



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e informática
Unidade Acadêmica de Engenharia Elétrica

Ensaio Experimentais com Amostras de Liga com Memória de Forma

Arthur Abrantes Paiva Pequeno

Relatório das Atividades de Estágio

Campina Grande
27 de Maio de 2009

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e informática
Unidade Acadêmica de Engenharia Elétrica

Arthur Abrantes Paiva Pequeno

Ensaio Experimentais com Amostras de Liga com Memória de Forma

*Relatório das atividades desenvolvidas durante o período de estágio,
submetido à Coordenação de Graduação em Engenharia Elétrica da
Universidade Federal de Campina Grande - Campus de Campina
Grande como parte dos requisitos necessários para a obtenção do
grau de Bacharel em Engenharia Elétrica.*

Área de concentração: Instrumentação eletrônica

José Sérgio da Rocha Neto

Orientador

Campina Grande , Paraíba, Brasil

©Arthur Abrantes Paiva Pequeno, 27 de Maio de 2009

Conteúdo

| | |
|--|--------|
| Lista de Figuras | p. iii |
| Lista de Abreviaturas e Símbolos | p. i |
| 1 Introdução Teórica | p. 1 |
| 1.1 Ligas com Memória de Forma | p. 1 |
| 1.2 Efeitos de Memória de Forma e Superelasticidade | p. 2 |
| 2 Materiais e Métodos | p. 5 |
| 2.1 Amostras de SMA | p. 5 |
| 2.2 Plataforma de Ensaio | p. 6 |
| 2.3 Computador Pessoal | p. 8 |
| 3 Uso do LabVIEW | p. 9 |
| 3.1 Programação em G | p. 9 |
| 3.2 Estruturas de Execução | p. 10 |
| 3.2.1 Estrutura de repetição <i>While</i> | p. 10 |
| 3.2.2 Estrutura de execução condicional <i>If</i> | p. 12 |
| 3.3 Funções VISA (<i>Virtual Instrument Software Architecture</i>) | p. 13 |
| 3.3.1 VISA Serial | p. 13 |
| 3.3.2 VISA Read | p. 14 |

| | | |
|----------|---|--------------|
| 3.3.3 | VISA Write | p. 15 |
| 3.3.4 | VISA Close Serial Port | p. 15 |
| 3.4 | Aquisição de dados | p. 16 |
| 3.4.1 | Tensão Mecânica | p. 16 |
| 3.4.2 | Deformação Linear | p. 18 |
| 3.4.3 | Temperatura | p. 20 |
| 3.5 | Acionamento dos Atuadores | p. 22 |
| 3.5.1 | Acionamento do motor de passo | p. 22 |
| 3.5.2 | Controle de Temperatura | p. 25 |
| 3.6 | Integração com MatLab | p. 26 |
| 4 | Resultados e Conclusões | p. 28 |
| | Bibliografia | p. 35 |

Lista de Figuras

| | | |
|-----|--|-------|
| 1.1 | Diagrama da transformação de fase induzida por temperatura entre austenita e martensita maclada na ausência de carga aplicada. Fonte: (VALENZUELA, 2007) | p. 2 |
| 1.2 | Fase de transformação induzida por tensão mecânica e comportamento de superelasticidade para $T > A_f$. Fonte: (VALENZUELA, 2007) | p. 3 |
| 1.3 | Comportamento quase plástico e deformação residual são gerados quando $T < M_f$. Fonte: (VALENZUELA, 2007) | p. 3 |
| 1.4 | Efeito de memória de forma em um SMA em que uma deformação residual é recuperada por meio de aquecimento. Fonte: (VALENZUELA, 2007) | p. 4 |
| 2.1 | (a)Fio em forma de fita. (b)Fio com secção circular | p. 6 |
| 2.2 | Vistas Frontal(a) e Posterior(b) da plataforma de Ensaios e Caracterização de Materiais. | p. 7 |
| 3.1 | Diagrama de Blocos G. Fonte: (INSTRUMENTS, 2008) | p. 10 |
| 3.2 | Interface de Usuário. Fonte: (INSTRUMENTS, 2008) | p. 11 |
| 3.3 | Bloco referente a uma estrutura de repetição While. Fonte: (LABVIEW, 2004) | p. 11 |
| 3.4 | Bloco referente a uma estrutura de condição If. Fonte: (LABVIEW, 2004) | p. 12 |
| 3.5 | Conjunto de blocos utilizados para iniciar a comunicação serial. | p. 13 |
| 3.6 | Janela de Usuário, campos de ajuste da comunicação serial. | p. 14 |
| 3.7 | Bloco de leitura da porta serial. Fonte: (LABVIEW, 2004) | p. 14 |
| 3.8 | Bloco de escrita na porta serial. Fonte: (LABVIEW, 2004) | p. 15 |
| 3.9 | Bloco de fechamento de comunicação. Fonte: (LABVIEW, 2004) | p. 16 |

| | | |
|------|--|-------|
| 3.10 | Diagrama de blocos para aquisição da tensão mecânica | p. 17 |
| 3.11 | Campos de entrada para os dados da amostra de SMA. | p. 17 |
| 3.12 | Curva da tensão mecânica vista na janela de usuário. | p. 18 |
| 3.13 | Diagrama de blocos para aquisição da deformação linear. | p. 19 |
| 3.14 | Curva da deformação linear vista na janela de usuário. | p. 20 |
| 3.15 | Diagrama de blocos da aquisição da temperatura. | p. 21 |
| 3.16 | Curva de temperatura na janela do usuário. | p. 22 |
| 3.17 | Modelo do sinal de controle que deve ser enviado para o microcontrolador. | p. 23 |
| 3.18 | Diagrama de blocos que controla a movimentação do motor de passos. | p. 23 |
| 3.19 | Campos de entrada de controle do motor, na janela de usuário. | p. 24 |
| 3.20 | Campos de entrada para controle de temperatura. | p. 24 |
| 3.21 | Construção de três vetores a partir de dados vindos das leituras dos sensores. | p. 26 |
| 4.1 | Curva Tensão mecânica x Deformação do ensaio do SMA | p. 29 |
| 4.2 | Curva Tensão mecânica x Deformação obtida em literatura | p. 30 |
| 4.3 | Curva Tensão mecânica x Deformação obtida em literatura | p. 30 |
| 4.4 | Gráfico para determinar a temperatura de transformação de fase. | p. 31 |
| 4.5 | Curva Tensão mecânica x Deformação obtida no primeiro ensaio com a fita delgada. | p. 32 |
| 4.6 | Curva Tensão mecânica x Deformação obtida no segundo ensaio com a fita delgada. | p. 33 |
| 4.7 | Curva Tensão mecânica x Deformação encontrada na literatura para fita delgada. Fonte: (NETO; ARAÚJO, 2008). | p. 33 |
| 4.8 | Gráfico tridimensional gerado pelo MatLab. | p. 34 |

Lista de Abreviaturas e Símbolos

| | |
|---------|---|
| Af | Temperatura final da transformação para fase Austenita |
| As | Temperatura inicial da transformação para fase Austenita |
| GUI | G User Interface |
| LabVIEW | Laboratory Virtual Instrument Engeneering Workbench |
| LMF | Ligas com Memória de Forma |
| M+ | Fase Martensita demaçada |
| MatLab | Matrix Laboratory |
| Mf | Temperatura final da transformação para fase Martensita |
| Ms | Temperatura inicial da transformação para fase Martensita |
| PC | Personal Computer |
| PWM | Pulse Wide Modulation |
| SMA | Shape Memory Alloy |
| VI | Virtual Instrument |
| VISA | Virtual Instrument Software Architecture |
| VISA | Virtual Instrument Software Architecture |

1 Introdução Teórica

1.1 Ligas com Memória de Forma

As Ligas com Memória de Forma - LMF ou SMA (*Shape Memory Alloy*) constituem uma classe de compostos metálicos que possuem a capacidade de desenvolver e recuperar deformações relativamente grandes podendo alcançar em torno de 8% sem apresentar deformações plásticas. Esta característica é devido, em grande parte, à capacidade que estes materiais possuem de desenvolver transformações de fase na presença de tensões mecânicas aplicadas e/ou variações de temperatura. Estas transformações são da fase austenítica, no estado livre de tensões e altas temperaturas, para diversas variantes da fase martensítica a temperaturas mais baixas que a da fase austenítica. Embora estes materiais possam ser considerados como uma tecnologia emergente, vários deles já são comercialmente disponíveis na forma de fios, tubos, chapas e barras e têm sido usados em aplicações diversas na engenharia e biomédica (VALENZUELA, 2005).

As Ligas com Memória de Forma produzidas na forma de fios e fitas delgadas são consideradas atuadores lineares por natureza e apresentam grande possibilidade de aplicação na robótica, na odontologia, na produção de atuadores miniaturizados, de sensores, antenas para celulares e de músculos artificiais, devido a sua grande capacidade de carga quando confrontados com as suas dimensões (NETO; ARAÚJO, 2008).

Porém o uso otimizado desse material como atuador requer conhecimento de alguns parâmetros da liga que nem sempre são fornecidos pelos fabricantes, como por exemplo, as tensões de formação e reversão de martensita durante a transformação de fase sob carregamento mecânico. Durante o período de estágio, construímos aplicações virtuais capazes de realizar automaticamente dois tipos de ensaios fundamentais para extrair curvas que caracterizam o comportamento termomecânico de amostras de Liga com Memória de Forma. As aplicações foram desenvolvidas utilizando

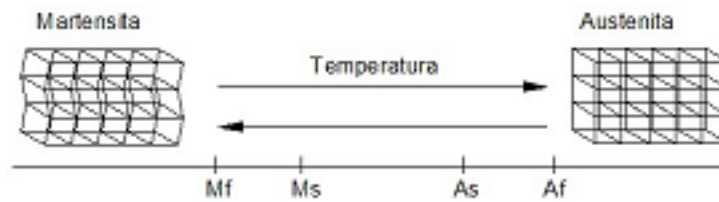


Figura 1.1: Diagrama da transformação de fase induzida por temperatura entre austenita e martensita maclada na ausência de carga aplicada. Fonte: (VALENZUELA, 2007)

uma plataforma para caracterização e ensaio de materiais em forma de fio ou fita. De acordo com (NETO; ARAÚJO, 2008), esses ensaios, mesmo que numerosos na literatura internacional, são extremamente escassos no Brasil, principalmente no caso dos atuadores lineares de LMF na forma de fitas delgadas.

1.2 Efeitos de Memória de Forma e Superelasticidade

A temperatura é uma das grandezas capazes de provocar transformações de fase em Ligas com Memória de Forma. Segundo (VALENZUELA, 2007), na ausência de uma tensão mecânica aplicada, o material transforma-se de austenita à martensita maclada, quando resfriado, e ao ser reaquecido retorna para a fase austenita. As temperaturas em que as transformações martensita e austenita iniciam são respectivamente M_s e A_s e as temperaturas em que as transformações estão completas são M_f e A_f , respectivamente. Na Figura 1.1 ilustra-se essas transformações e ao observá-la pode-se ter uma idéia clara das temperaturas críticas de transformação.

