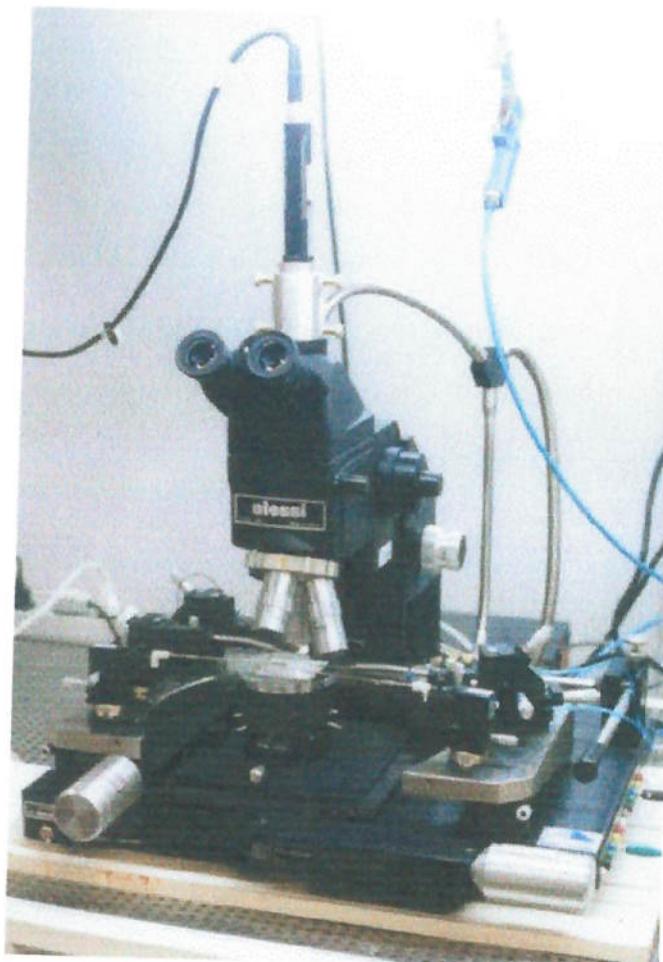


Fig. 3.6 – Microscópio com Câmera CCD acoplada

O substrato com o mecanismo é fixado à estação microprovadora através de vácuo e pontas de prova de tungstênio são usadas para a aplicação, nos terminais da estrutura, da diferença de potencial gerada pela fonte DC (modelo Agilent E3649A), que informa a corrente fornecida no processo. As pontas de prova, seus suportes e o substrato fixados podem ser vistos na Fig. 3.7.

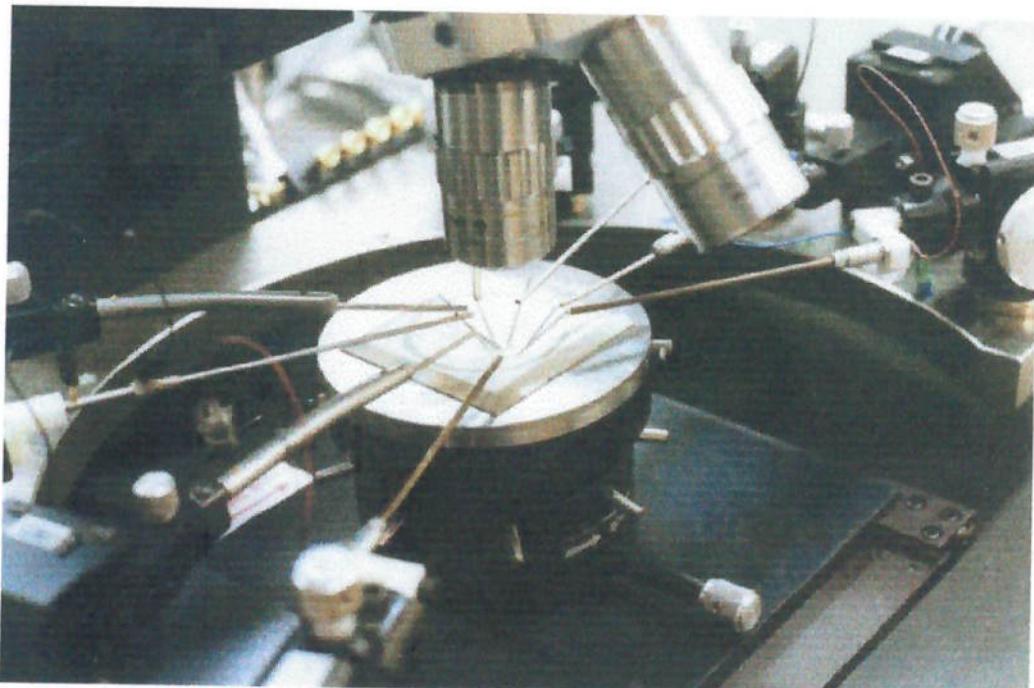


Fig. 3.7 - Substrato fixado ao microscópio e com as pontas de prova posicionadas

Para obtenção de imagens que necessitem de aumentos menores, pode ser usado um microscópio Mitutoyo com lentes de 2 \times /10 \times /20 \times , sem qualquer recurso de fixação do substrato.

Poderia ser usada uma outra fonte DC (Advantest TR6162), mas esta tem apenas uma saída de tensão, o que inviabilizaria os testes da mesa XY, que requer duas. Apesar disso, esta fonte foi importante para a verificação da simulação, já que a fonte Agilent não consegue fornecer a corrente requerida, já que seu limite é de 1,4A.

As imagens da mesa XY foram capturadas e analisadas em um microcomputador com a placa Matrox Meteor II e o programa Matrox Inspector 3.1, que permitia determinar os deslocamentos quando ocorria a atuação. Os demais dados (distribuição de temperatura e tensões mecânicas) não podem ser determinados experimentalmente com os equipamentos disponíveis.

O aparato usado para os testes do comando da fonte DC por “joystick” pode ser visto na Fig. 3.8:

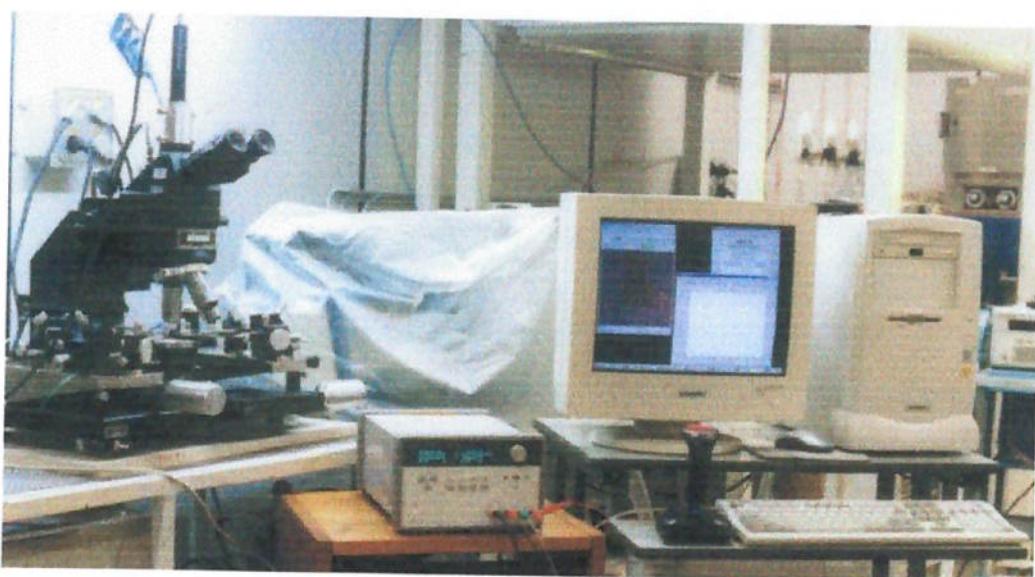


Fig. 3.8 - Equipamento usado no comando da fonte DC por “joystick”

Além do programa de captura e análise de imagens, foi necessário o programa Matlab 6 para a execução do programa feito para o monitoramento do “joystick” e comando da fonte DC.

3.4. Simulação Revisada da Mesa XY

Com o experimento, foram percebidos alguns equívocos na modelagem do mecanismo e algumas diferenças não esperadas no seu comportamento. Assim, de forma a verificar alguns aspectos não identificáveis experimentalmente, a simulação da mesa XY teve seu modelo corrigido.

O modelo, desta vez, comprehende todo o mecanismo, com as estruturas responsáveis pela atuação em ambos os eixos. Como no experimento, os terminais T_{1x} , T_{2x} , T_{1y} e T_{2y} permanecerão sempre aterrados, independente do eixo atuado. Para se aproximar mais do mecanismo fabricado, os terminais terão a mesma espessura que a porção suspensa.

Foram feitas duas análises em separado: uma envolvendo aspectos elétricos e térmicos e outra estrutural. A análise elétrica e térmica usou o elemento SOLID98 e teve como resultado a distribuição de temperatura. A análise estrutural usou o elemento SOLID92, considerou grandes deslocamentos e teve como parte das condições de contorno o resultado da primeira análise.

O modelo em ANSYS é mostrado na Fig. 3.9, com um dos terminais detalhados na Fig. 3.10.

Todas as outras características, como tamanho do atuador de cada eixo, propriedades do níquel eletrodepositado e demais condições de contorno, foram mantidas.

A simulação foi feita atuando-se em apenas um dos eixos, neste caso o X, ou seja aplicando-se tensão elétrica nos terminais V_{1x} e V_{2x} .

