



Universidade Federal do Rio Grande
Instituto de Matemática, Estatística e Física
Programa de Pós-Graduação em Física

Lucas Heck dos Santos

CONTROLE E AUTOMAÇÃO DE UM SISTEMA DE MEDIDAS DE PROPRIEDADES ELÉTRICAS

Rio Grande

2018

LUCAS HECK DOS SANTOS

**CONTROLE E AUTOMAÇÃO DE UM SISTEMA DE
MEDIDAS DE PROPRIEDADES ELÉTRICAS**

Dissertação apresentada ao Instituto de Matemática, Estatística e Física da Universidade Federal do Rio Grande para obtenção do título de Mestre em Física pelo Programa de Pós-graduação em Física.

Orientador: Professor Dr. Jorge Luiz Pimentel Junior

Rio Grande
2018

Ficha catalográfica

S237c Santos, Lucas Heck dos.

Controle e automação de um sistema de medidas de propriedades elétricas / Lucas Heck dos Santos. – 2018.
74 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Física, Rio Grande/RS, 2018.

Orientador: Dr. Jorge Luiz Pimentel Junior.

1. Resistividade Elétrica 2. Automatização 3. Controle de Temperatura I. Pimentel Junior, Jorge Luiz II. Título.

CDU 53:681.5

Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

Dedico este trabalho à minha família.

Sumário

RESUMO.....	5
ABSTRACT	6
INTRODUÇÃO	7
1 - PROPRIEDADES ELÉTRICAS	8
1.1 – TEORIA DE FENÔMENOS DE TRANSPORTE.....	8
1.2 – RESISTIVIDADE ELÉTRICA	11
1.3 – TÉCNICAS DE MEDIDA DE RESISTIVIDADE ELÉTRICA	15
1.3.1 – Método de Corrente Contínua (DC)	15
1.3.2 – Método de Corrente Alternada (AC).....	16
1.3.3 – Método da Curva Características I-V (Corrente-Voltagem)	16
2 – PROCEDIMENTOS EXPERIMENTAIS E COMPUTACIONAIS	18
2.1 – SISTEMA TÉRMICO	18
2.1.1 – Criostato	18
2.1.2 – Procedimento de Vácuo	20
2.2 – ELETRÔNICA ASSOCIADA	21
2.2.1 – Resistômetro Diferencial.....	22
2.2.2 – Controlador de Temperatura.....	23
2.2.3 – Ações de Controle do Aquecedor.....	24
2.2.4 – Controlador PID.....	30
2.2.5 – Métodos de Sintonia.....	31
2.3 – AQUISIÇÃO DOS DADOS/INSTRUMENTAÇÃO	35
2.3.1 – Camada dos Barramentos.....	37
2.3.2 – Camada de Abstração de Hardware.....	43
2.3.3 – SCPI (Standard Commands for Programmable Instruments)	44
2.3.3 – Camada de Controle(Driver) do Instrumento.....	48
2.3.4 – Camada da Aplicação.....	50
2.4 – AUTOMAÇÃO	50
2.4.1 – A Linguagem de Programação	50
2.4.2 – Interação Software-Equipamento	52
2.4.3 – Desenvolvimento do software	54
2.4.5 – Experimentos	56
3 – RESULTADOS	59
3.1 – CALIBRAÇÃO DO PID	59
3.2 – RESULTADOS DO PID	64
3.3 – AMOSTRA PADRÃO DE COBRE.....	66
3.4 – CALIBRAÇÃO DO TERMÔMETRO	68
4 – CONCLUSÃO E PERSPECTIVAS.....	71
REFERÊNCIAS	72

RESUMO

Este trabalho apresenta a montagem, o controle e automação de um sistema de medidas de propriedades elétricas de amostras sólidas em ambiente criogênico. O sistema permite determinar a resistividade elétrica de materiais em um intervalo de 80K a 300K. Foi desenvolvido um *software*, utilizando a linguagem de programação Python, com rotinas que automatizam o experimento. Para validação do sistema, foi realizada a medida da resistividade de uma amostra padrão de cobre. Realizamos também a calibração de um termômetro de platina, que poderá ser utilizado em outras montagens experimentais. A construção deste equipamento de medidas de transporte elétrico possibilitará a realização de diversos estudos referentes à caracterização elétricas de materiais no Laboratório de Supercondutividade, Magnetismo e Materiais Avançados do IMEF/FURG, contribuindo para os trabalhos experimentais da linha de pesquisa em Física da Matéria Condensada do Programa de Pós Graduação em Física da FURG.

Palavras-chave: resistividade elétrica, automatização, controle de temperatura.