As transformações de fase podem também ser induzidas por meio de carga aplicada, como ilustrado nas Figuras 1.2 e 1.3. Na Figura 1.2, a temperatura do material está acima de A_f , o material exibe uma quase linear relação deformação-tensão mecânica à medida que a tensão mecânica é elevada até σ_{M_s} , quando começa a transição da fase austenita para a fase martensita demaclada M+. Com a redução da tensão mecânica, o material novamente exibe comportamento quase linear na fase martensita até a tensão mecânica alcançar σ_{A_s} , quando começa o processo reverso. No regime de temperatura alta é observada a completa recuperação da forma sob descarregamento, esse fenômeno é chamado de superelasticidade (VALENZUELA, 2007). Para temperaturas abaixo de M_f , como ilustrado na Figura 1.3, o material conserva uma deformação residual ϵ_r , quando descar-

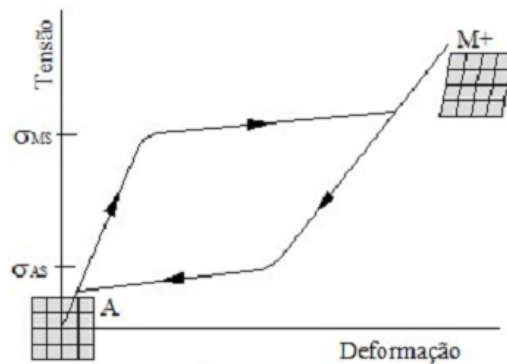


Figura 1.2: Fase de transformação induzida por tensão mecânica e comportamento de superelasticidade para $T > A_f$. Fonte: (VALENZUELA, 2007)

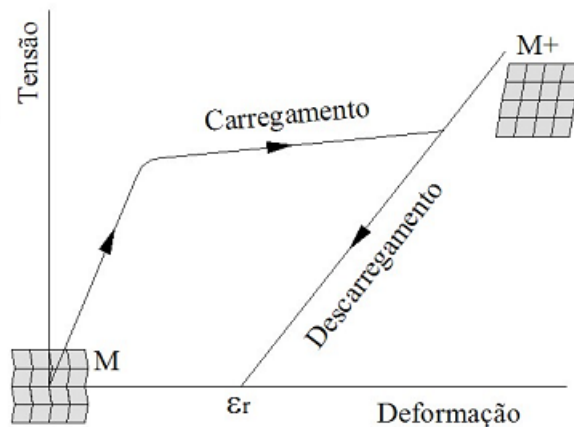


Figura 1.3: Comportamento quase plástico e deformação residual são gerados quando $T < M_f$. Fonte: (VALENZUELA, 2007)

regado. Se aquecido acima da temperatura de A_f a deformação é recuperada e o material retorna para sua forma original, como ilustrado na Figura 1.4. O comportamento nessa temperatura baixa é chamado de quase plástico e a recuperação, por meio de aquecimento, da deformação mecânica residual induzida por tensão mecânica representa o efeito de memória de forma (VALENZUELA, 2007).

Neste relatório, expomos todas as etapas que nos levaram à realização dos ensaios para verificar o efeito de memória de forma para duas amostras de Ligas com Memória de Forma, sendo uma em forma de fio, e a segunda em forma de fita delgada. Nosso intuito foi reproduzir a curva mostrada na Figura 1.3, adicionando um terceiro eixo no gráfico, no nosso caso a temperatura das amostras.

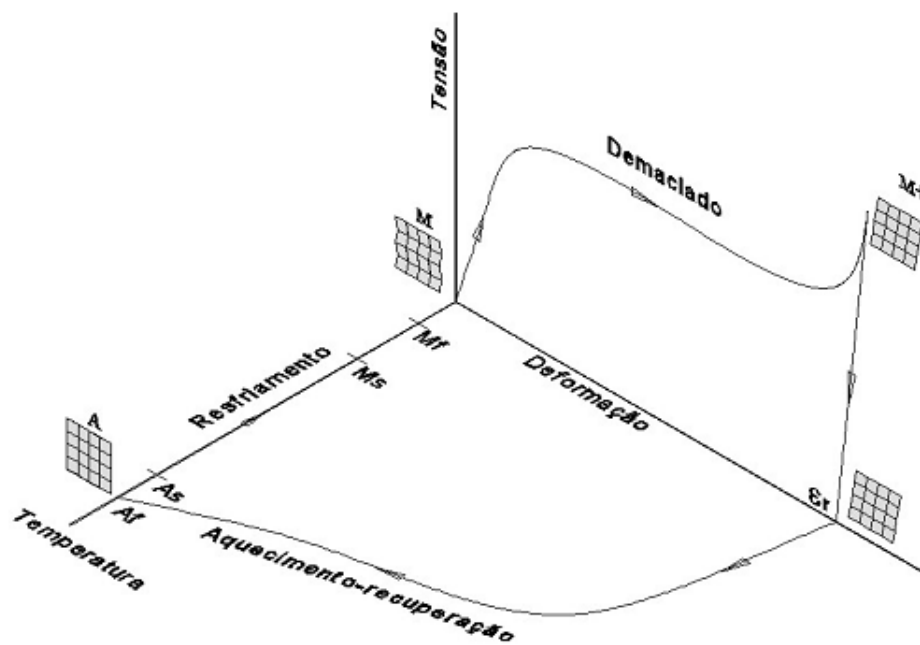


Figura 1.4: Efeito de memória de forma em um SMA em que uma deformação residual é recuperada por meio de aquecimento. Fonte: (VALENZUELA, 2007)

2 *Materiais e Métodos*

Os experimentos foram realizados com o intuito de traçar as curvas de comportamento termo-mecânico de duas amostras, sendo, em ambos os casos, extraídas curvas de Tensão mecânica (*MPa*) x Deformação Linear (%).

2.1 Amostras de SMA

Os materiais básico usado para os experimentos são amostras da Liga com Memória de Forma. Duas amostras foram utilizadas, sendo ambas submetidas ao experimento proposto, para que nos fosse possível comparar e discutir os resultados obtidos.

A primeira é um fio de secção circular, como exemplificado na Figura 2.1a, com diâmetro $d = 0,27 \text{ mm}$, e com um comprimento $C1 = 931 \text{ mm}$. Sua área de secção transversal é calculada como se vê na equação 2.1.

$$\text{Área} = \pi * r^2 \quad (2.1)$$

A segunda amostra é um fio em forma de fita, ou fio laminado, como exemplificado na Figura 2.1b, com largura $L = 0,8 \text{ mm}$, espessura $h = 0,077 \text{ mm}$ e comprimento $C2 = 255 \text{ mm}$. A área de secção transversal da segunda amostra pode ser calculada como se vê na equação 2.2.

$$\text{Área} = h * L \quad (2.2)$$

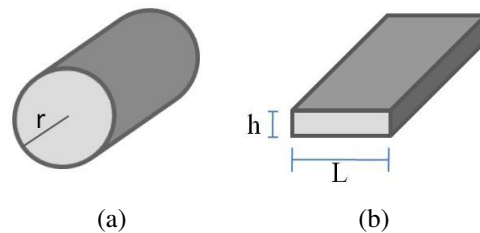


Figura 2.1: (a)Fio em forma de fita. (b)Fio com secção circular

2.2 Plataforma de Ensaio

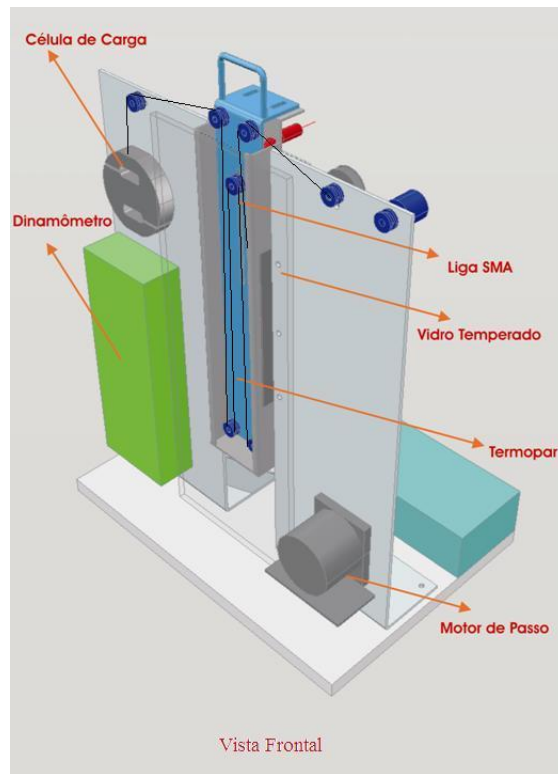
Para a realização dos ensaios utilizamos uma plataforma de ensaio e caracterização de materiais em forma de fio ou de fita. A plataforma foi construída no próprio laboratório e foi disponibilizada para que pudéssemos realizar os experimentos. Na Figura 2.2 está ilustrada uma representação da plataforma utilizada, que possui sensores por meio dos quais poderemos extrair dados para traçarmos as curvas desejadas. Devemos ter informações acerca das seguintes grandezas:

- Tensão mecânica aplicada sobre a amostra - Unidade: megaPascals (*MPa*);
- Temperatura da amostra - Unidade: graus Celsius (*°C*);
- Deformação linear do fio - Unidade: milímetros (*mm*).

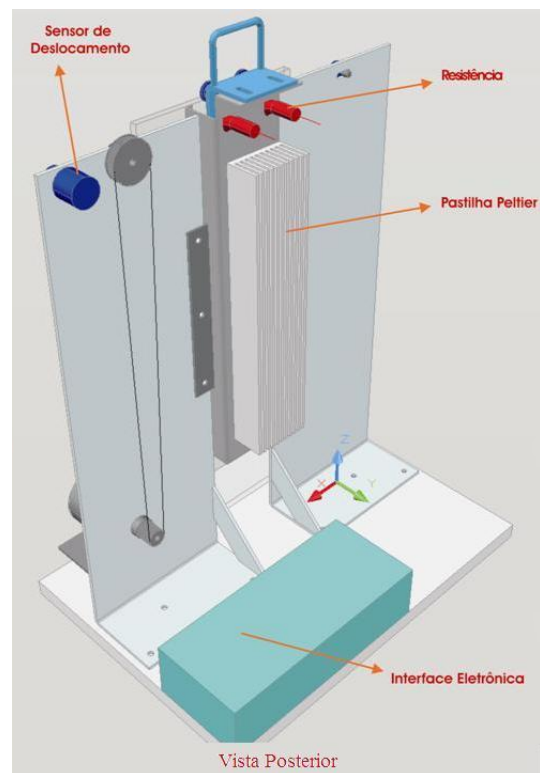
Além de fazer leituras das grandezas envolvidas, é necessário também ter a opções de ação sobre a SMA. Para isso a plataforma disponibiliza três atuadores principais, que podem ser acionados de acordo com a necessidade do usuário. São eles:

- Motor de Passo - Pode aumentar ou reduzir a tensão mecânica aplicada e deformação sobre o fio;
- Módulos de Efeito Peltier - Atuam sobre a temperatura do sistema, resfriando-o;
- Resistência Elétrica - Atuador responsável pelo aumento de temperatura do sistema.

Para fazer as leituras dos sensores e o acionamento dos atuadores, a plataforma utiliza um microcontrolador ADuC 842, pertencente à *Analog Devices*, que está sempre executando uma rotina de comunicação com a porta serial RS-232. É através de um microcomputador conectado ao microcontrolador que poderemos controlar o funcionamento do sistema.



(a) Vista Frontal



(b) Vista Posterior

Figura 2.2: Vistas Frontal(a) e Posterior(b) da plataforma de Ensaio e Caracterização de Materiais.

2.3 Computador Pessoal

Um computador pessoal - PC (*Personal Computer*) é utilizado conectado ao microcontrolador por meio de comunicação serial RS-232. O *software* usado para essa comunicação é o LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*), produzido pela *National Instruments*. A função do PC é a de receber, interpretar e armazenar os dados captados dos sensores; e enviar ordens de acionamento dos atuadores. Por ser de tão ampla importância nas atividades desenvolvidas neste estágio, dedicamos o próximo capítulo deste relatório para expor melhor como foi dado o uso do LabVIEW, suas funções e as aplicações virtuais que foram desenvolvidas para os ensaios.

3 *Uso do LabVIEW*

A ferramenta mais utilizada durante o período de estágio foi um software de instrumentação virtual, desenvolvido pela National Instruments, chamado LabVIEW. Este software deve permanecer rodando um computador, ligado à plataforma de caracterização através da porta serial, usando o padrão de comunicação RS-232.