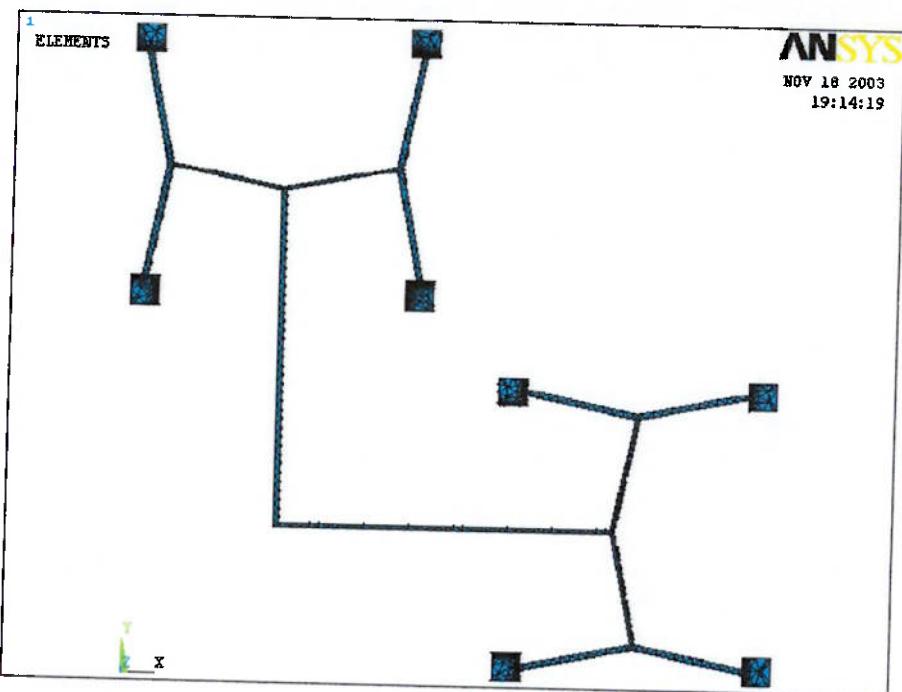


Fig. 3.9 - Modelo Revisado da Mesa XY em ANSYS

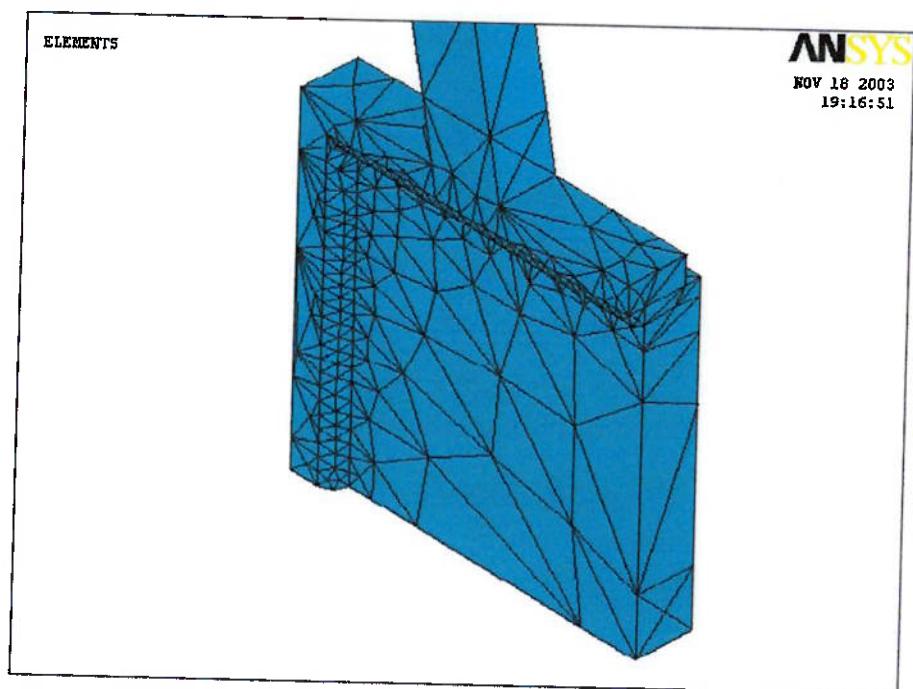


Fig. 3.10 - Modelo Revisado da Mesa XY em ANSYS (detalhe de um dos terminais)

4. RESULTADOS

4.1. Simulações de MEMS

4.1.1. Garra (projeto otimizado)

Para a tensão em U_x , temos:

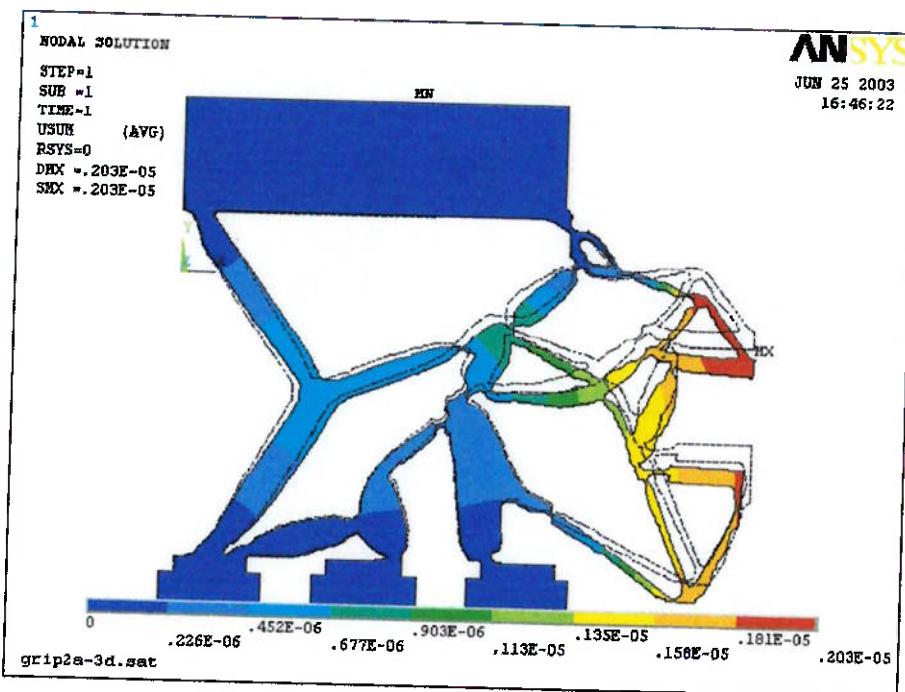


Fig. 4.1 - Simulação da Garra: Deslocamento Total (tensão em U_x)

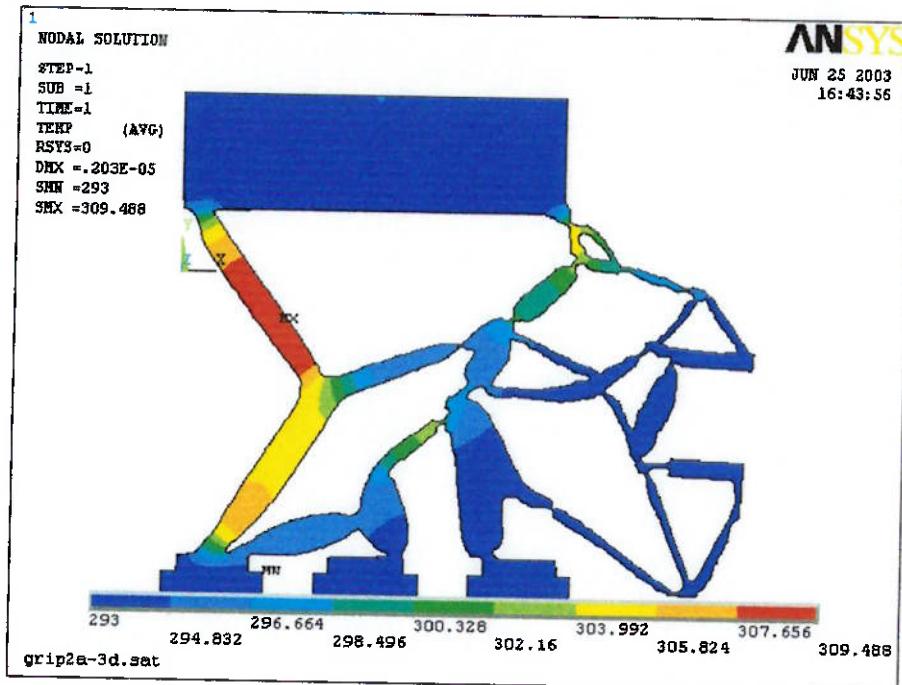


Fig. 4.2 - Simulação da Garra: Distribuição de Temperatura (tensão em U_x)

Para a tensão em U_y , temos:

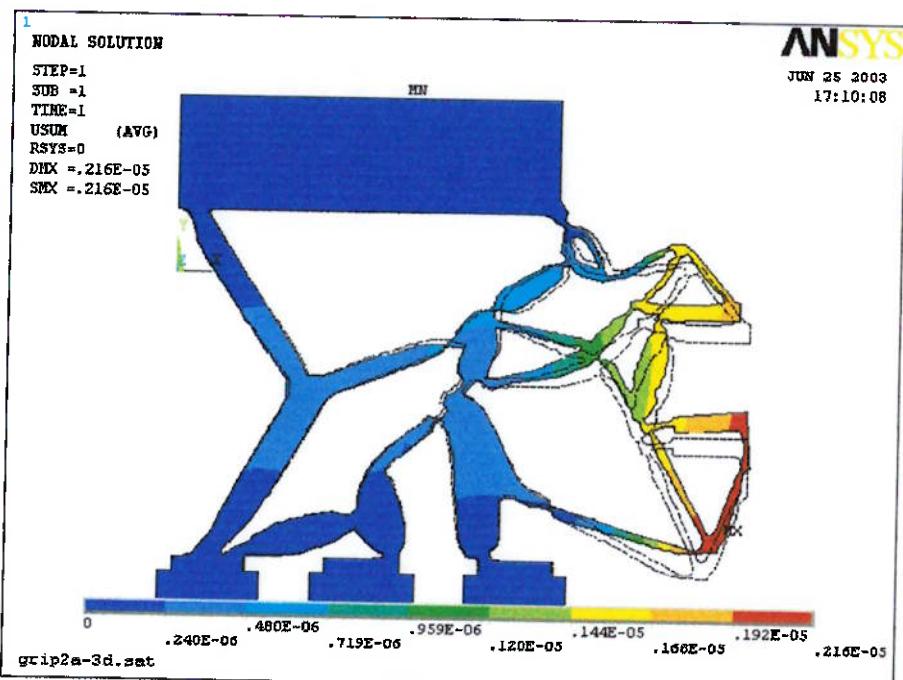


Fig. 4.3 - Simulação da Garra: Deslocamento Total (tensão em U_y)

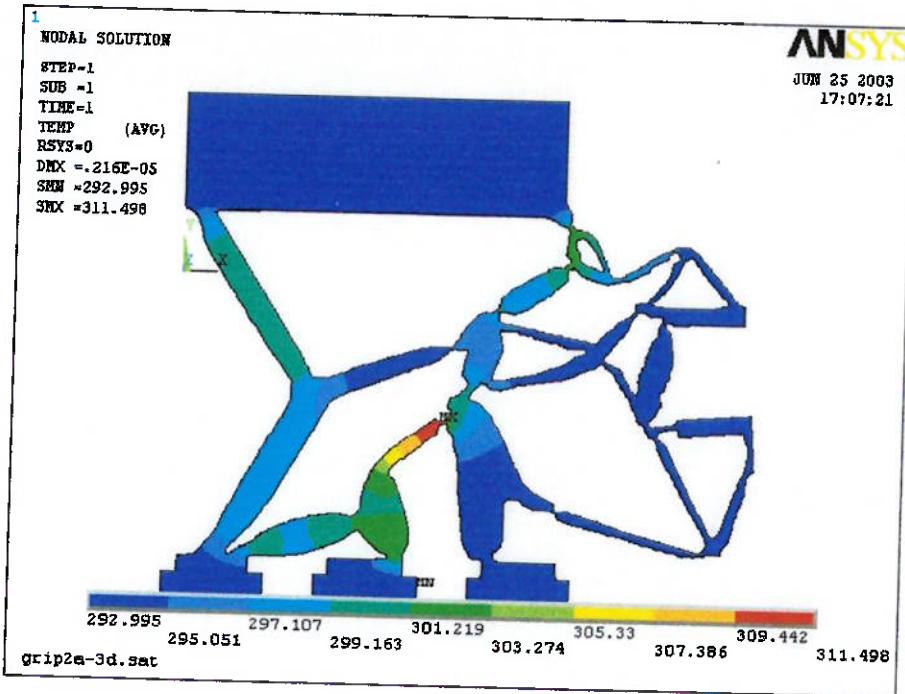


Fig. 4.4 - Simulação da Garra: Distribuição de Temperatura (tensão em U_y)

Para a tensão em U_g , temos:

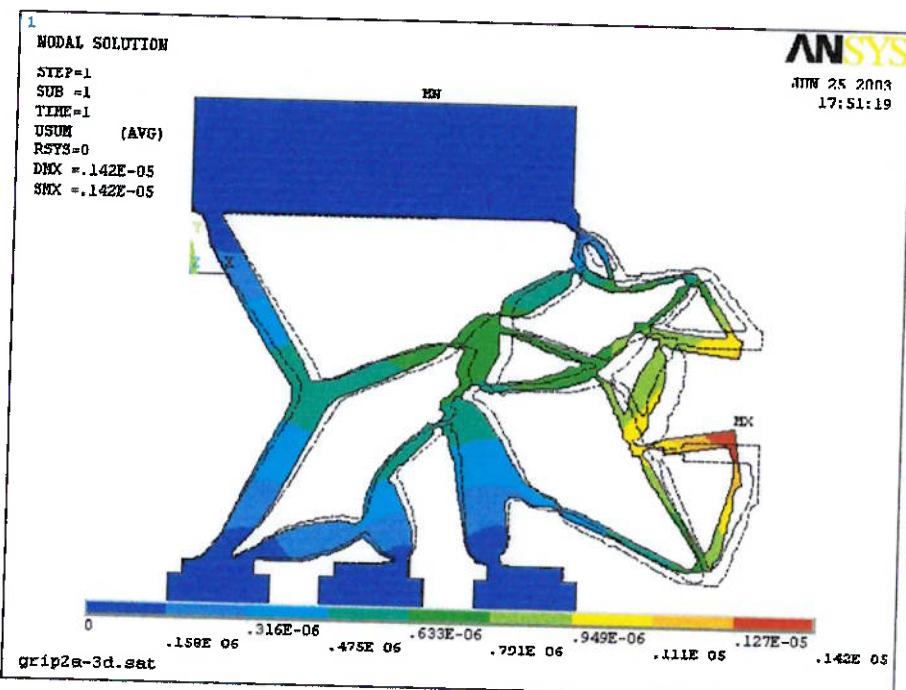


Fig. 4.5 - Simulação da Garra: Deslocamento Total (tensão em U_g)

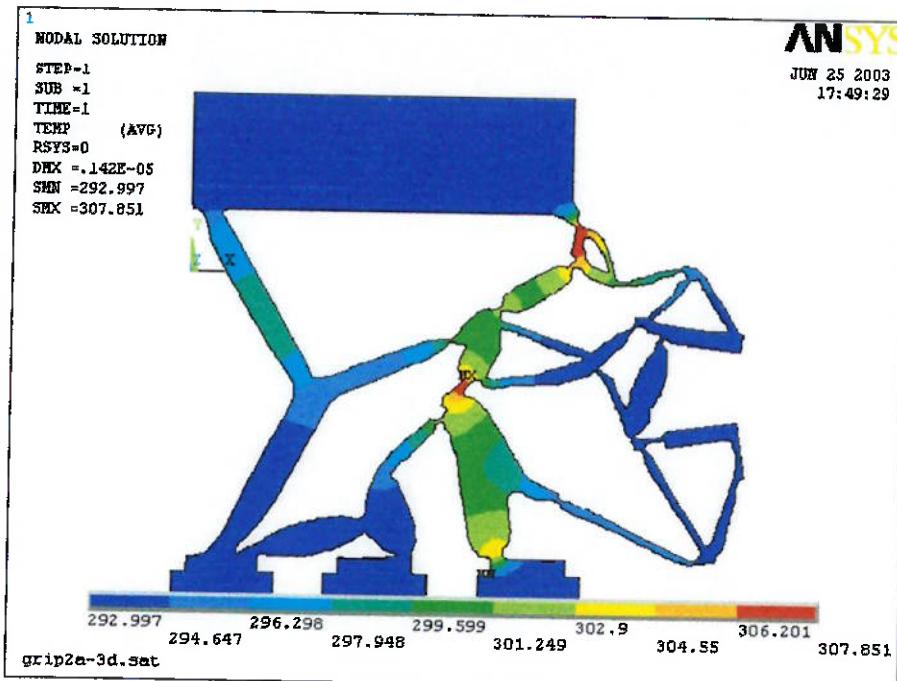


Fig. 4.6 - Simulação da Garra: Distribuição de Temperatura (tensão em U_g)

As correntes, temperaturas máximas e tensões máximas para cada um dos casos são:

Tab. 4.1 - Alguns resultados da simulação (garra)

Terminal com Tensão	Corrente (A)	Temperatura Máxima (K)	Tensão Máxima (Von Mises) (MPa)
U_x	3,3998	309,488	85,8
U_y	2,6504	311,498	92,7
U_g	2,6155	307,851	92,8

4.1.2. Mesa XY (projeto intuitivo)

Obtivemos uma corrente, em cada par de terminais, de aproximadamente 2,1A.

Outros resultados foram:

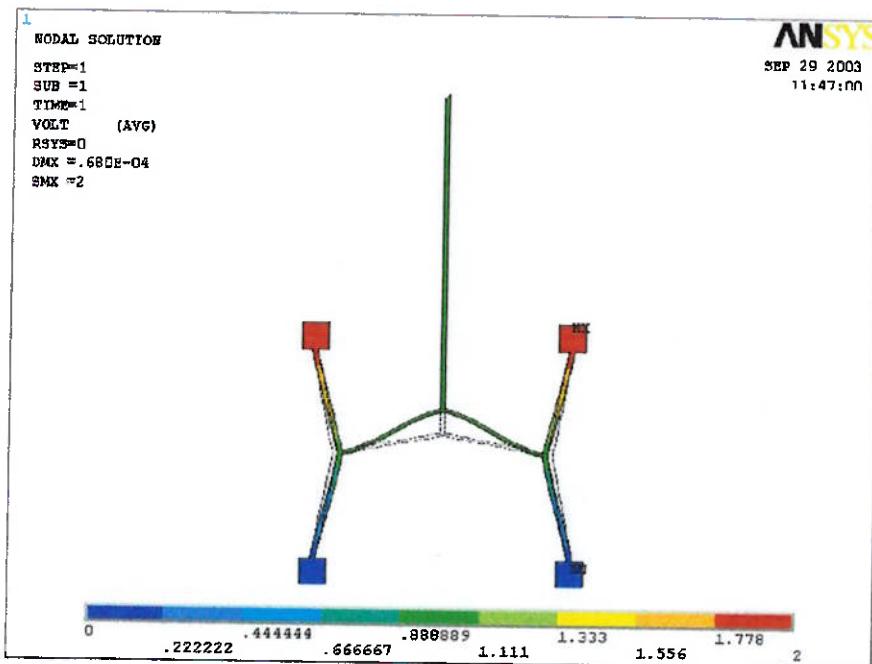


Fig. 4.7 - Simulação da Mesa XY: Distribuição de Tensão Elétrica

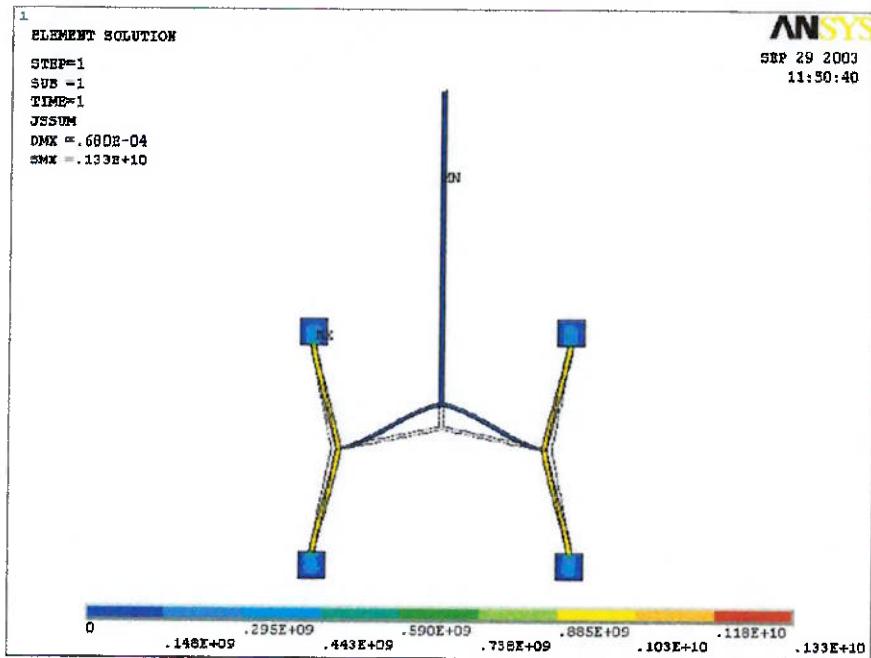


Fig. 4.8 - Simulação da Mesa XY: Densidade de Corrente

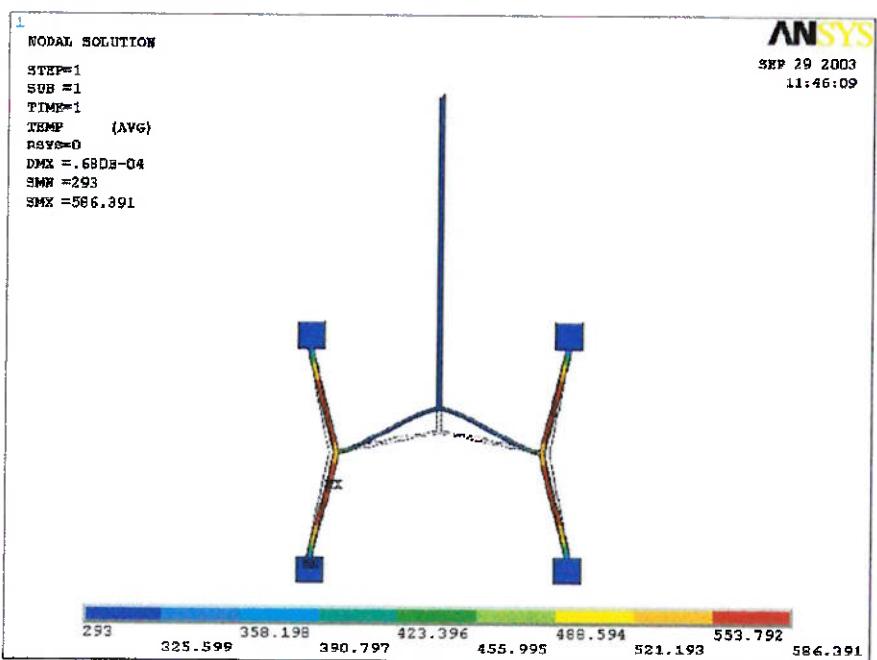


Fig. 4.9 - Simulação da Mesa XY: Distribuição de Temperatura

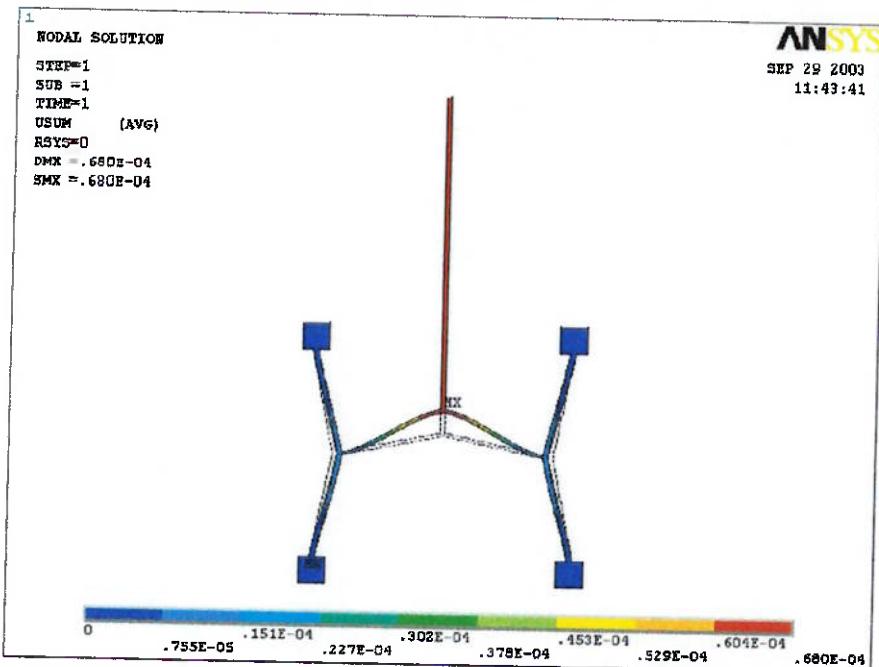


Fig. 4.10 - Simulação da Mesa XY: Deslocamento

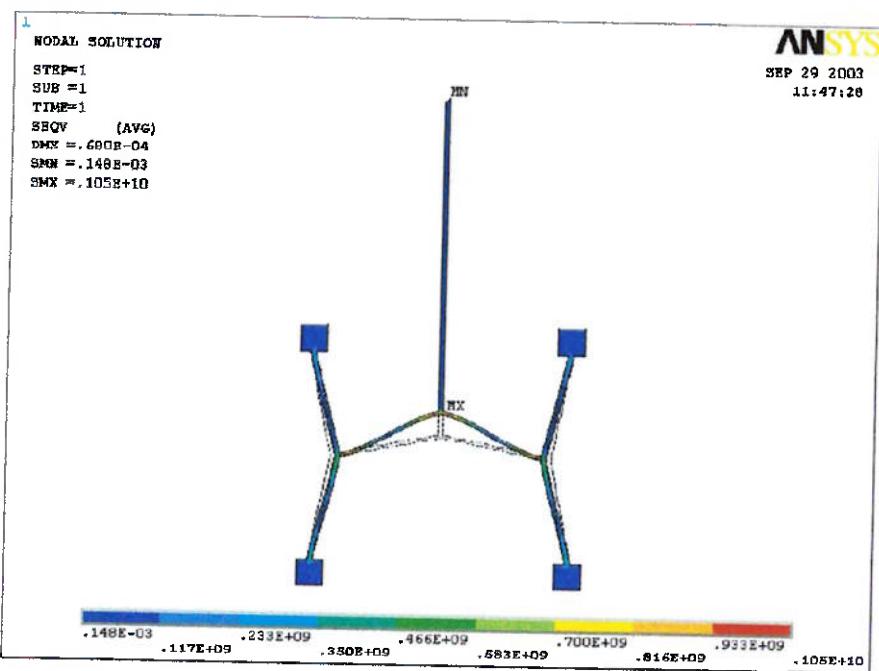


Fig. 4.11 - Simulação da Mesa XY: Tensão Mecânica (Von Mises)

4.2. Interface do Comando Remoto da Fonte DC

A Fig. 4.12 ilustra a interface gráfica construída no MATLAB utilizando a ferramenta guide, para uma operação amigável e fácil de usar pelo usuário.

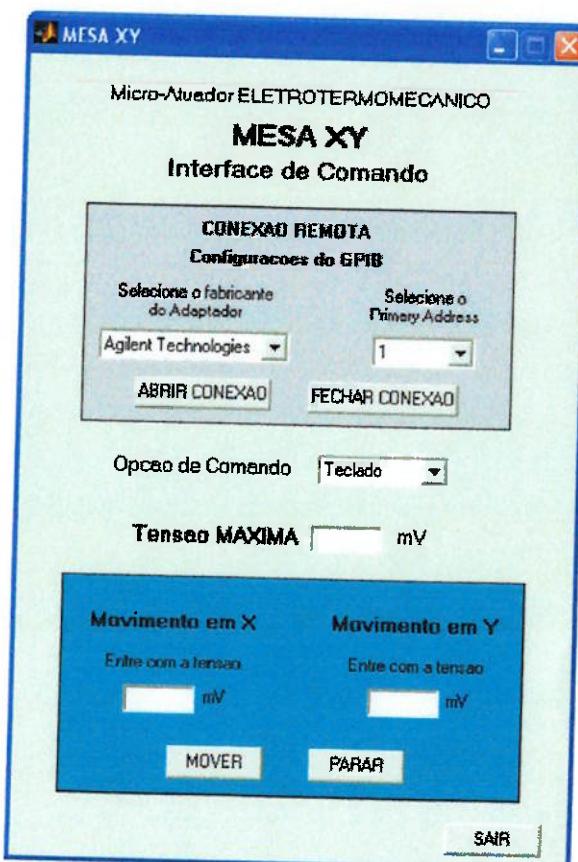


Fig. 4.12 – Interface com o usuário

Primeiramente, o usuário deve definir as CONFIGURAÇÕES DA INTERFACE GPIB (Fig. 4.13): o fabricante da placa GPIB que está conectada à placa mãe do computador, e o endereço primário, cujo valor deve ser igual ao ajustado manualmente na fonte DC. Para abrir a conexão remota com a fonte DC, deve-se pressionar o botão ABRIR.



Fig. 4.13 – Configurações da interface GPIB

O usuário deve selecionar o TIPO DE COMANDO, se TECLADO ou “JOYSTICK”. Para ambos os casos, a TENSÃO MÁXIMA permitida para o comando deve ser informada a fim de se evitar falhas por sobre-tensão da fonte, e do microatuador. Os campos de seleção do TIPO DE COMANDO e TENSÃO MÁXIMA estão mostrados na Fig. 4.14.

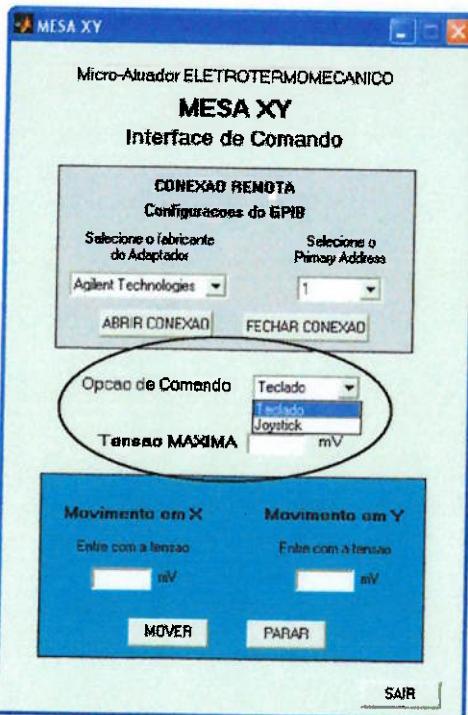


Fig. 4.14 – Seleção do TIPO DE COMANDO e TENSÃO MÁXIMA

Para o comando via TECLADO, o usuário deve entrar com as tensões desejadas para os movimentos dos eixos X e Y nos campos indicados e pressionar o botão MOVER. Caso alguma dessas tensões seja superior à tensão máxima permitida, a seguinte mensagem de erro será mostrada (Fig. 4.15):

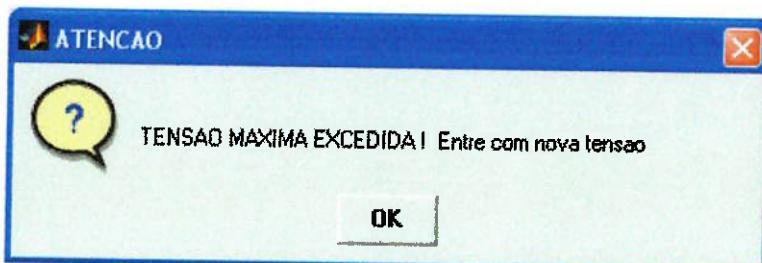


Fig. 4.15 – Mensagem de erro: Tensão Máxima Excedida

Nesse caso, o usuário deverá fornecer um novo valor de tensão válido.

No caso do comando via “JOYSTICK”, a tela da Fig. 4.16 será mostrada. Essa tela apresenta o espaço de trabalho do microatuador em valores percentuais dos movimentos dos eixos X e Y e a posição atual do “joystick”. Ao se realizar um

movimento com o “joystick”, a tensão de comando a ser enviada à fonte será calculada como um valor percentual da tensão máxima estabelecida. Portanto, o ajuste correto desse valor é de extrema importância para se definir o alcance do movimento.

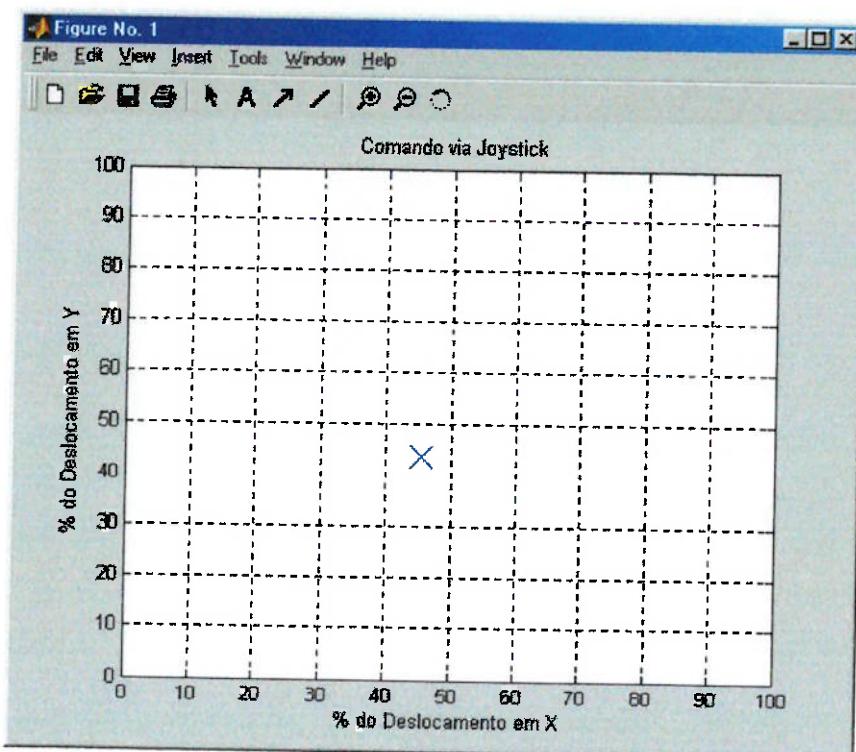


Fig. 4.16 – Tela de operação via “joystick”

Quando em operação via “joystick”, caso o usuário tente entrar com os valores de tensão para os movimentos X e Y, uma mensagem de erro (Fig. 4.17) será mostrada.

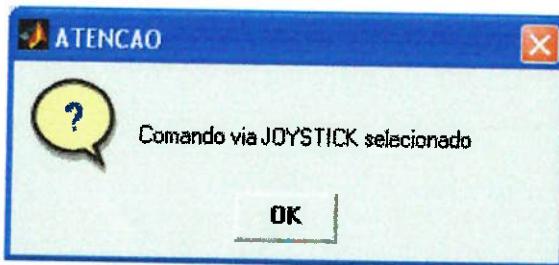


Fig. 4.17 – Mensagem de erro: Operação via “joystick”

Para encerrar a execução do comando remoto, deve-se pressionar o botão SAIR e confirmar na caixa de diálogo como mostrada na Fig. 4.18.



Fig. 4.18 – Confirmação de saída

4.3. Experimento com a Mesa XY

O mecanismo fabricado está ilustrado na Fig. 4.19. No mesmo substrato, ou lâmina, de alumina (Fig. 4.20) foram fabricados diversos outros mecanismos, que não serão abordados neste trabalho. Essas duas imagens foram feitas usando o microscópio Mitutoyo.