ABSTRACT

In this work we present the mounting, the control and automation of a system that measures electrical properties from solid samples in a cryogenic ambient. The system allows to determine the electrical resistivity of materials in an interval from 80K to 300K. With the programming language Python we developed a software that automatize the experiment. For validation of the system, the resistivity of a standard sample of copper has been measured. We also calibrated a platinum thermometer, which could be used in another experiment montages. The construction of this equipment allows the study and electrical characterization of several material in the Laboratory of Superconductivity, Magnetism and Advanced Material of IMEF/FURG, contributing to the experimental work in Condensed Matter Physics in the Federal University of Rio Grande.

Keyword: electrical resistivity, automation, temperature control.

INTRODUÇÃO

O estudo de materiais proporciona a possibilidade de desenvolvimento de novas tecnologias e compreende importantíssima área de estudo. Devido a isto é necessária a realização de uma série de experimentos acerca das propriedades dos materiais com o intuito de caracterizá-los. Neste trabalho, propomos uma montagem experimental de um sistema de medidas de propriedades elétricas em baixas temperaturas cujo controle é feito por um *software*. Para validação da montagem realizamos a medida da resistividade elétrica de uma amostra de cobre. Deste resultado calibramos um termômetro de platina para medida da temperatura em outros experimentos.

O controle da montagem permite a realização de variação controlada da temperatura, necessário para precisão das medidas. Essa precisão garante que haja estabilidade no sistema, assegurando a validade da caracterização elétrica. Com isto, é possível estudar diversos fenômenos, como a transição para o estado supercondutor, que ocorre quando a resistividade de alguns materiais se anula ao atingir uma determinada temperatura crítica.

Para automatizar o processo de coleta de dados desenvolvemos um *software*, utilizando a linguagem Python, para estabelecer comunicação com os instrumentos da montagem através de uma interface GPIB. Assim diminuimos a possibilidade de erro humano prejudicar o experimento, introduzimos métodos no *software* que auxiliam na medida e no ajuste de parâmetros importantes na configuração do sistema.

No capítulo 1 são introduzidos conceitos referentes à teoria de transporte elétrico em materiais sólidos, à teoria da resistividade e as técnicas de medidas

No capítulo 2 é apresentada a montagem experimental e o procedimento para o desenvolvimento do *software*.

No capítulo 3 são apresentados e analisados os resultados do ajuste do PID, da calibração do termômetro de platina e da medida da resistividade elétrica de uma amostra padrão de cobre.

No capítulo 4 são apresentadas as conclusões e perspectivas futuras deste trabalho

1 - Propriedades Elétricas

Neste capítulo iremos apresentar a teoria e os métodos utilizados para a caracterização eletrônica de materiais.

1.1 – Teoria de Fenômenos de Transporte

Proposta em 1900, apenas três anos após a descoberta do elétron por J. J. Thomson, a teoria de Drude (KITTEL, 2000) foi uma das primeiras a ser capaz de explicar o comportamento dos elétrons em um sólido, desenvolvendo uma eficaz teoria para a condução térmica e elétrica nos materiais, em especial nos metais.

A teoria parte da consideração dos elétrons livres presentes no material como um ‘gás de elétrons’, enquanto os íons são considerados estáticos e impenetráveis. Na ausência de campo magnéticos os elétrons devem se mover com base nas leis de Newton, resultante em deslocamento aleatório no material com velocidade térmica média \vec{v} de aproximadamente 10^5 m/s (SRIVASATAVA, 2014). Independentemente do número de elétrons no material a única interação possível tem origem nas colisões elásticas via elétron-íon. Com a aplicação de um campo elétrico \vec{E} a direção de movimento dos elétrons passa a ser especificada, contrária ao campo, dando o surgimento de uma densidade de corrente elétrica \vec{J} descrita pelo somatório do deslocamento de todos os elétrons livres como

$$\vec{J} = \sum -e[\vec{v} - \vec{v}_d] \quad (1.1)$$

Além da velocidade térmica média \vec{v} , a equação 1.1 apresenta a velocidade média de deriva \vec{v}_d cujo valor aproximado é de apenas 10^{-3} m/s (SRIVASATAVA, 2014), pois representa a velocidade no eixo da corrente, isto é, o deslocamento efetivo do elétron no material. Define-se esta velocidade conforme a equação

$$\vec{v}_d = \frac{e\vec{E}\tau}{m_e} \quad (1.2)$$

em que m_e corresponde a massa do elétron, e à carga elétrica do elétron e τ ao período, proposto por Drude como uma constante, entre o início e fim de uma colisão chamado de tempo de relaxamento.