As aplicações criadas utilizando este software são chamadas de VIs (*Virtual Instruments*). Um VI pode conter inúmeras instruções, que permitem realizar diversas operações e funções. Um programa feito em LabVIEW é dividido em duas partes: um diagrama de blocos e uma janela de interface de usuário (*Graphical User Interface - GUI*). A GUI é a janela onde o usuário final irá trabalhar, inserir e extrair dados. É a janela que fica exibida na tela enquanto o programa é executado. O diagrama de blocos é a janela de programação, onde serão definidas todas as relações entre as entradas e as saídas do VI. Na janela de diagrama de blocos é feita a programação propriamente dita, podendo incluir laços de repetição *for* ou *while*, estruturas de decisões, como *if*, entre outras opções.

O VI que foi criado para realizar os ensaios propostos trabalha com um recurso chamado “VISA Serial”, que permite leitura e escrita através da porta serial do PC. É através desse recurso que fazemos a leitura dos dados dos sensores, bem como a escrita de ordens de acionamento dos atuadores.

3.1 Programação em G

A programação de um VI é feita em linguagem ‘G’, que de acordo com (INSTRUMENTS, 2008) é uma linguagem gráfica de programação de fluxo de dados de alto nível. (SUMATHI; SUREKHA, 2007) aponta algumas vantagens da instrumentação virtual, relacionada com instrumentos tradicionais: melhor performance, menor custo e maior flexibilidade.

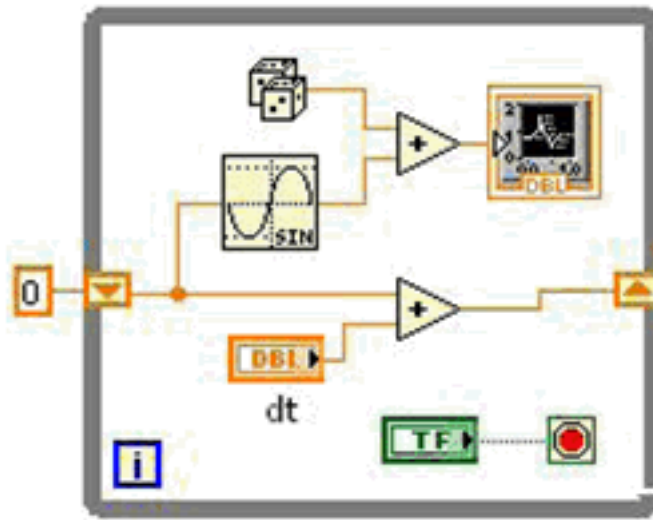


Figura 3.1: Diagrama de Blocos G. Fonte: (INSTRUMENTS, 2008)

A execução do programa é determinada pela estrutura de um diagrama de blocos gráfico, onde o programador conecta diferentes nós-funções desenhando fios de conexão. Os fios propagam variáveis e qualquer nó pode ser executado assim que todos seus dados de entrada se tornarem disponíveis. Sendo no caso de múltiplos nós simultâneos, a linguagem G é perfeitamente capaz de execução paralela. *Hardware* de multiprocessamento são automaticamente explorados pelo organizador interno, que multiplexa os barramentos do sistema operacional para os nós prontos para execução (SUMATHI; SUREKHA, 2007).

O diagrama de blocos mostrado na Figura 3.1 é um programa editado em uma janela de programação G, exemplo apresentado por (INSTRUMENTS, 2008). O programa mostra o resultado da soma de um sinal senoidal com um valor aleatório. Os dados de entrada e resultados são manipulados e mostrados na janela GUI, mostrada na Figura 3.2.

3.2 Estruturas de Execução

3.2.1 Estrutura de repetição *While*

O funcionamento de uma estrutura *While*, traduzido do inglês, a partir da documentação de ajuda oferecida pelo software LabVIEW, é dado como:

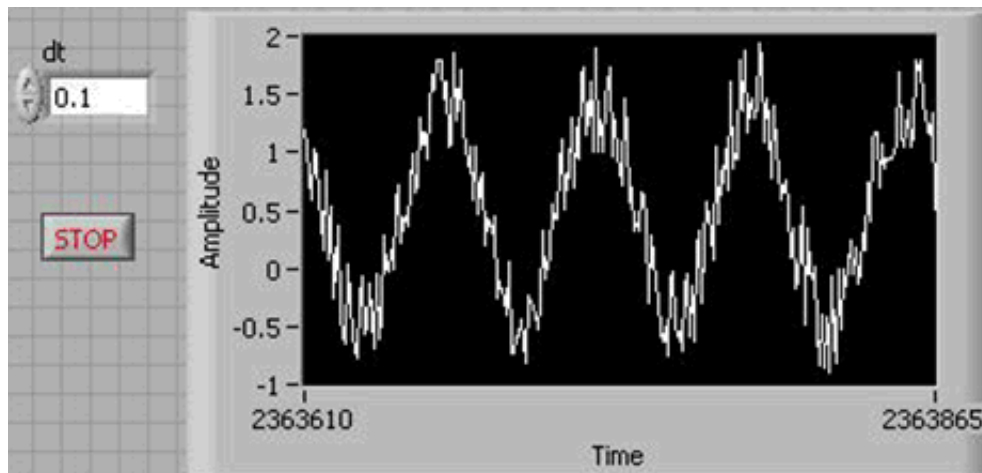


Figura 3.2: Interface de Usuário. Fonte: (INSTRUMENTS, 2008)

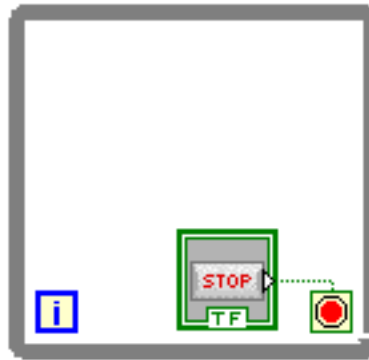


Figura 3.3: Bloco referente a uma estrutura de repetição While. Fonte: (LABVIEW, 2004)

Repete o subdiagrama em seu interior até que o terminal de condição, um terminal de entrada, receba um determinado valor Booleano. Este valor depende do comportamento de continuidade do *loop While*. Clicando no terminal de condição com o botão direito do *mouse*, existem as opções “*Stop if True*” (Parar se Verdadeiro) ou “*Continue if True*” (Continuar se Verdadeiro), ou seja, o usuário pode decidir a condição de continuidade do laço. O *loop While* é sempre executado ao menos uma única vez. O terminal de iterações “i” provê um contador de iterações, onde a saída é zero para a primeira iteração. O botão “*Stop*” (Parar) aparece conectado ao terminal de condição, dando ao usuário a opção de interromper a execução do *loop* a qualquer momento (LABVIEW, 2004).

Na Figura 3.3 é mostrada a representação de um *loop While* na janela de programação do LabVIEW.

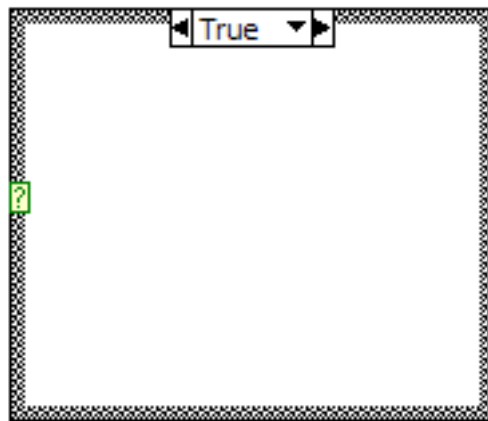


Figura 3.4: Bloco referente a uma estrutura de condição If. Fonte: (LABVIEW, 2004)

Nas nossas aplicações, a estrutura *While* foi utilizada para manter o programa fazendo leituras dos sensores repetidamente e para criar o sinal de controle do motor de passos, como será mostrado mais detalhadamente adiante.

3.2.2 Estrutura de execução condicional *If*

O funcionamento de uma estrutura *If*, traduzido do inglês, a partir da documentação de ajuda oferecida pelo software LabVIEW, se dá da seguinte maneira:

Possui um ou mais subdiagramas, ou casos, onde apenas um é executado a cada execução da estrutura. O valor ligado ao terminal seletor determina qual caso irá ser executado e pode ser um valor Booleano, uma “*string*”(valor em forma de caracteres) ou um número inteiro. Clicando com o botão direito do mouse na borda da estrutura é possível adicionar ou remover casos (LABVIEW, 2004).

A representação de uma estrutura *If* na janela de programação do LabVIEW está mostrada na Figura 3.4. Nós usamos estruturas *If* sempre contendo dois casos, “*True Case*” (Caso Verdadeiro) e “*False Case*” (Caso Falso), de modo que as instruções em seu interior seriam executadas baseadas no acontecimento ou não de determinados eventos. Essas estruturas foram empregadas, por exemplo, para temporizar a movimentação do motor de passos, para o ajuste de PWM e para alternar o resfriamento entre Ligado/Desligado. Essas operações serão mostradas detalhadamente mais adiante.

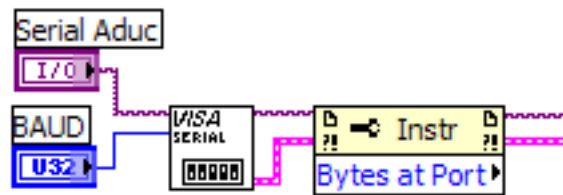


Figura 3.5: Conjunto de blocos utilizados para iniciar a comunicação serial.

3.3 Funções VISA (*Virtual Instrument Software Architecture*)

As funções VISA constituem uma camada de funções do LabVIEW que permitem a comunicação do software com outros dispositivos, sendo essa comunicação possível de ser realizada através da porta serial do PC (RS-232), GPIB e outras. Nós a utilizamos para fazer a comunicação serial do LabVIEW com o microcontrolador existente na plataforma de ensaios. Quatro funções VISA foram de nosso interesse: VISA Serial, VISA Read, VISA Write e VISA Close.

3.3.1 VISA Serial

Função usada para abrir a conexão serial. Foi colocada sempre no início dos programas, tornando possível o ajuste da comunicação serial. Para realizar esses ajustes, outro bloco foi adicionado junto ao VISA Serial, o nó de propriedades, também pertencente à biblioteca VISA. Com esse conjunto de blocos é possível escolher qual a porta do PC que deve ser utilizada, qual a taxa de comunicação serial, a quantidade de *bits*, a existência ou não de *bits* de paridade e *bit* de parada e o controle de fluxo. O conjunto possui dois terminais de saída, sendo um denominado “*Error Out*” (Erro de saída), onde é informado o erro ocorrido durante a comunicação, para que este seja repassado às próximas funções que utilizem a comunicação serial. O segundo terminal de saída é o “*VISA Resource Name*” (Nome do recurso VISA), que possui informações sobre qual porta serial está sendo usada; também é ligado a outros blocos de função VISA.

O trecho do diagrama de blocos onde a comunicação serial é estabelecida é mostrado na Figura 3.5, e os campos de ajuste disponíveis na janela de usuário estão mostrados na Figura 3.6. Em nossos programas, por motivo de simplicidade, apenas dois dos ajustes da porta serial estão disponíveis na janela de interação do usuário: a taxa de comunicação e a porta utilizada. O PC usado nos experimentos usava a porta COM2, e o microcontrolador da plataforma de ensaios está

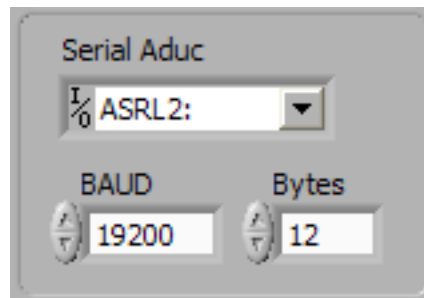


Figura 3.6: Janela de Usuário, campos de ajuste da comunicação serial.