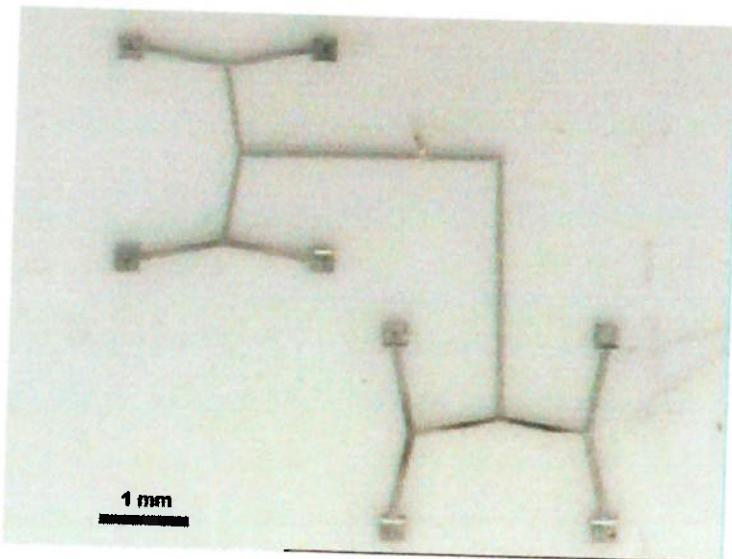


Fig. 4.19 - Mecanismo da Mesa XY fabricado

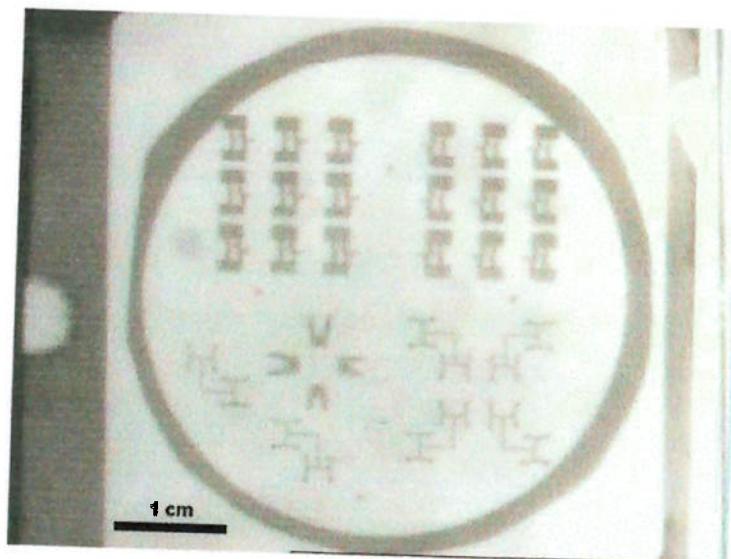


Fig. 4.20 - Substrato de alumina que contém a Mesa XY fabricada

Para uma tensão de 2V, os resultados foram:

Tab. 4.2 - Resultados experimentais (mesa XY)

Estrutura Atuada		Corrente Elétrica (A)	Deslocamento (μm)
Eixo X	1	1,440	5,12
	2	1,490	3,82
	1 e 2	1,890	5,97
Eixo Y	1	1,550	5,50
	2	1,550	9,00
	1 e 2	1,225	26,44

Ao testar os deslocamentos nos eixos X e Y de forma isolada, o deslocamento em X foi realizado em uma direção 10° diferente da esperada, indicando até uma certa atuação não desejada no eixo Y.

O “joystick” pode ser usado conforme planejado (Fig. 4.21), aplicando as tensões nos terminais do mecanismo de acordo com a posição em que se encontrava. A posição da extremidade da mesa XY podia ser visualizada em tempo real através do monitor do microcomputador. A imagem pode ser vista na Fig. 4.22, capturada na estação microprovadora usando lentes de 200×.

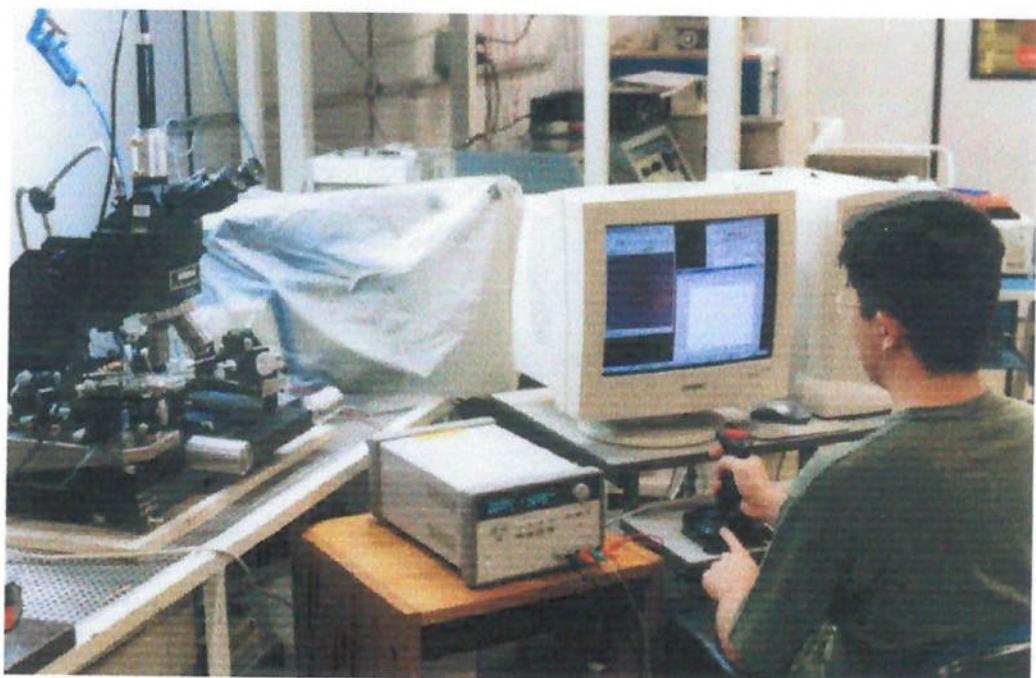


Fig. 4.21 - Equipamento com usuário operando o comando por “joystick”

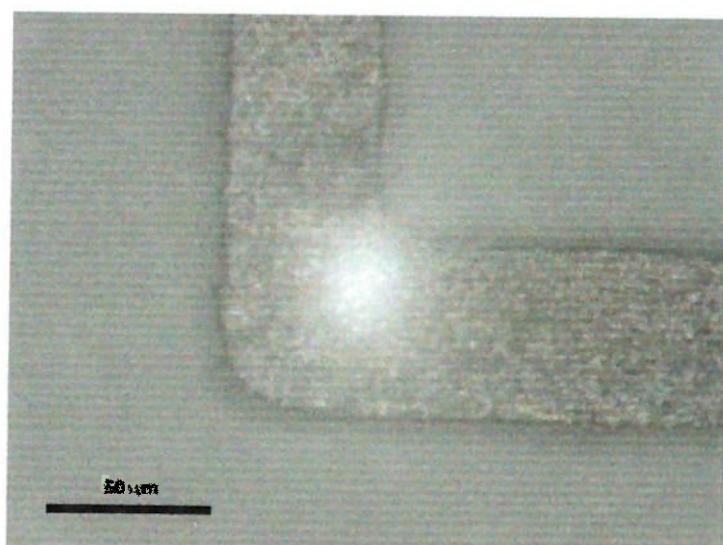


Fig. 4.22 - Extremidade da Mesa XY cuja posição era controlada por “joystick”

4.4. Simulação Revisada da Mesa XY

As correntes nos terminais T_{1x} e T_{2x} , mais próximos dos terminais onde a tensão elétrica foi aplicada, foram de aproximadamente 1,96A. Nos terminais T_{1y} e T_{2y} , as correntes simuladas ficaram próximas de 94mA. No total, pelos terminais V_{1x} e V_{2x} passaram 4,1A.