Introduzindo a velocidade descrita pela equação 1.2 na equação 1.1 e realizando o somatório, temos o primeiro termo consistindo em somar as velocidades aleatórias de todos os elétrons, portanto, por definição isso é zero. Restando assim apenas o segundo termo

$$\vec{j} = \frac{ne^2\vec{E}\tau}{m_e} \quad (1.3)$$

Em que n é a densidade de portadores de carga. Conforme a definição da densidade de corrente $\vec{j} = \sigma\vec{E}$ obtemos que a condutividade elétrica (σ) pode ser descrita como

$$\sigma = \frac{ne^2\tau}{m_e} \quad (1.4)$$

Da teoria cinética dos gases (WILSON, 1954) temos a definição de livre caminho médio, isto é, a distância média percorrida de duas colisões, sendo $l = \vec{v} \cdot \tau$ e a relação entre velocidade e temperatura $\frac{1}{2}m_e\langle v^2 \rangle = \frac{3}{2}k_B T$, no qual k_B é a constante de Boltzmann e T a temperatura. Reescrevendo a equação 1.4, em termos das equações definidas neste parágrafo, obtemos:

$$\sigma = \frac{ne^2l}{(3m_e k_B T)^{\frac{1}{2}}} \quad (1.5)$$

Para validação deste resultado utiliza-se a lei de Wiedemann-Franz, definida através da relação:

$$\frac{\kappa}{\sigma} = LT \quad (1.6)$$

Em que κ corresponde a condutividade térmica e L é o número de Lorenz, isto é útil pois, experimentalmente, obtêm-se que o número de Lorenz pode ser considerado aproximadamente constante como $L = 2,45 \times 10^{-8} W\Omega/K^2$ (KUMAR *et al.*, 1993). Da teoria de Drude, temos que $\kappa = \frac{1}{3}c_v vl$, consequentemente substituindo isto e a equação 1.5 na equação 1.6 obtemos o número de Lorenz $L = 1,11 \times 10^{-8} W\Omega/K^2$, exibindo a falha do modelo, visto que o valor corresponde apenas a cerca da metade do esperado.

Lorentz, em 1905, (LORENTZ, 1905) com o intuito de resolver as falhas da teoria de Drude, desenvolveu a teoria de Drude-Lorentz utilizando a estatística de Maxwell-Boltzmann. Com isso, ele definiu a equação de Boltzmann incluindo um campo elétrico e um gradiente de temperatura. Porém, apesar de abandonar a ideia de que todos os elétrons possuem a mesma velocidade térmica média, e consequentemente ter uma descrição mais real do fenômeno, ao observarmos a condutividade elétrica deste modelo, conforme equação 1.7, temos que a única diferença com relação ao modelo de Drude, da equação 1.5, é a introdução do valor $\left(\frac{8}{3\pi}\right)^{\frac{1}{2}} \sim 0,92 \sim 92\%$, apresentando uma diferença no valor de apenas 8%, de modo que não significou uma grande evolução na compreensão da dependência da temperatura em relação a

condutividade. Porém, demonstrou que mesmo ao utilizar uma teoria mais avançada o resultado ainda se reduz a uma forma próxima a obtida por Drude.

$$\sigma_L = \left(\frac{8}{3\pi}\right)^{\frac{1}{2}} \frac{ne^2 l}{(3mk_B T)^{\frac{1}{2}}} \quad (1.7)$$

Em 1928, Sommerfeld (SOMMERFELD, 1928) apresentou sua teoria em que utiliza a estatística de Fermi-Dirac para a solução do problema do fenômeno de transporte. A grande falha desta teoria é seguir a ideia de colisão elétron-íon, entretanto, o uso da distribuição de Fermi-Dirac tornou possível compreender melhor o que são os elétrons de condução e sua relação direta com a temperatura. Na figura 1.1 temos a representação da energia referente aos estados ocupados, no inserto (a) é a situação no estado fundamental, $T = 0K$, com todos os elétrons livres ocupando os estados abaixo da energia máxima E_{max} , chamada de energia de Fermi. No inserto (b), há um acréscimo de energia $k_B T$, através da transferência de calor, assim uma pequena fração dos elétrons é excitada acima da energia máxima, passando a serem os elétrons de condução, diferente da proposta de Drude em que todos os elétrons livres eram também elétrons de condução.

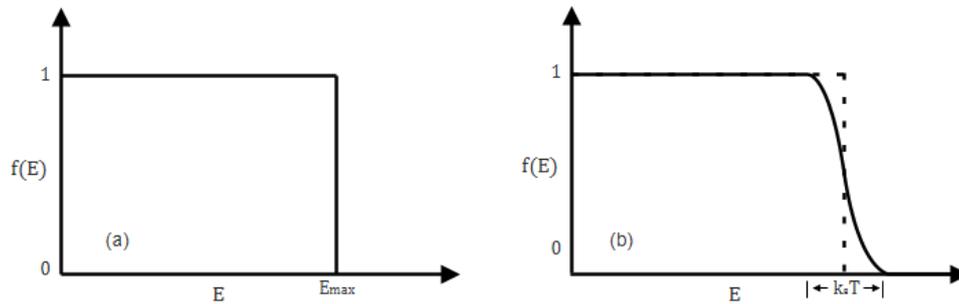


Figura 1.1: Estados ocupados na distribuição de Fermi em (a) $T = 0$ e (b) $T > 0$. Adaptado de (MEADEN, 2013).