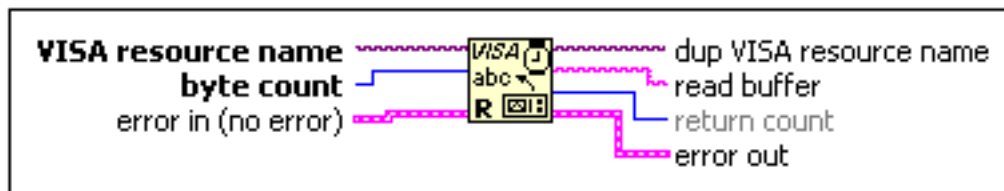


Figura 3.7: Bloco de leitura da porta serial. Fonte: (LABVIEW, 2004)

configurado para trabalhar com uma taxa de comunicação de 9600 *bits/s* até 19200 *bits/s*.

3.3.2 VISA Read

Abaixo segue o funcionamento de um bloco VISA *Read*, traduzido do inglês, a partir da documentação de ajuda oferecida pelo software LabVIEW:

Lê um determinado número de *bytes* (de acordo com a entrada “*bytes*”) do dispositivo ou interface especificado pela entrada “*VISA resource name*” e retorna os dados no terminal de saída “*read buffer*” (*buffer* de leitura). A operação só retorna quando a transferência termina (LABVIEW, 2004).

Essa função foi extremamente utilizada em nossos programas, pois através dela pudemos fazer a leitura de todos os dados vindos dos sensores presentes na plataforma. Este bloco está representado na Figura 3.7. As entradas e saídas de erro e de identificação da porta serial são conectadas a outros blocos VISA, partindo do VISA Serial, como mencionado anteriormente. A entrada de contagem de *bytes* a serem lidos é decidida pelo usuário, através do campo “*Bytes*”, mostrado na Figura 3.6. As operações feitas com os dados de saída desta função estão explicitadas adiante, quando falaremos sobre as leituras dos sensores.

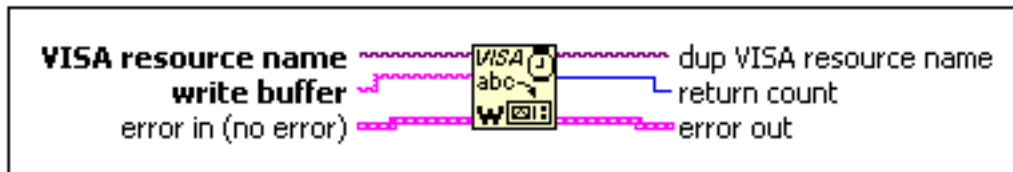


Figura 3.8: Bloco de escrita na porta serial. Fonte: (LABVIEW, 2004)

3.3.3 VISA Write

O funcionamento de um bloco *VISA Write* se dá da seguinte maneira, de acordo com a documentação de ajuda oferecida pelo software LabVIEW, traduzida do inglês:

Escreve o dado da entrada “*write buffer*” (*Buffer* de escrita) no dispositivo ou interface especificado pela entrada “*VISA resource name*”. A ligação do terminal de saída “*bytes count*” (contador de *bytes*) é opcional; este terminal retorna a quantidade de *bytes* que foram transferidos. A operação só retorna quando a transferência termina (LABVIEW, 2004).

Outra função que foi amplamente explorada em nossas aplicações, pois é através dela que fazemos a escrita de qualquer dado que deca chegar até o microcontrolador da plataforma. Por exemplo, para acionar o resfriamento, ou para configurar o *duty cycle* do PWM que controla o aquecimento do sistema. Este bloco está representado na Figura 3.8. As entradas e saídas de erro e de identificação da porta serial são conectadas a outros blocos VISA, partindo do VISA Serial, como mencionado anteriormente. A saída de contagem de *bytes* que foram escritos não está sendo utilizada por nós. Os dados a serem escritos na entrada “*write buffer*” dependem da operação desejada, e estão mais detalhados adiante, quando falaremos do acionamento dos atuadores.

3.3.4 VISA Close Serial Port

Abaixo segue o funcionamento de um bloco *VISA Close*, traduzido do inglês, a partir da documentação de ajuda oferecida pelo software LabVIEW:

Fecha a comunicação com um dispositivo, evento ou objeto especificado na entrada “*VISA resource name*”. Para cada sessão VISA que for aberta, deve haver um bloco de fechamento para finalizar a sessão quando esta chegar ao fim. Esta função aceita todas as classes disponíveis (LABVIEW, 2004).

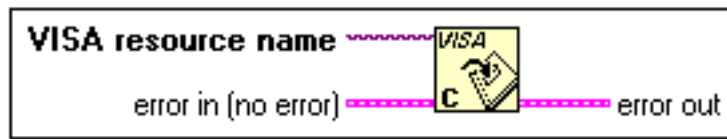


Figura 3.9: Bloco de fechamento de comunicação. Fonte: (LABVIEW, 2004)

Como diz a ajuda do *software*, este bloco cessa qualquer comunicação que tenha sido aberta em outro ponto do VI. Em nosso caso, sempre abrimos uma comunicação serial no início dos programas, por isso adicionamos sempre um bloco de fechamento no final dos programas, para finalizar a conexão serial, liberando a porta. A estrutura de um bloco VISA *Close* é mostrada na Figura 3.9.

3.4 Aquisição de dados

3.4.1 Tensão Mecânica

Para adquirir dados sobre a tensão mecânica sob a qual a amostra do fio está submetida, lançamos mão da informação que a plataforma é capaz de nos fornecer acerca da tração exercida sobre o fio. A plataforma disponibiliza essa informação através de um conversor A/D presente no microcontrolador. Para que essa informação chegue ao PC, é necessário que haja um “pedido” por parte do LabVIEW, através da serial. O microcontrolador recebe esse “pedido” e retorna pela serial um valor em milivolts (*mV*) correspondente à leitura do sensor de tração. Cabe ao programador do VI interpretar esse valor em milivolts e traduzí-lo na unidade desejada de tensão mecânica. O diagrama de blocos mostrado na Figura 3.10 representa a parte da programação em que o sinal de tração é adquirido e transformado de *mV* em megaPascal (*MPa*).

Os bloco de escrita VISA envia uma informação ao microcontrolador, “pedindo” o valor lido no pino 2 do conversor A/D; o bloco de leitura VISA faz a leitura na serial do valor de tração dado pelo sensor, em *mV*. Esse valor é lido como um texto (*string*), e por isso a presença de um bloco conversor, que converte os caracteres numéricos da *string* em um numero inteiro. Para transformar o valor em *mV* para um valor de tensão em *MPa*, tivemos que realizar uma série de operações. Primeiramente transformamos esse valor para tração em kilograma-força (*Kgf*), através de uma constante de relação linear do sensor de tração. Depois transformamos o valor em *Kgf* para

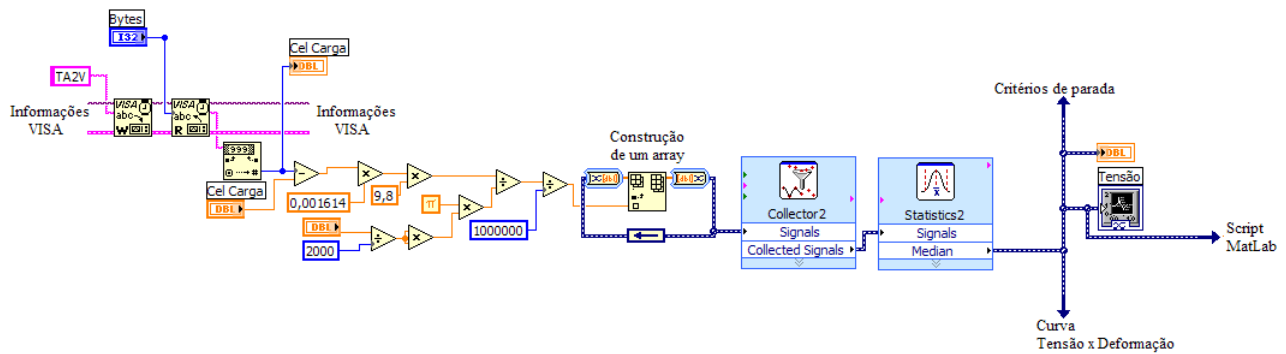


Figura 3.10: Diagrama de blocos para aquisição da tensão mecânica

| | |
|-----------------|---------------|
| SMA | |
| Comprimento (m) | Diametro (mm) |
| 0,765 | 0,28 |

Figura 3.11: Campos de entrada para os dados da amostra de SMA.

Newtons (N), através da relação na equação 3.1:

$$\text{Tração}(N) = \text{Tração}(Kgf) * 9,8 \quad (3.1)$$

Para transformar a tração em Newtons para tensão mecânica em MPa , usamos a equação 3.2:

$$\text{Tensão}(MPa) = \frac{\text{Tração}(N)}{\text{Área}(m)} * 10^3 = \frac{\text{Tração}(N)}{\pi * r^2(m)} * 10^3 \quad (3.2)$$

Ao fim, temos o valor de tensão mecânica de acordo com a área de secção transversal do fio. Essa área é calculada de acordo com o diâmetro do fio, que deve ser informado através dos campos mostrados na Figura 3.11. A saída após as operações matemáticas é o valor de tensão que queremos. Os blocos seguintes constituem uma espécie de filtragem digital do sinal, tratando-se da construção de um *array*, onde podemos armazenar as amostras coletadas, um coletor que agrupa 20 amostras e as envia para um bloco de tratamento estatístico. Este último recebe as 20 amostras do coletor e calcula a mediana das amostras, de modo que assim eliminamos eventuais amostras muito altas ou muito baixas, causadas por ruído no sinal.

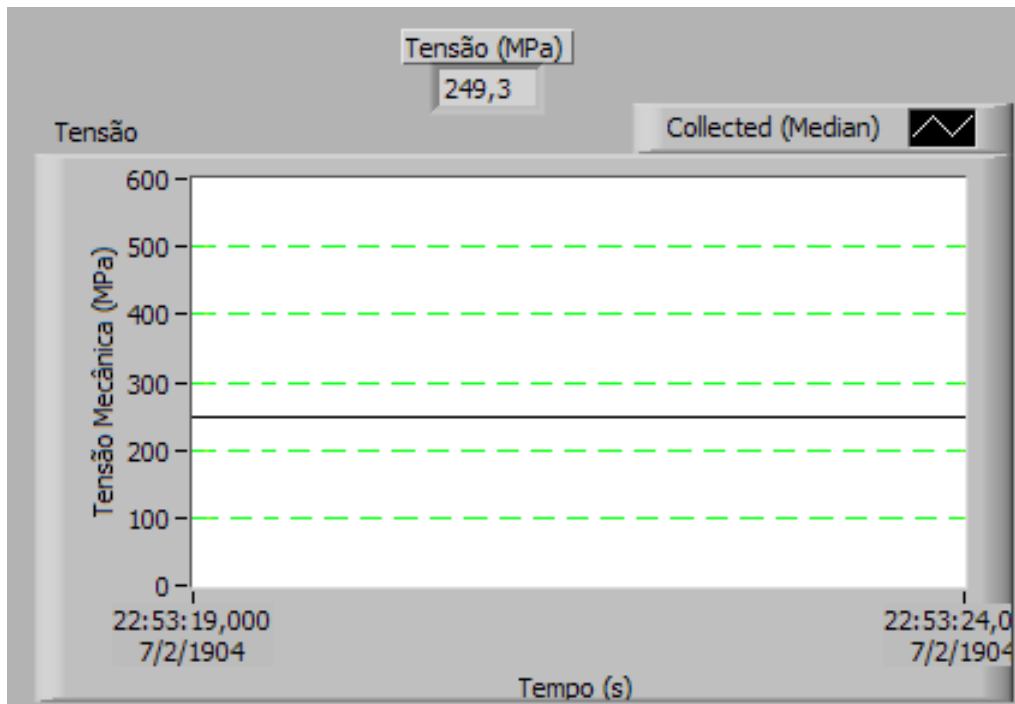


Figura 3.12: Curva da tensão mecânica vista na janela de usuário.