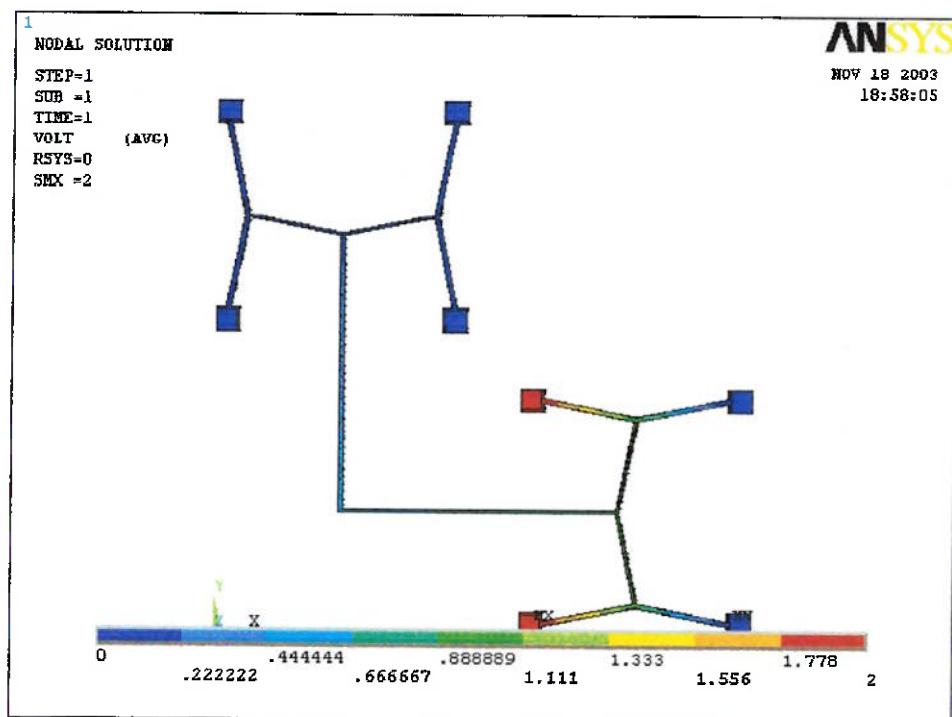


Fig. 4.23 - Simulação Revisada da Mesa XY: Distribuição de Tensão Elétrica

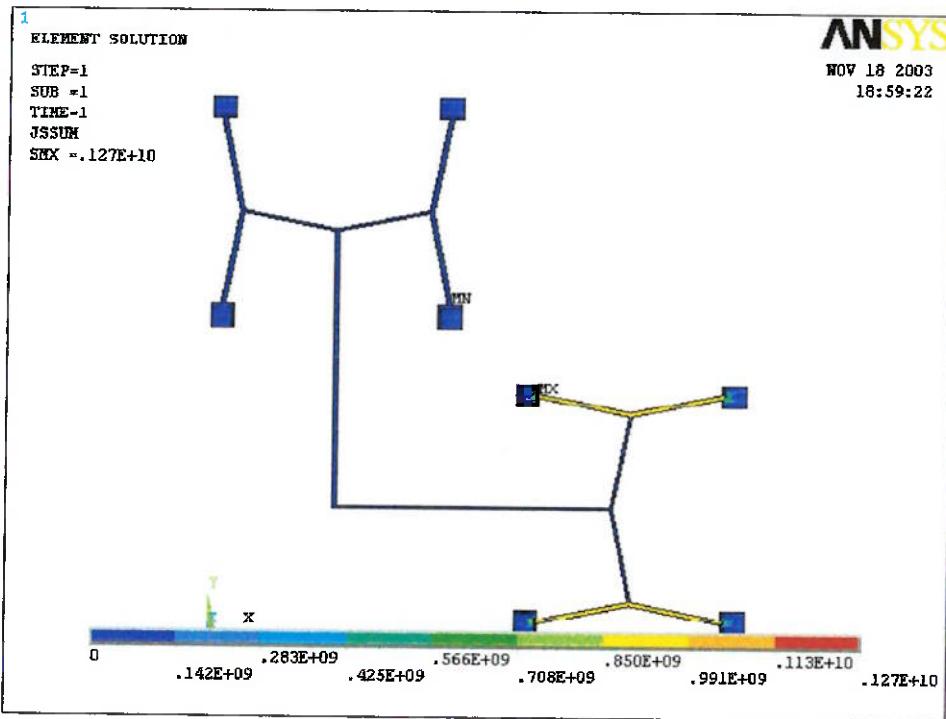


Fig. 4.24 - Simulação Revisada da Mesa XY: Densidade de Corrente

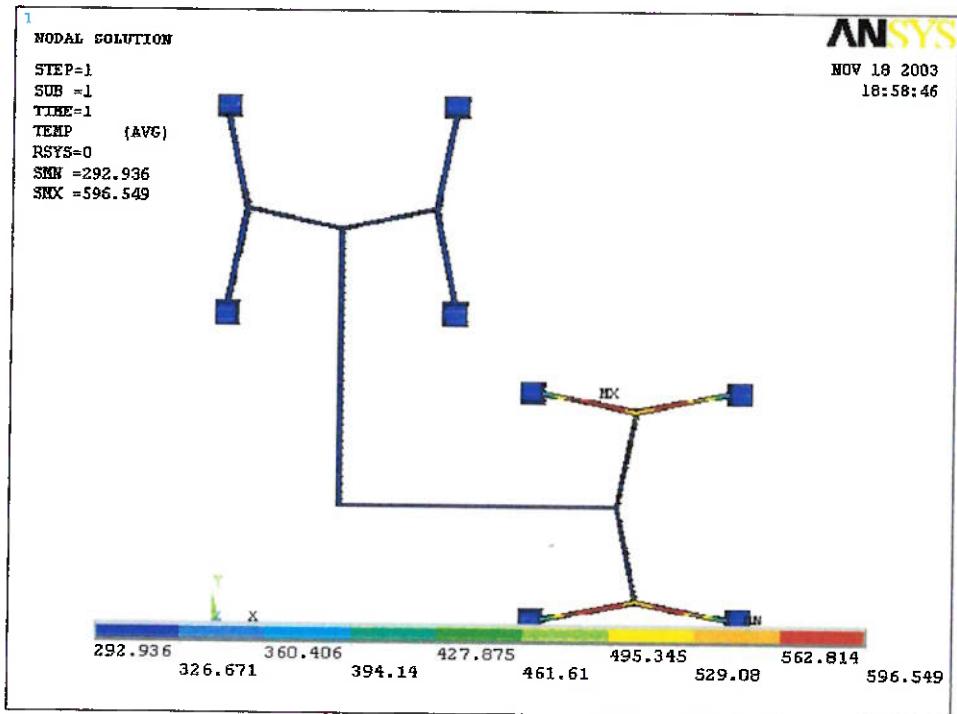


Fig. 4.25 - Simulação Revisada da Mesa XY: Distribuição de Temperatura

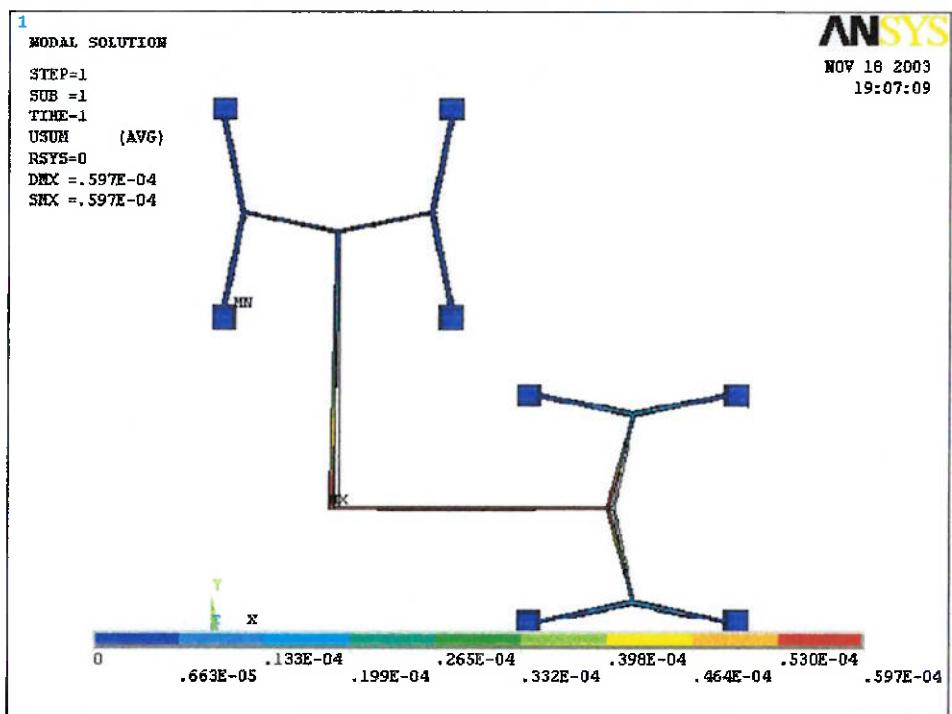


Fig. 4.26 - Simulação Revisada da Mesa XY: Deslocamento

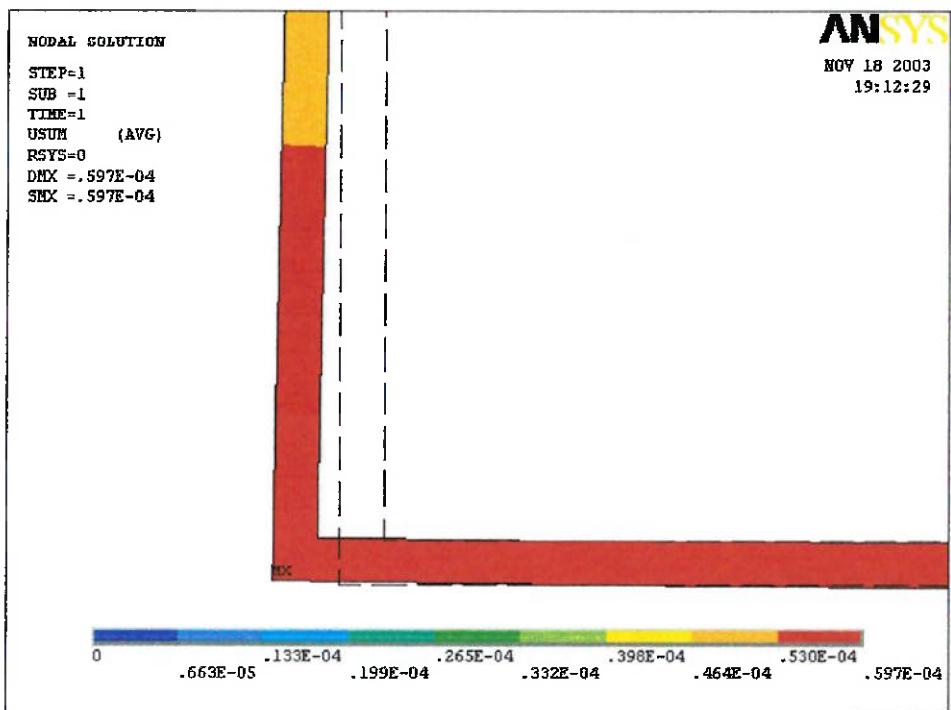


Fig. 4.27 - Simulação Revisada da Mesa XY: Deslocamento (detalhe)

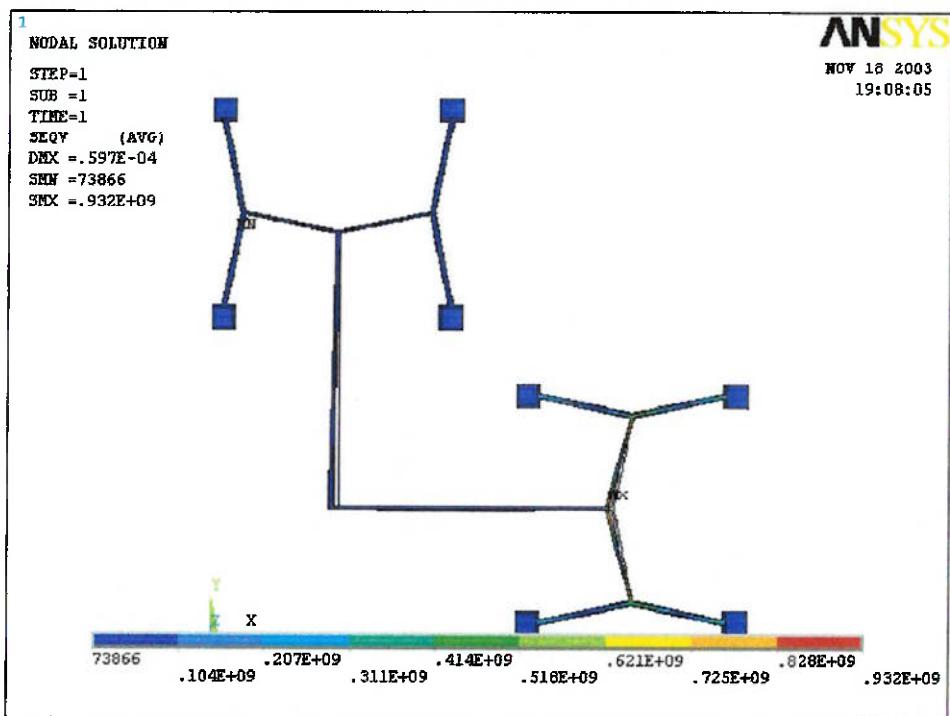


Fig. 4.28 - Simulação Revisada da Mesa XY: Tensão Mecânica (Von Mises)

5. DISCUSSÃO

5.1. Simulações de MEMS

5.1.1. Garra (projeto otimizado)

A atuação da garra para um deslocamento em X não produz o efeito desejado, resultando em um deslocamento na direção Y, no sentido negativo (para baixo). Isso acontece pelo fato do mecanismo otimizado ser muito sensível a alterações geométricas: uma pequena mudança nas juntas flexíveis pode fazer com que um modo de flexão mais provável seja substituído por outro, resultando em um movimento não desejado.

Pode-se verificar que a maioria dos resultados obtidos pela simulação da garra permitem a execução do projeto, se for aceitável o deslocamento obtido de alguns micrômetros:

- A temperatura máxima de 311K (38°C) permite seu uso até em aplicações biológicas, uma aplicação particularmente sensível, sem danificar possíveis tecidos e células
- A tensão de escoamento do níquel é de aproximadamente 620MPa (eletrodepositado com corrente de 20mA/cm² (FRITZ et al., 2000)), bem superior à máxima tensão (Von Mises) obtida (92,8MPa)

A corrente consumida é o único ponto que não permitiria o uso da garra com a fonte DC controlada remotamente (Agilent E3649A), cuja corrente máxima fornecida é de 1,4A. Para efeito de verificação da simulação, não haveria problemas, visto que o laboratório onde foi conduzido os experimentos possui uma fonte que suporta 10A (Advantest TR6162).