O sinal de tensão mecânica filtrado é então enviado para cinco blocos: um que irá analisar a tensão e decidir quando parar o ensaio, um campo mostrador inserido na janela GUI, um gráfico da tensão mecânica em função do tempo, uma curva de tensão mecânica x deformação linear e um *script* de interação com MatLab. O campo mostrador e o gráfico em função do tempo podem ser vistos na Figura 3.12; o envio para o *script* MatLab serve para armazenagem e será explicado na seção “Integração com o MatLab”. A obtenção dos dados para traçar uma curva Tensão Mecânica (MPa) x Deformação (%) é o nosso objetivo principal, e o traço desta curva está apresentado no capítulo “Resultados e Conclusões”.

3.4.2 Deformação Linear

Para adquirir dados sobre a deformação linear sofrida pela amostra do fio, lançamos mão da informação que a plataforma disponibiliza a informação através de um conversor A/D presente no microcontrolador. Para que essa informação chegue ao PC, é necessário que haja um “pedido” por parte do LabVIEW, através da serial. O microcontrolador recebe esse “pedido” e retorna pela serial um valor em milivolts (*mV*) correspondente à leitura do sensor de deformação. Cabe ao programa-

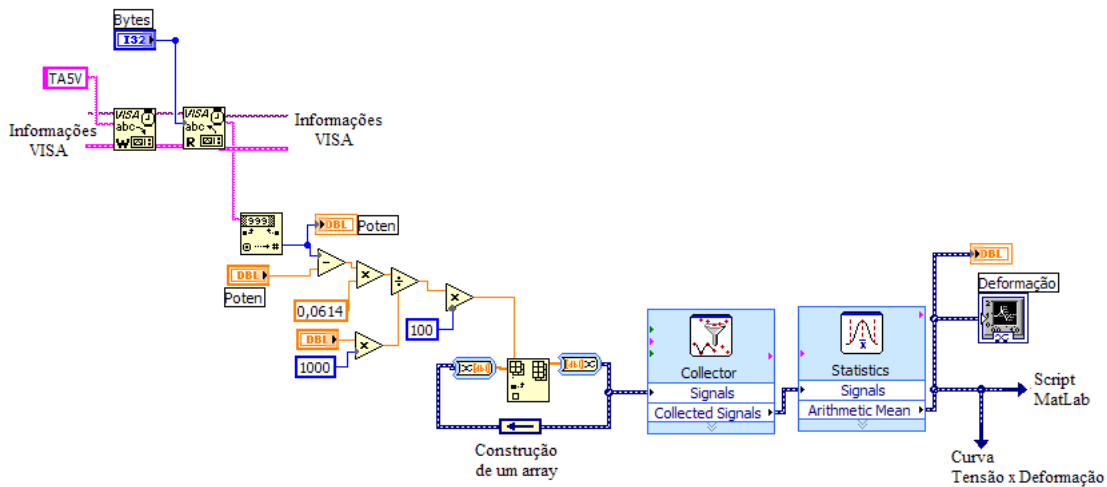


Figura 3.13: Diagrama de blocos para aquisição da deformação linear.

do VI interpretar esse valor em milivolts e traduzi-lo na unidade desejada de deformação linear. O diagrama de blocos mostrado na Figura 3.13 representa a parte da programação em que o sinal de deformação linear em valores percentuais é adquirido e transformado de mV em porcentagem de deformação (%).

Os bloco de escrita VISA envia uma informação ao microcontrolador, “pedindo” o valor lido no pino 4 do conversor A/D; o bloco de leitura VISA faz a leitura na serial do valor de deformação dado pelo sensor, em mV . Esse valor é lido como um texto (*string*), e por isso a presença de um bloco conversor, que converte os caracteres numéricos da *string* em um numero inteiro. Para transformar o valor em mV para um valor de deformação linear em porcentagem, tivemos que realizar uma série de operações. Temos a informação de que o sinal do sensor tem uma relação linear com a deformação linear sofrida pelo fio. Por isso, ao multiplicarmos o valor em mV por uma constante de relação linear do sensor de deformação, temos a leitura exata em milímetros (mm).

Então, para transformar de mm para valores percentuais, precisamos da informação do comprimento do fio, que é informado pelo usuário nos campos mostrados na Figura 3.11. Basta então que usemos a equação 3.3 e teremos o valor de saída em percentual.

$$Deformação(\%) = \frac{Deformação(mm)}{Comprimento(mm)} * 100 \quad (3.3)$$

A saída após as operações matemáticas é o valor de deformação que queremos. Os blocos

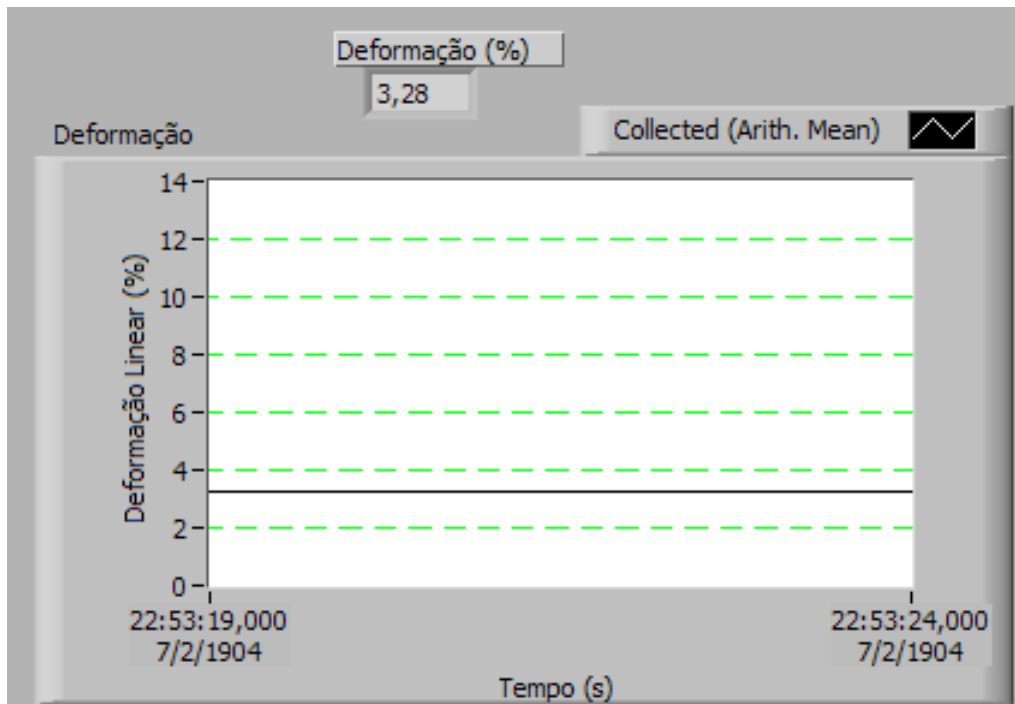


Figura 3.14: Curva da deformação linear vista na janela de usuário.

seguintes constituem uma espécie de filtragem digital do sinal, tratando-se da construção de um array, onde podemos armazenar as amostras coletadas, um coletor que agrupa 50 amostras e as envia para um bloco de tratamento estatístico. Este último recebe as 50 amostras do coletor e calcula a média das amostras, de modo que assim diminuimos a instabilidade que o sinal estava apresentando, causada por ruído no sinal.

O sinal de deformação linear percentual filtrado é então enviado para quatro blocos: um campo mostrador inserido na janela GUI, um gráfico da deformação linear em função do tempo, uma curva de tensão mecânica x deformação linear e um *script* de interação com MatLab. O campo mostrador e o gráfico em função do tempo podem ser vistos na Figura 3.12; o envio para o *script* MatLab serve para armazenagem e será explicado na seção “Integração com o MatLab”.

3.4.3 Temperatura

Para extrairmos as informações de temperatura do sistema, usamos um sensor de temperatura que está conectado ao conversor A/D presente no microcontrolador. Para que essa informação chegue ao PC, é necessário que haja um “pedido” por parte do LabVIEW, através da serial. O micro-

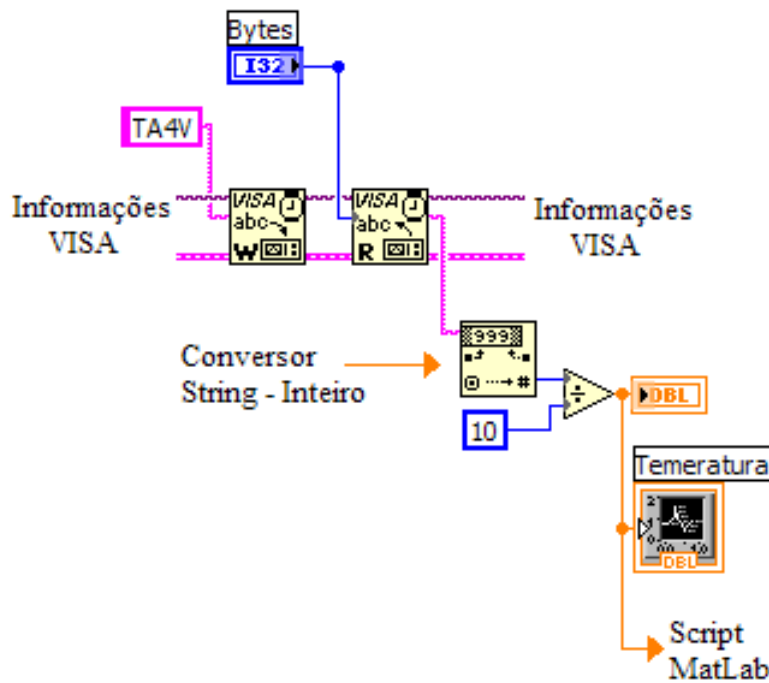


Figura 3.15: Diagrama de blocos da aquisição da temperatura.

controlador recebe esse “pedido” e retorna pela serial um valor em milivolts (mV) correspondente à leitura do sensor de temperatura. Cabe ao programador do VI interpretar esse valor em milivolts e traduzí-lo na unidade desejada de temperatura. O diagrama de blocos mostrado na Figura 3.15 representa a parte da programação em que o sinal de temperatura é adquirido e transformado de mV em graus Celsius ($^{\circ}C$).

Os bloco de escrita VISA envia uma informação ao microcontrolador, “pedindo” o valor lido no pino 4 do conversor A/D; o bloco de leitura VISA faz a leitura na serial do valor de temperatura dado pelo sensor, em mV . Esse valor é lido como um texto (*string*), e por isso a presença de um bloco conversor, que converte os caracteres numéricos da *string* em um numero inteiro. Temos a informação de que o sinal do sensor tem uma relação linear com a temperatura de medição, mais especificamente, de $10\ mV/^{\circ}C$. Por isso, ao dividirmos o valor em mV por 10, teremos a leitura exata em graus Celsius. A saída do divisor é o valor de temperatura que queremos, e é enviado para três blocos: um campo mostrador inserido na janela GUI, um gráfico da temperatura em função do tempo e um *script* de interação com MatLab. O campo mostrador e o gráfico em função do tempo podem ser vistos na Figura 3.16; o envio para o *script* MatLab serve para armazenagem e será explicado na seção “Integração com o MatLab”.

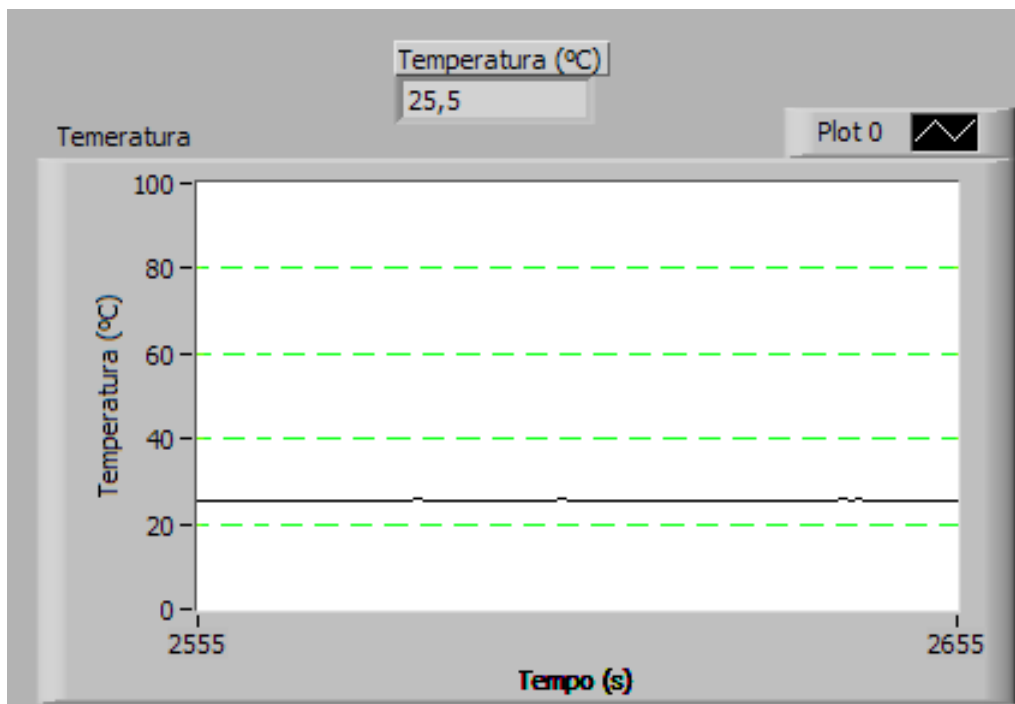


Figura 3.16: Curva de temperatura na janela do usuário.

3.5 Acionamento dos Atuadores

3.5.1 Acionamento do motor de passo

Para aplicar tensão mecânica sobre a amostra da liga, a plataforma usa um motor de passos, controlado por um sinais enviados pelo microcontrolador. São dois os sinais que controlam a movimentação do motor e estes sinais deve ser produzido via *software*.

O primeiro sinal controla o sentido de rotação do motor e pode assumir dois valores: um valor em nível lógico alto para fazer o motor girar no sentido horário ou um em nível lógico baixo para fazer o motor girar em sentido anti-horário. Para produzi-lo fizemos o LabVIEW enviar comandos via serial para o microcontrolador, comandos estes que fazem um pino de uma porta digital do microcontrolador permanecer no nível alto ou baixo de tensão, de acordo com o comando enviado.

O segundo sinal controla a movimentação do motor, através dos passos. O sinal a ser gerado deve obedecer o modelo mostrado na Figura 3.17. O sinal permanece em nível alto de tensão, e para cada pulso negativo dado no sinal, o motor avança um passo. O sinal deve permanecer no mínimo $4,5\mu s$ entre um pulso e outro para que o motor interprete pulsos diferentes. O sinal deve

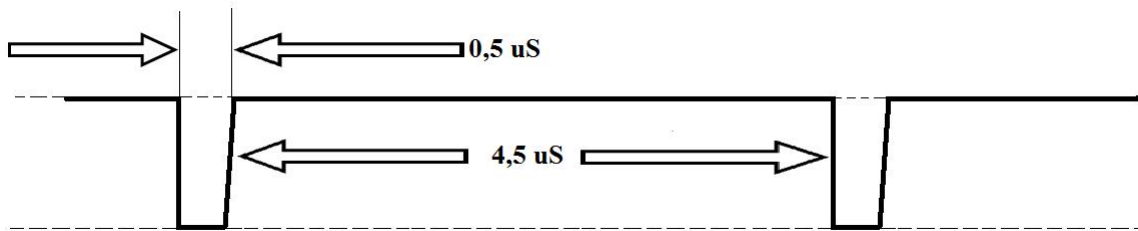


Figura 3.17: Modelo do sinal de controle que deve ser enviado para o microcontrolador.

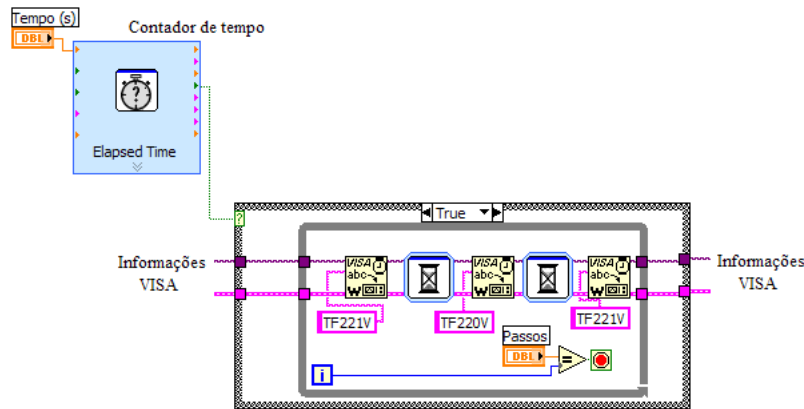


Figura 3.18: Diagrama de blocos que controla a movimentação do motor de passos.

permanecer no mínimo $0,5\mu s$ em nível baixo para caracterizar um pulso. Para gerar o sinal usamos o mesmo mecanismo que usamos para gerar o primeiro sinal de controle, através das portas digitais do microcontrolador, enviando comandos para alternar o nível do sinal da porta de alto para baixo e vice-versa. O bloco construído para fazer com que o motor avance está mostrado na Figura 3.18.

Como nosso programa foi construído para realizar automaticamente o ensaio de tração, ele deve enviar automaticamente pulsos para fazer o motor girar com velocidade fixa no tempo. No caso, usamos o bloco “*Elapsed Time*” (Tempo decorrido) para controlar o envio de pulsos com intervalos de tempo iguais. O controle do motor está dentro de uma estrutura *While*, para que o motor avance uma quantidade determinada de passos, escolhida pelo usuário. A quantidade de passos avançados pelo motor é limitado pelo contador de iterações do *While*. Podem ser vistos na Figura 3.19 os campos da janela de usuário, onde este pode decidir o tempo entre os avanços do motor, e a quantidade de passos que ele deve avançar por vez e um campo de tensão mecânica máxima a ser imposta à amostra de SMA. Isso significa que a plataforma irá aplicar tensão até este valor escolhido, depois irá iniciar o processo de descarregamento, onde o motor gira no sentido

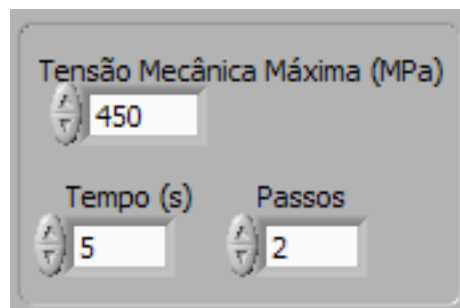


Figura 3.19: Campos de entrada de controle do motor, na janela de usuário.

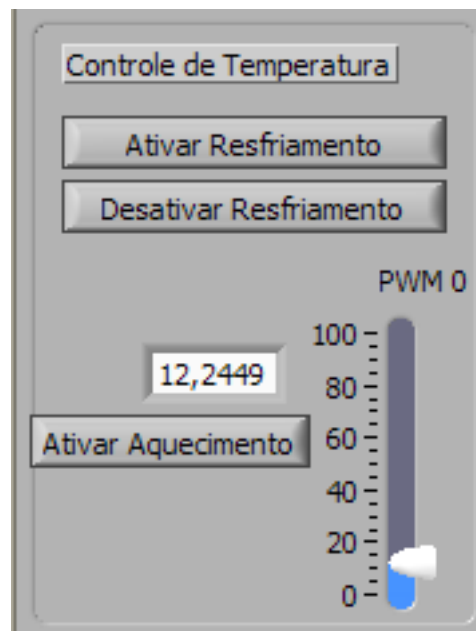


Figura 3.20: Campos de entrada para controle de temperatura.

de liberar a tensão aplicada, que começará a diminuir. Quando a tensão mecânica chega a zero, o motor pára, e podemos visualizar a deformação residual que ficou na amostra do fio. No caso da Figura 3.19, o usuário tinha interesse de verificar o comportamento do fio nas tensões de 0 a 450 *MPa*, fazendo o motor andar dois passos a cada cinco segundos. A estrutura que faz com que o motor inverta o sentido de rotação é o bloco que chamamos de critério de parada, que foi mencionado na seção de leitura do sensor de tração. É por isto que o sinal de tensão mecânica tem um terminal sendo conectado ao critério de parada.

3.5.2 Controle de Temperatura

Ativação do resfriamento

O usuário tem duas opções no que diz respeito ao resfriamento do sistema. São os dois botões vistos na Figura 3.20, “Ativar Resfriamento” e “Desativar Resfriamento”. Ao clicar no botão “Ativar Resfriamento”, um comando é enviado pela serial através da função *VISA Write*, ordenando que um pino de uma porta digital do microcontrolador saia do nível baixo de tensão e passe para o nível alto. Este pino está conectado a um transistor que faz o acionamento dos atuadores de resfriamento.

Para cessar o resfriamento, o usuário clica no botão “Desativar Resfriamento”, o que faz com que o LabVIEW envie um comando ao microcontrolador, analogamente ao funcionamento de ativação do resfriamento, com a diferença que o comando enviado faz com que o mesmo pino da mesma porta digital do microcontrolador passe do nível alto de tensão para o nível baixo, de modo que a tensão na base do transistor seja igual a zero, desativando o funcionamento dos atuadores.

Ativação do Aquecimento

Para aquecer a amostra a ser estudada, a plataforma utiliza uma resistência elétrica de 300 W produzida especialmente para funcionamento em imersão. Isso porque a amostra da LMF está imersa em um fluido de silicone, que facilita a distribuição uniforme de temperatura ao longo do recipiente em que ela encontra-se inserida. Essa resistência elétrica é alimentada por uma fonte variável de 0 a 230 V_{AC} , onde a tensão de saída é determinada de acordo com o *duty cycle* de um sinal de PWM (*Pulse Wide Modulation*).

O microcontrolador é capaz de gerar um sinal de PWM, então nossa função é enviar ao microcontrolador a informação do valor do *duty cycle* que desejamos. De modo que, ao controlarmos o *duty cycle*, controlamos a tensão que alimenta a resistência, controlando então a corrente que a atravessa, e conseqüentemente controlando o calor gerado pelo efeito Joule na resistência elétrica. É dessa forma que podemos aumentar ou diminuir a quantidade de calor a ser inserida no fluido de silicone e conseqüentemente na amostra de SMA.

Os campos de entrada onde o usuário pode ajustar o valor do *duty cycle* estão ilustrados na Figura 3.20. Uma barra deslizante pode variar o valor de 0 a 100 %, e ainda, esse valor pode ser

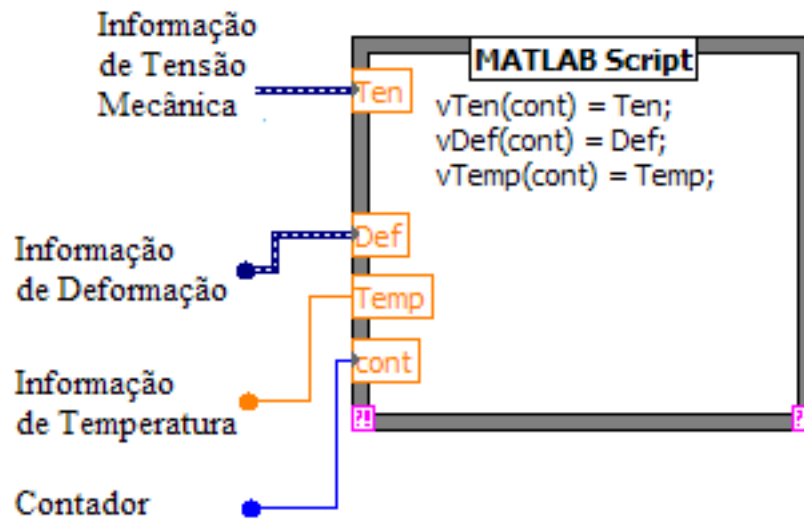


Figura 3.21: Construção de três vetores a partir de dados vindos das leituras dos sensores.

inserido com exatidão no campo à esquerda da barra.