Para que fosse possível um deslocamento de maior magnitude, com a temperatura máxima mais próxima do limite para o níquel usado na fabricação, seria necessário uma corrente maior. Essa corrente poderia ultrapassar o limite da fonte DC empregada, até o da Advantest. Principalmente se for considerado que já é

necessária uma corrente de 8,7A para atuar todos os graus de liberdade, conforme observado na simulação.

Para reduzir a corrente necessária para incrementar mais a temperatura, poderia-se diminuir as dimensões da garra e com isso aumentar a resistência à passagem de corrente e reduzir a área total por onde o calor é perdido. Mas não é simples diminuir dimensões da garra simulada: seu tamanho é consequência da metodologia de fabricação que seria empregada. Para a confecção das máscaras usadas no processo de fotolitografia, seria empregado um fotolito comum, cuja resolução permitiria dimensões mínimas de 50 μm . Assim, a menor largura no perfil do mecanismo flexível não poderia ser inferior a esse valor, o que levou a aumentá-lo proporcionalmente nas duas dimensões até que o valor mínimo de largura fosse atingido.

Assim, desejando um deslocamento maior sem precisar mudar o processo de fabricação atual para um mais oneroso, este mecanismo não foi selecionado para fabricação.

5.1.2. Mesa XY (projeto intuitivo)

Dada a geometria mais simples e uniforme, foi possível reduzir o tamanho da mesa XY de modo a permitir a sua fabricação pelo processo proposto e ter uma corrente aceitável de aproximadamente 2A por par de terminais (4A, se considerarmos a atuação completa de um eixo). A tensão elétrica aplicada que gerou este valor de corrente foi considerada como o valor máximo que seria usado nas verificações experimentais, já que se obteve uma temperatura máxima de 586K (313°C) que está bem próxima do limite de 350°C normalmente usado para MEMS eletrotermomecânicos de níquel eletrodepositado (QUE; PARK; GIANCHANDANI, 2001). Para estas condições foi determinado um deslocamento de 68 μm , algo bem maior que o que foi simulado para a garra (da ordem de micrômetros).

A máxima tensão mecânica (Von Mises) encontrada, 1,05GPa, é maior que a tensão de escoamento para o níquel eletrodepositado (FRITZ et al., 2000), o que indica que o funcionamento da mesa XY levaria a possíveis danos permanentes na

estrutura já nos primeiros instantes de uso, degradando a performance e funcionalidade.

Devido aos indicadores favoráveis, este mecanismo foi o escolhido para a fabricação, verificação experimental e testes do comando por “joystick”.

5.2. Interface do Comando Remoto da Fonte DC

A escolha do MATLAB como a plataforma computacional de interface de “hardware” e “software” foi acertada porque ele fornece, através dos “toolboxes”, uma funcionalidade adequada para implementar as comunicações necessárias entre os diversos componentes do sistema de comando do microatuaador, isto é, as seguintes interações:

- USUÁRIO-PROGRAMA, através da interface gráfica desenvolvida com o *guide*.
- “JOYSTICK”-MATLAB, através da construção do adaptador para o “Game Port” utilizando o “Data Acquisition Toolbox Adaptor Kit”.
- MATLAB-FONTE DC, através do “Instrument Control toolbox”, que fornece facilidade funcional para a comunicação através do barramento GPIB.

O *guide* fornece, além da facilidade de arranjo dos objetos gráficos, um suporte à programação das rotinas de “callback”, pois gera automaticamente um arquivo “.m” pré-formatado com as chamadas a essas rotinas, associadas aos seus respectivos eventos.

Além disso, o “Data Acquisition Toolbox Adaptor Kit”, através do adaptador de demonstração, simplifica a construção de um adaptador para um “hardware” de aquisição de dados específico, pois é feita através da modificação de um conjunto de arquivos já existentes para compilar a biblioteca (arquivo “.DLL”).

Finalmente, a comunicação através do barramento GPIB é realizada através do “Instrument Control Toolbox”, com funções de escrita e leitura de dados fáceis de programar e configurações específicas para determinados fabricantes de adaptadores já existentes.

5.3. Experimento com a Mesa XY

Pode-se perceber uma diferença grande entre o deslocamento simulado e o medido experimentalmente. O obtido experimentalmente, no melhor caso (atuação do eixo Y), chegou a $26\mu\text{m}$, menos da metade dos $68\mu\text{m}$ observados na simulação. Houve diferenças também na corrente resultante ao aplicar 2V nos pares de terminais: experimentalmente, a corrente em cada par de terminais ficou próxima de 1,5A, um valor menor que o da simulação, de 2,1A. Essas divergências podem ser fruto de uma resistividade elétrica e condutividade térmica usadas na simulação diferentes do encontrado para o níquel da estrutura. Outro fator que pode ter causado as diferenças é o coeficiente de convecção adotado na modelagem em ANSYS. Deve-se notar que a temperatura no mecanismo fabricado chegou ao limite máximo, fato que pode ser comprovado pela rápida oxidação da estrutura, conforme observado em Que; Park; Gianchandani (2001) e que pode ser visto na Fig. 4.19. Em outras palavras, experimentalmente a estrutura atingiu uma temperatura superior à simulada com 1,5A, ao invés de 2A.

Pela Tab. 4.2, pode-se verificar que a atuação completa dos eixos X e Y, individualmente, consome uma corrente diferente da soma das que foram obtidas ao atuar cada uma das metades da estrutura completa do eixo em questão. Para que isso aconteça, podem estar ocorrendo alterações na resistividade elétrica do material ao longo do experimento.

Pela mesma tabela, verifica-se que, ao invés de haver amplificação do deslocamento das estruturas atuadas eletrotermicamente, existe uma redução no caso do eixo X, com um deslocamento total de aproximadamente $9\mu\text{m}$ tornando-se menor que $6\mu\text{m}$. Uma explicação para isso é o deslocamento amplificado ter saído do plano XY, o que poderia ter acontecido por causa imperfeições de fabricação ou singularidades da estrutura, elementos que teriam esse efeito principalmente se estivessem localizados na parte superior ou inferior do mecanismo. Também devido a singularidades, poderia ter acontecido uma flexão não esperada no mecanismo de amplificação, perdendo sua funcionalidade.

A Tab. 4.2 ainda indica uma marcada assimetria entre os movimentos obtidos ao atuar os dois eixos. Na Fig. 4.22, pode-se notar uma diferença próxima de 10% entre as larguras das hastas dos dois eixos. Essa diferença pode estar presente por todo o mecanismo, causando diferenças de resistividade elétrica e condutividade térmica entre as estruturas dos dois eixos, gerando a assimetria observada. Um fator que pode ser investigado é o quanto é afetada a dimensão dos MEMS se estes estiverem alinhados vertical ou horizontalmente no fotolito do processo de fabricação (pode existir diferença de resolução entre os eixos da máquina que os produz).

Como resultado do acionamento do eixo X, houve um deslocamento com ângulo de 10° não em direção à estrutura do eixo Y, mas no lado oposto, como se houvesse uma atuação neste último eixo. Isso pode ser explicado pelo aterrramento da estrutura do eixo Y durante a atuação do eixo X, o que permite a existência de corrente no mecanismo não atuado proveniente da haste cuja extremidade está sendo observada pelo microscópio. Dependendo da densidade de corrente, a estrutura pode aquecer e expandir, como se estivesse sendo atuada. Esse fato pode ser comprovado pela atuação do eixo X seguida da atuação do Y. Assim, observou-se que o deslocamento em Y não tem quase alteração na direção (é praticamente a direção desejada, sem desvios), dado que a hipotética corrente que passaria pelas hastas dos eixos é muito pequena se comparada à atuação que já ocorre.

Outro fato curioso foi a não ocorrência de falha da mesa XY, já que, pela simulação, existiam tensões muito superiores à tensão de escoamento do níquel eletrodepositado. Houve também um deslocamento muito menor, conforme já discutido, o que explicaria uma possível diminuição da tensão mecânica presente no mecanismo suficiente para esta não ser suficiente para uma deformação plástica, permanecendo no regime elástico.

O uso do “joystick” foi bem sucedido, apesar da limitação de 1,4A da fonte DC. Um atraso de aproximadamente meio segundo na resposta do mecanismo foi detectado. Um tempo de resposta dessa magnitude era esperado, devido à natureza da atuação baseada na expansão térmica. Esse atraso não pode ser proveniente do tempo de processamento do programa usado, em MATLAB nesse caso, visto que a fonte

DC indicava a tensão desejada no mesmo instante em que o comando era dado pela interface. Para que a reação fosse mais rápida, seria necessário também ter o controle sobre o método de resfriamento da mesa XY, composto principalmente pela convecção, reduzindo-a na atuação e aumentando-a ao cessar o aquecimento da estrutura.

5.4. Simulação Revisada da Mesa XY

Mesmo com as alterações na modelagem, a corrente consumida manteve-se em aproximadamente 2A por par de terminais, ou seja, os mesmos 4A por eixo. A temperatura obtida também manteve-se próxima do limite de 350°C, chegando a 596K (323°C). O deslocamento foi de 60µm, próximo dos 68µm obtidos na simulação anterior. A tensão mecânica (Von Mises) continua no mesmo nível danoso à estrutura, chegando a 932MPa.

Através dessa nova simulação, foi observada uma corrente nos terminais T_{1y} e T_{2y} , que pode ser a responsável pela atuação não desejada no eixo Y. Não foi observada uma atuação involuntária nesse eixo proveniente de um aumento apreciável de temperatura em sua estrutura, mas com a alteração de alguns parâmetros, como resistividade elétrica e convecção, ou a introdução de uma singularidade que aumentasse a densidade de corrente puntualmente seria possível que houvesse uma certa expansão térmica não esperada.

6. CONCLUSÕES

Neste trabalho, foram realizadas análises dos MEMS eletrotermomecânicos utilizando Método de Elementos Finitos, cujos resultados foram úteis para se obter uma estimativa das grandezas envolvidas, tanto da corrente consumida como do deslocamento resultante. Também usou-se técnicas de microfabricação de superfície relativamente baratas (fotolitografia utilizando fotolitos comuns), tendo como resultado uma estrutura que possibilitou o comando remoto por “joystick”, cujo programa de comando foi feito usando o MATLAB.

Feitas as simulações e experimentos, foram encontradas discrepâncias, que podem ser minimizadas através de parâmetros mais representativos da condição de utilização, como os valores das propriedades elétricas e térmicas do níquel eletrodepositado. Como trabalho futuro, pretende-se caracterizar o níquel eletrodepositado e a convecção do ambiente experimental de modo a termos uma simulação mais próxima da realidade.

O conhecimento adquirido terá grande utilidade na dissertação de mestrado a ser desenvolvida por um dos alunos, que deverá abordar a otimização de MEMS eletrotermomecânicos multifásicos.