A programação feita em LabVIEW para enviar essa informação para o microcontrolador foi simples. Um bloco recebe o valor do campo de entrada do *duty cycle* e converte esse número em uma *string*. Essa *string* é então convertida em um comando e enviado pela porta serial através da função VISA *Write*, já mencionada anteriormente. Assim o PWM fica controlado pelo usuário na janela GUI.

3.6 Integração com MatLab

Após ter sido realizado o ensaio, é de interesse que os dados obtidos durante o experimento possam ser armazenados para verificação posterior. Isto é feito usando uma ferramenta do LabVIEW que permite uma integração deste *software* com um outro chamado MatLab (*Matrix Laboratory*). O MatLab é uma poderosa ferramenta criada pela empresa MathWorks. O programa permite a armazenagem dos dados em *arrays* e também nos permite plotar gráficos em duas ou três dimensões. O bloco em LabVIEW que faz a interação com o MatLab é o “MatLab Script”. Observa-se na Figura 3.21 um bloco que projetamos para construir os *arrays* de deformação, temperatura e tensão mecânica.

Em nosso programa, usamos o MatLab Script para recolher os dados de todas as amostras

obtidas durante o ensaio, separados em três *arrays*: deformação, tensão e temperatura. Ao fim do ensaio, programamos o *script* para executar duas ações: a primeira é plotar um gráfico tridimensional contendo as três grandezas envolvidas durante os ensaios: temperatura, tensão mecânica e deformação; um gráfico como este está mostrado no capítulo “Resultados e Conclusões”. A segunda ação do MatLab Script é criar um arquivo contendo todas as informações recolhidas, separadas nestes três *arrays*. Este arquivo pode ser consultado a qualquer momento, mesmo depois que o LabVIEW é fechado. Outra grande vantagem deste processo é a possibilidade de se transportar este arquivo para diferentes computadores, para que os resultados possam ser discutidos em quaisquer lugares que não o laboratório onde foi feito o experimento.

4 *Resultados e Conclusões*

Durante o período de estágio, foi desenvolvido um VI no LabVIEW capaz de realizar automaticamente um ensaio de tração de amostras de Liga com Memória de Forma, extraíndo sua curva Tensão Mecânica (*MPa*) x Deformação Linear Percentual (%), e expondo o resultado num gráfico tridimensional, onde a terceira dimensão foi adotada como sendo a temperatura de operação do ensaio.

Como resultados, apresentamos quatro entre as curvas obtidas, sendo duas correspondentes a ensaios com uma amostra em forma de fita, e as duas restantes correspondendo a ensaios com a amostra em forma de fio. A seqüência de procedimentos que foram executados para realizar cada um dos experimentos foi a seguinte:

- Medição do comprimento e da área de secção transversal da amostra;
- Fixação da amostra na plataforma;
- Ensaio de tração com a amostra;
- Armazenagem dos dados do ensaio;
- Comparação das curvas obtidas com curvas encontradas na literatura.

RESULTADOS PARA A AMOSTRA DE FIO COM SECÇÃO CIRCULAR:

Comprimento: 931 *mm*

Área de secção transversal: 0,05 *mm*²

Mostrada na Figura 4.1, está uma curva Tensão Mecânica (*MPa*) x Deformação Linear (%) obtida nos experimentos com a primeira amostra. Nas Figuras 4.2 e 4.3 estão expostas curvas

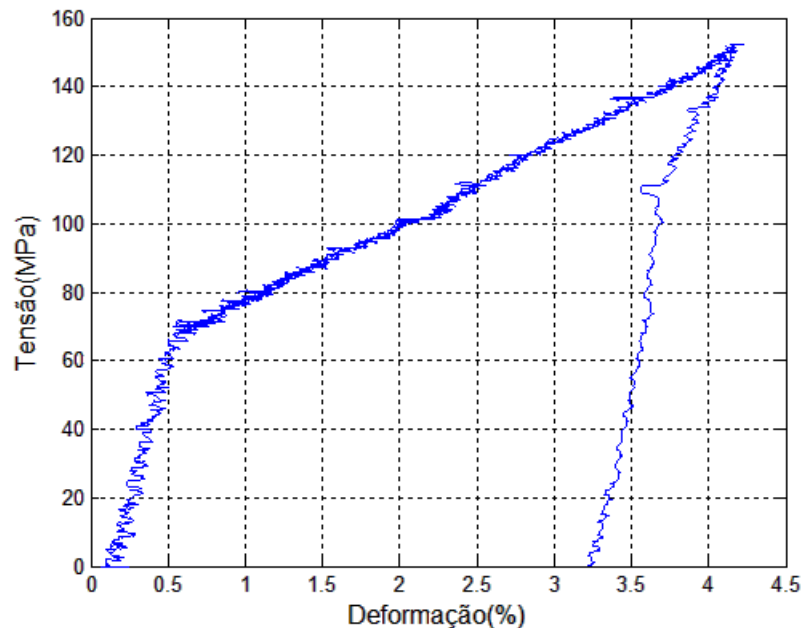


Figura 4.1: Curva Tensão mecânica x Deformação obtida no ensaio com um fio de SMA.

semelhantes encontradas na literatura, sendo as fontes: (NETO; ARAÚJO, 2008) e (HODGSON; WU, 2002), respectivamente.

RESULTADOS PARA A AMOSTRA EM FORMA DE FITA DELGADA:

Comprimento: 255 mm

Área de secção transversal: 0,06 mm²

No caso desta segunda amostra, fizemos dois experimentos consecutivos, a fim de verificar a repetibilidade do sistema. Então adicionamos três procedimentos àqueles mencionados no início deste capítulo, ficando então a sequência de procedimentos da seguinte maneira:

- Medição do comprimento e da área de secção transversal da amostra;
- Fixação da amostra na plataforma;
- Ensaio de tração com a amostra;
- Armazenagem dos dados do ensaio;
- Aquecimento da amostra, para que esta retorne à fase inicial;

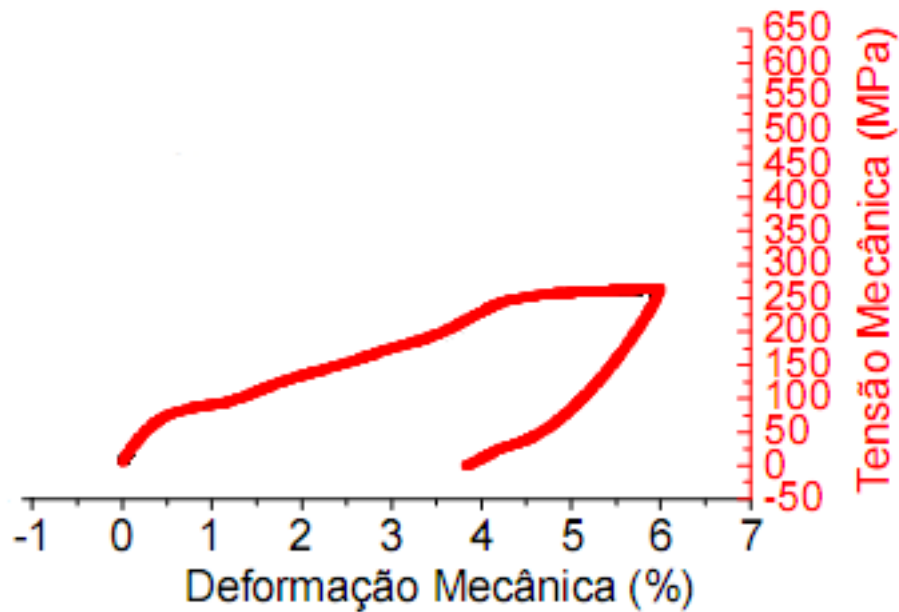


Figura 4.2: Curva Tensão mecânica x Deformação encontrada na literatura para um fio de SMA.
Fonte: (NETO; ARAÚJO, 2008)

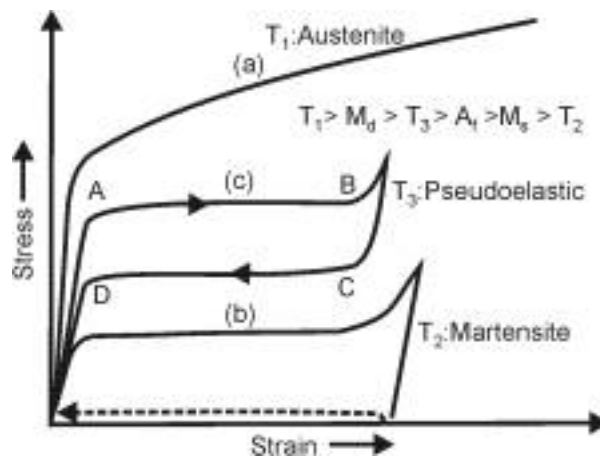


Figura 4.3: Curva Tensão mecânica x Deformação encontrada na literatura para um fio de SMA.
Fonte: (HODGSON; WU, 2002)

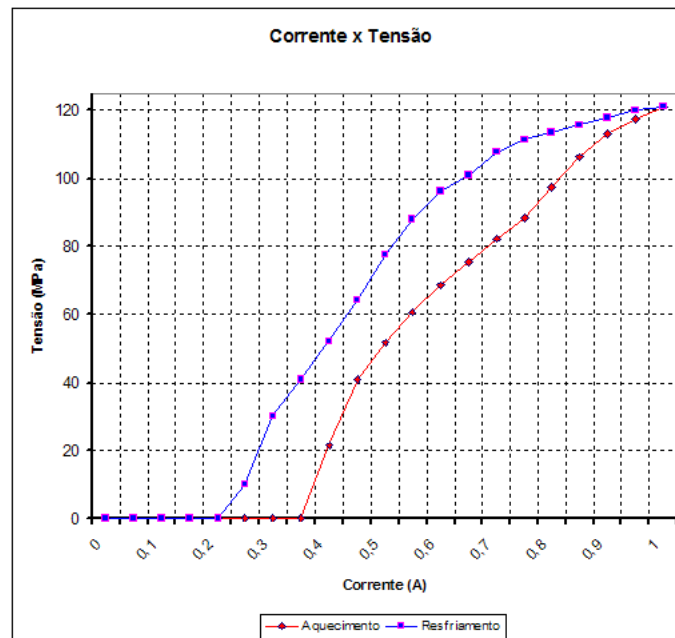


Figura 4.4: Gráfico para determinar a temperatura de transformação de fase.

- Repetição do experimento, obtendo nova curva Tensão (MPa) x Deformação (%);
- Armazenagem dos novos dados;
- Comparação das curvas obtidas com curvas encontradas na literatura.

Antes de realizar os ensaios com esta amostra, fizemos um pequeno experimento para determinar a temperatura que deveria atingir a amostra para que esta retornasse ao estado inicial. Isto não foi feito com a primeira amostra porque já possuíamos esta informação. Para aquecer a amostra neste experimento extra nós usamos o princípio do efeito Joule diretamente sobre o fio, aplicando corrente em suas extremidades. Então, plotamos um gráfico de corrente elétrica x tensão mecânica, com a deformação fixa, ou seja, com o fio firmemente preso. Analisamos o gráfico encontrado e observamos que a certa temperatura a tensão mecânica inicia um processo de saturação, indicando que a transformação de fase já ocorreu. Esta temperatura onde a tensão mecânica satura é a temperatura de transformação que deveríamos atingir para fazer a amostra retornar ao estado inicial entre um ensaio e outro. Na Figura 4.4, encontra-se o gráfico discutido neste parágrafo. Percebemos que a amostra saturou em 0,9 A.