7. ANEXO A – Código-Fonte do “software” de Comando em MATLAB

```

function varargout = interface(varargin)
% Arquivo ".m" de aplicacao da interface para interface.fig
%   FIG = INTERFACE inicia o GUI da interface.
%   INTERFACE('callback_name', ...) chama o "callback" especificado.

if nargin == 0 % inicia GUI

    fig = openfig(mfilename,'reuse');

    % Gera uma estrutura de operacao ("handle") para se passar aos "callbacks",
    % e armazena-lo.
    handles = guihandles(fig);
    guidata(fig, handles);

    if nargout > 0
        varargout{1} = fig;
    end

elseif ischar(varargin{1}) % chama a subfuncao ou o "callback"

    try
        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
    catch
        disp(lasterr);
    end
end

% =====
% ESTRUTURA DOS "CALLBACKS"
% =====

% -----
function varargout = Adaptador_Callback(h, eventdata, handles, varargin)
% "Pop-Up Menu" para selecao do Fabricante do adaptador
% Agilent Technologies - Value = 1
% National Instruments - Value = 2

% -----
function varargout = PrimAdd_Callback(h, eventdata, handles, varargin)
% "Pop-Up Menu" para selecao do "Primary Address" do instrumento

% -----
function varargout = Abrir_Conexao_Callback(h, eventdata, handles, varargin)
% "Toggle Button" para iniciar comunicacao com a fonte DC

    % Objeto de instrumento GPIB, variavel global
    global Objeto_GPIB;

    % Verifica o fabricante selecionado
    VENDOR = get(handles.Adaptador, 'Value');

    if VENDOR == 1
        adapt = 'agilent';
    else adapt = 'ni';
    end

    % Verifica o "Primary Address" selecionado
    PRIMARYADDRESS = get(handles.PrimAdd, 'Value');

    % Criacao de um objeto GPIB
    Objeto_GPIB = gpib (adapt, 0, PRIMARYADDRESS);

    % Abre comunicacao com a fonte DC
    fopen (Objeto_GPIB);

```

```

% -----
function varargout = Fechar_Conexao_Callback(hObject, eventdata, handles, varargin)
% "Toggle Button" para encerrar comunicacao com a fonte DC

    % Objeto de instrumento GPIB, variavel global
    global Objeto_GPIB;

    fclose (Objeto_GPIB);
    delete (Objeto_GPIB);
    clear;

% -----
function varargout = Tensao_MAX_Callback(hObject, eventdata, handles, varargin)
% "Edit Text" para ajustar a Tensao Maxima

% -----
function varargout = TipoComando_Callback(hObject, eventdata, handles, varargin)
% "Pop-Up Menu" para selecionar o Tipo de Comando

    % Objeto de instrumento GPIB, variavel global
    global Objeto_GPIB;

    % Verifica o tipo de comando selecionado
    TIPO_DE_COMMANDO = get(handles.TipoComando, 'Value');

    if TIPO_DE_COMMANDO == 2      % "Joystick"

        % Registro do Adaptador de Joystick "joy.DLL"
        daqregister ('joy.dll');

        % Verifica a tensao maxima ajustada
        Tensao_MAX = get(handles.Tensao_MAX, 'String');

        % Conversao de "string" para numero
        [row_Tens_MAX, col_Tens_MAX] = size (Tensao_MAX);
        Tensao_MAXIMA = 0;
        j=col_Tens_MAX - 1;
        for i=1:col_Tens_MAX
            Tensao_MAXIMA = Tensao_MAXIMA + (str2num(Tensao_MAX(i)))*(10^j));
            j=j-1;
        end

        load chirp
        v = y; clear y;

        % Cria conexao com o "joystick".
        % Adiciona dois canais. Um para x e um para y.
        ai=analoginput('joy',1);
        addchannel(ai,[1 2]);

        % Configura a tela de operacao via "joystick"
        hFig=figure('DoubleBuffer','on');
        hJoy = plot(0,0,'x','MarkerSize',15);
        hold on;
        grid on;
        axis([0 100 0 100]);
        title('Comando via Joystick');
        xlabel ('% do Deslocamento em X');
        ylabel ('% do Deslocamento em Y');

        % Ajusta a resolucao da tensao
        erro = 0.1;

        Tensao_X=0;
        Tensao_Y=0;
        joystick_atual_X=0;
        joystick_atual_Y=0;

        % Ajuste inicial de tensao (zero) nas saidas 1 e 2
        fprintf (Objeto_GPIB, 'INST:SEL OUT1');
        fprintf (Objeto_GPIB, 'VOLT %d', Tensao_X);
        fprintf (Objeto_GPIB, 'INST:SEL OUT2');
        fprintf (Objeto_GPIB, 'VOLT %d', Tensao_Y);
        fprintf (Objeto_GPIB, 'OUTPUT ON');

```

```

while 1
    % Amostragem da posicao do "joystick"
    joystick = getsample(ai);

    % Valores negativos de posicao nao geram movimento do MEMS
    if joystick(1)<0
        joystick(1)=0;
    end
    if joystick(2)>0
        joystick(2)=0;
    end

    % Mostra posicao do "joystick" na tela
    set(hJoy,'XData',joystick(1)*10,'YData',-joystick(2)*10);
    drawnow;

    % Se o movimento em X for maior do que a resolucao definida
    if abs(joystick_atual_X-joystick(1)) >= erro
        joystick_atual_X = joystick(1);

        % Calculo da tensao a ser enviada para a fonte
        Tensao_X = (joystick(1)*(Tensao_MAXIMA/1000))/10;
        fprintf (Objeto_GPIB, 'INST:SEL OUT1'); % Output 1
        fprintf (Objeto_GPIB, 'VOLT %d', Tensao_X); % Envia tensao
    end

    % Se o movimento em Y for maior do que a resolucao definida
    if abs(joystick_atual_Y-joystick(2)) >= erro
        joystick_atual_Y = joystick(2);

        % Calculo da tensao a ser enviada para a fonte
        Tensao_Y = -(joystick(2)*(Tensao_MAXIMA/1000))/10;
        fprintf (Objeto_GPIB, 'INST:SEL OUT2'); % Output 2
        fprintf (Objeto_GPIB, 'VOLT %d', Tensao_Y); % Envia tensao
    end
end
end

%
% -----
function varargout = TensaoX_Callback(hObject, eventdata, handles, varargin)
% "Edit Text" para entrada de tensao (mV) do movimento em X, via teclado

%
% -----
function varargout = TensaoY_Callback(hObject, eventdata, handles, varargin)
% "Edit Text" para entrada de tensao (mV) do movimento em Y, via teclado

%
% -----
function MOVER_Callback(hObject, eventdata, handles)
% "Toggle Button" para movimento via teclado (mover MEMS)

    % Verifica o tipo de comando selecionado
    TIPO_DE_COMMANDO = get(handles.TipoComando, 'Value');

    if TIPO_DE_COMMANDO == 1      % Teclado

        % Objeto de instrumento de GPIB, variavel global
        global Objeto_GPIB;

        % Verifica tensao maxima ajustada
        Tens_MAX = get(handles.Tensao_MAX, 'String');

        % Conversao de "string" para numero
        [row_Tens_MAX, col_Tens_MAX] = size (Tens_MAX);
        Tensao_MAXIMA = 0;
        j=col_Tens_MAX - 1;
        for i=1:col_Tens_MAX
            Tensao_MAXIMA = Tensao_MAXIMA + (str2num(Tens_MAX(i))*(10^j));
            j=j-1;
        end
    end

```

```

% Verifica o valor da tensao X ajustada
X = get(handles.TensaoX, 'String');

% Verifica o valor da tensao Y ajustada
Y = get(handles.TensaoY, 'String');

% Conversao de "string" para numero
[row_X, col_X] = size (X);
[row_Y, col_Y] = size (Y);
direcao_X=0;
direcao_Y=0;
j=col_X-1;
for i=1:col_X
    direcao_X = direcao_X + (str2num(X(i)))*(10^j));
    j=j-1;
end

j=col_Y-1;
for i=1:col_Y
    direcao_Y = direcao_Y + (str2num(Y(i)))*(10^j));
    j=j-1;
end

% Se tensoes X e Y forem menores que a tensao maxima
if (direcao_X <= Tensao_MAXIMA) & (direcao_Y <= Tensao_MAXIMA)

    % Calculo da tensao a ser enviada para a fonte (em VOLTS)
    direcao_X = direcao_X/1000;
    direcao_Y = direcao_Y/1000;

    fprintf (Objeto_GPIB, 'INST:SEL OUT1');           % Output 1
    fprintf (Objeto_GPIB, 'VOLT %d', direcao_X);      % Envia tensao
    fprintf (Objeto_GPIB, 'INST:SEL OUT2');           % Output 2
    fprintf (Objeto_GPIB, 'VOLT %d', direcao_Y);      % Envia tensao
    fprintf (Objeto_GPIB, 'OUTPUT ON');                % Aciona Outputs 1 e 2

% Se tensoes X e/ou Y forem maiores que a tensao maxima
else
    % Mensagem de erro
    button = questdlg('TENSAO MAXIMA EXCEDIDA ! Entre com nova tensao',
                      'ATENCAO', 'OK', 'OK');
    switch button
        case 'OK', quit cancel;
    end
end

% Se pressionar o botao "MOVER" com o tipo de comando "joystick" selecionado
else
    % Mensagem de erro
    button = questdlg('Comando via JOYSTICK selecionado', 'ATENCAO', 'OK', 'OK');
    switch button
        case 'OK', quit cancel;
    end
end

%
function PARAR_Callback(hObject, eventdata, handles)
% "Toggle Button" para movimento via teclado (parar MEMS)

% Objeto de instrumento GPIB, variavel global
global Objeto_GPIB;

fprintf (Objeto_GPIB, 'INST:SEL OUT1'); % Ouptut 1
fprintf (Objeto_GPIB, 'VOLT 0');        % Envia tensao nula
fprintf (Objeto_GPIB, 'INST:SEL OUT2'); % Output 2
fprintf (Objeto_GPIB, 'VOLT 0');        % Envia tensao nula
fprintf (Objeto_GPIB, 'OUTPUT ON');     % Aciona Outputs 1 e 2
fprintf (Objeto_GPIB, 'OUTPUT OFF');    % Desliga Outputs 1 e 2

```

```
% -----
function varargout = SAIR_Callback(h, eventdata, handles, varargin)
% "Toggle Button" para encerrar o programa

    button = questdlg('Deseja encerrar?', ...
                       'Saida', 'Sim', 'Nao', 'Nao');
switch button
case 'Sim', quit;
case 'Nao', quit cancel;
end
```

8. LISTA DE REFERÊNCIAS

FRITZ, T.; LEUERER, T.; KRÜGER, C.; MOKWA, W.; SCHNAKENBERG, U. **Mechanical Properties of Electroplated Nickel.** Techn. Dig. Micro Materials Micro Mat, p.752-755, 2000. /Apresentado a 3rd International Conference and Exhibition, Berlin, Germany, April 17-19 (2000)/

JONSMANN, J. **Technology Development for Topology Optimised Thermal Microactuators.** 1999. p.116. Tese (Doutorado) – Microelectronics Centre (Technical University of Denmark). Technical University of Denmark, February 25, 1999.

MENCIASSI, A. **Microfabricated Grippers for Micromanipulation of Biological and Mechanical Objects.** 1999. p.124. Tese (Doutorado) – Microfabrication Technologies Laboratory (Scuola Superiore Sant'Anna, Pisa). Pisa, Italy, January 1999.

MOULTON, T.; ANANTHASURESH, G. H. **Micromechanical Devices with embedded electro-thermal compliant actuation.** Sensors and Actuators A: Physical, v.90, p.38-48, 2001. Disponível em: <<http://www.sciencedirect.com/science/journal/09244247>>. Acesso em: 7 Nov 2003.

QUE, L.; PARK, J.; GIANCHANDANI, Y. B. **Bent-Beam Electrothermal Actuators—Part I: Single Beam and Cascaded Devices.** Journal of Microelectromechanical Systems, v.10, n.2, June 2001.

THE MATHWORKS, “Building a Joystick Adapter for The Data Acquisition Toolbox”. Apresenta um artigo que descreve a implementação da leitura de um “joystick” usando a “Data Acquisition Toolbox”. Disponível em: <http://www.mathworks.com/company/digest/june01/joystick.shtml>. Acesso em: 19 de nov. 2003.