Mostradas nas Figuras 4.5 e 4.6, estão as curvas Tensão Mecânica (MPa) x Deformação Linear

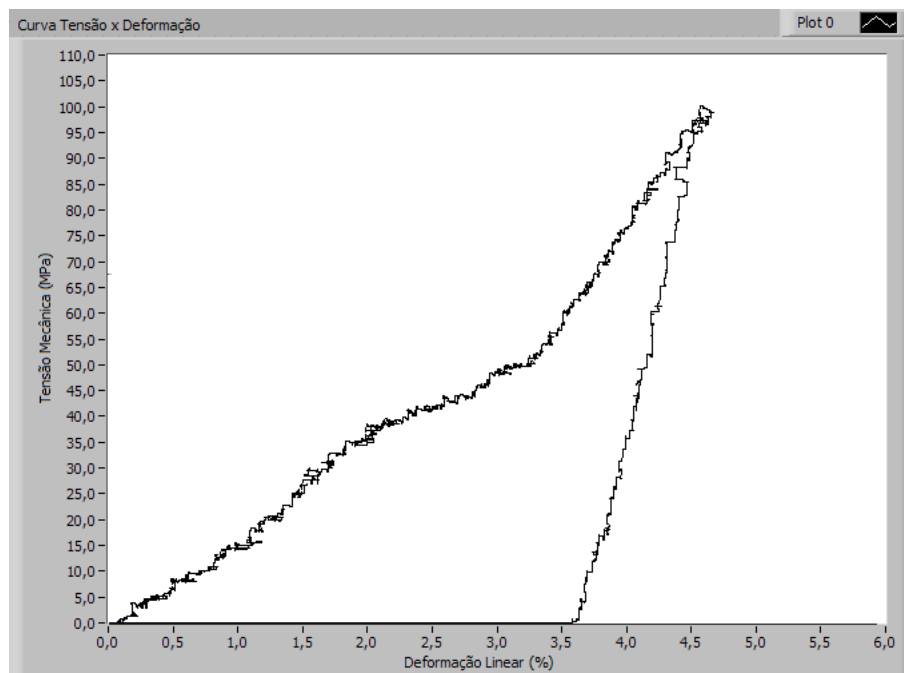


Figura 4.5: Curva Tensão mecânica x Deformação obtida no primeiro ensaio com a fita delgada.

(%) obtidas nos experimentos com a amostra de LMF em forma de fita delgada, no primeiro e no segundo ensaios, respectivamente. Na Figura 4.7 está exposta uma curva semelhante encontrada na literatura, sendo a fonte: (NETO; ARAÚJO, 2008).

O VI que foi discutido exerceu a função para a qual foi projetado, que era de aplicar tensão sobre a amostra até um valor estabelecido pelo usuário, detectar quando este valor fosse alcançado, e então liberar a tensão gradualmente até atingir valor zero, sempre obtendo os dados de temperatura, tensão mecânica e deformação da amostra, e plotando em tempo real uma curva Tensão Mecânica (MPa) x Deformação Linear (%). O programa também dá a opção de ativar/desativar o resfriamento ou o aquecimento do meio onde se encontra a amostra, tornando possível a realização dos ensaios a diferentes temperaturas. Ao final da execução, é feito um arquivo de saída, contendo todas as informações obtidas, que permite consultas futuras; bem como é exibido o gráfico tridimensional proposto, como o ilustrado na Figura 4.8, contendo três eixos: Temperatura ($^{\circ}\text{C}$), Tensão Mecânica (MPa) e Deformação (%).

Concluimos então que o funcionamento do programa foi satisfatório, e que ao comparar as curvas obtidas experimentalmente com aquelas vistas na literatura e aqui expostas, os ensaios foram realizados com sucesso, obtendo resultados igualmente satisfatórios, dada a semelhança encontrada

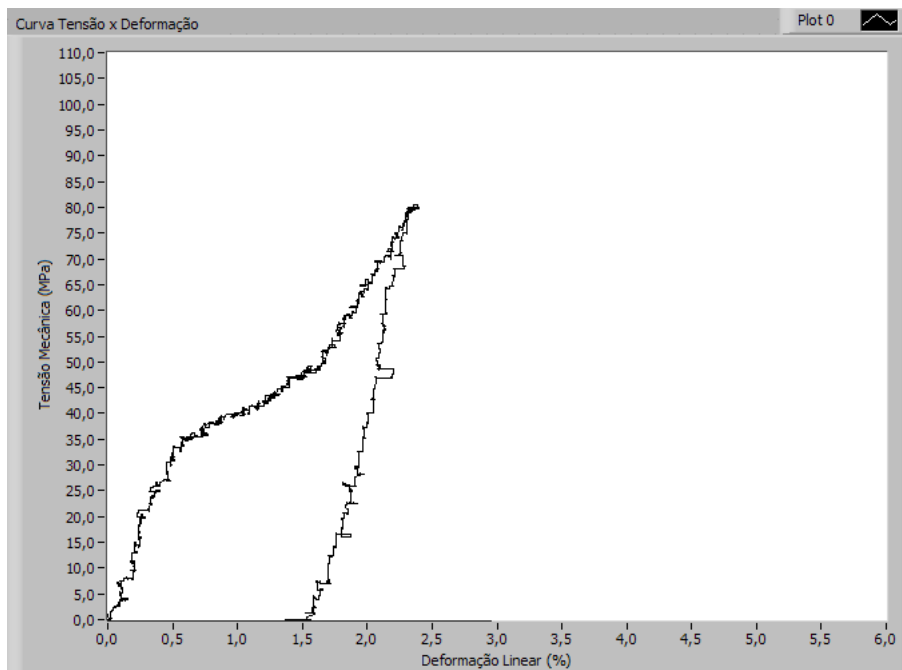


Figura 4.6: Curva Tensão mecânica x Deformação obtida no segundo ensaio com a fita delgada.

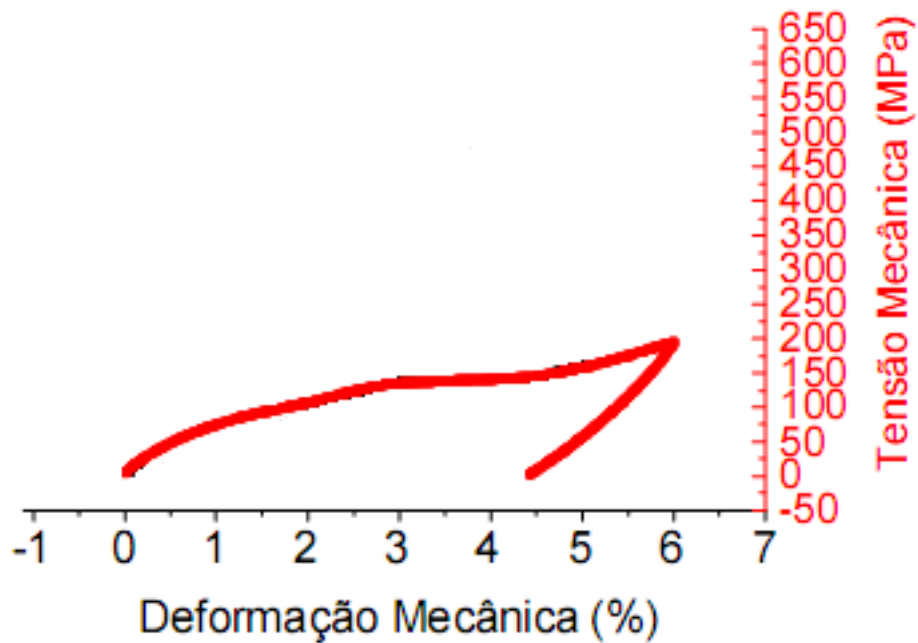


Figura 4.7: Curva Tensão mecânica x Deformação encontrada na literatura para fita delgada. Fonte: (NETO; ARAÚJO, 2008).

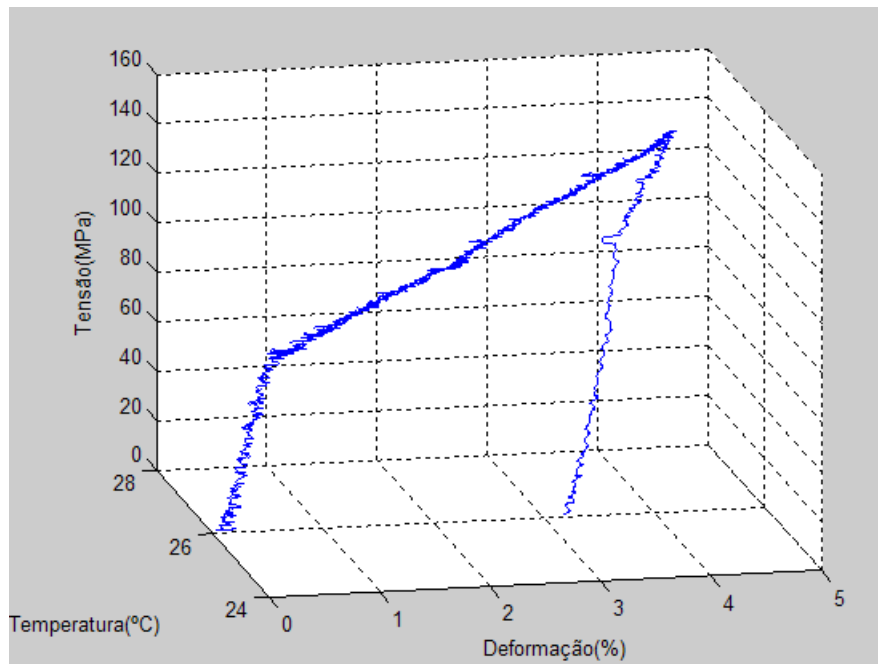


Figura 4.8: Gráfico tridimensional gerado pelo MatLab.

nas curvas. Como sugestão para trabalhos futuros, fica a possibilidade de projeto e implementação de um controlador de temperatura rigoroso, a exemplo de um controlador PD ou um PI, que poderá adicionar mais uma funcionalidade poderosa ao programa desenvolvido, permitindo testes mais amplos e eficientes.

Bibliografia

HODGSON, D. E.; WU, M. H. *Shape Memory Alloys*. 2002.

INSTRUMENTS, N. *Introdução à Programação G*. 2008.

[Http://digital.ni.com/worldwide/brazil.nsf/web/all/CE9ADD7916BC507E8625757D006CFB27](http://digital.ni.com/worldwide/brazil.nsf/web/all/CE9ADD7916BC507E8625757D006CFB27).

LABVIEW, N. I. *Ajuda do Software LabVIEW*. 2004.

NETO, J. F. C.; ARAÚJO, C. J. de. Estudo experimental do comportamento eletro-termomecânico de atuadores lineares de niti com memória de forma. *Congresso Nacional de Engenharia Mecânica - CONEM*, Agosto 2008.

SUMATHI, S.; SUREKHA, P. *LabVIEW based Advanced Instrumentation Systems*. 1º. ed. New York: Springer, 2007. ISBN 10 3-540 48500-7.

VALENZUELA, W. A. *Sistema de articulação atuado por meio de liga com memória de forma*. Dissertação (Mestrado) — Universidade Federal de Campina Grande, Campina Grande, 2005.

VALENZUELA, W. A. *Modelagem e Simulação de um Atuador de Liga Metálica com Memória de Forma*. Tese (Projeto de Pesquisa) — Universidade Federal de Campina Grande, Campina Grande, Maio 2007.