A condutividade obtida pela teoria de Sommerfeld (σ_S) apresentada na equação 1.8 tem uma formulação similar à de Drude, com a alteração de que o tempo de relaxamento é uma função da energia de Fermi.

$$\sigma_S = \frac{ne^2 \tau(\epsilon_F)}{m} = \frac{ne^2 l}{mv_F} \quad (1.8)$$

Devido a estatística de Fermi-Dirac o livre caminho médio $l = v_F \cdot \tau(\epsilon_F)$ é função também da velocidade térmica de Fermi $v_F = \frac{\hbar}{m} (3\pi^2 n)^{\frac{1}{3}}$. Utilizando este novo modelo obtêm-se o número de Lorenz $L = 2,443 \times 10^{-8} \frac{W}{K^2}$, correspondendo a um valor mais próximo ao

esperado. Apesar disto, olhando a equação vemos que a única variável relacionada a temperatura é o livre caminho médio l , sendo assim, no regime de baixas temperaturas, para um metal, é necessário que l seja da ordem de milhares de distâncias interatômicas, para obter condutividade alta. Entretanto, sem o uso de teoria quântica era impossível a compreensão dos altos valores de l , pois com a teoria clássica não era possível explicar como o elétron se deslocaria por distâncias tão longas sem sofrer colisão com íons.

A explicação para esta dificuldade foi proposta por Bloch, em 1928, (BLOCH, 1928) que passou a considerar a movimentação dos elétrons como sendo periódica, cujo período é igual ao da rede cristalina. Nesta visão, em um material com periodicidade perfeita o elétron se deslocará através da rede sem sofrer resistência, resultando em um deslocamento livre médio com ordem de grandeza igual à do material. Essa situação é impossível, visto que o material pode ter diversas imperfeições, entretanto resolveu o problema das teorias anteriores e apresentou dois importantes fatores que influenciam o deslocamento do elétron e, conseqüentemente, a condutividade: as vibrações e imperfeições da rede. Esses fatores correspondem a propriedade chamada de resistividade elétrica, que tem como função impedir o deslocamento dos elétrons através do material, damos sua definição como o inverso da condutividade: $\rho \equiv \frac{1}{\sigma}$. Além disto, podemos ainda escrevê-la com base nas duas contribuições apresentadas anteriormente para obtermos a equação 1.9, em que ρ_0 corresponde às imperfeições da rede, independente da temperatura, e $\rho_i(T)$ às vibrações dos íons, dependente da temperatura.

$$\rho = \rho_0 + \rho_i(T) \quad (1.9)$$

Reescrevendo a equação 1.4 para a resistividade obtemos

$$\rho = \frac{m_e}{ne^2\tau} \quad (1.10)$$

e assim, podemos observar os termos que apresentam as origem das contribuições definidas na equação 1.9. Isto é, os termos m_e e e^2 são constantes, enquanto n e τ dependem de fatores como estrutura cristalina e temperatura. Na próxima seção discutiremos o impacto dessas dependências na caracterização da amostra.

1.2 – Resistividade Elétrica

O primeiro modo de observar características que compõe a resistividade é através da relação entre a densidade de corrente elétrica \vec{J} com o campo elétrico \vec{E} conforme a equação:

$$\vec{E} = \rho \vec{J} \quad (1.11)$$

Integrando os dois lados da equação acima para o caso de uma amostra na forma de um paralelepípedo e considerando que estamos tratando de um campo elétrico constante, obtemos:

$$[\rho] = \frac{V A}{I l} \quad (1.12)$$

em que V é diferença de potencial, I corrente, A seção de choque, l o comprimento da amostra e $[\rho]$ a resistividade elétrica tensorial. Desta equação, conseguimos obter a resistividade através de propriedades mensuráveis, mas ainda podemos reescreve-la através de uma expressão matemática, cujos dispositivos que a respeitam são denominados ôhmicos, dada por:

$$R = \frac{V}{I} \quad (1.13)$$

em que R é a resistência elétrica do material, que difere da resistividade devido a sua dependência geométrica.

Na equação 1.12 apresentamos a resistividade na forma de tensorial, correspondente ao caso generalizado, entretanto o caso de maior simplificação consiste na situação onde ρ é um escalar. Obtemos isto com uma corrente linear em relação ao campo, neste caso o tensor reduz à matriz diagonal com todos os termos iguais:

$$[\rho_{ij}] = \begin{pmatrix} \rho_{11} & 0 & 0 \\ 0 & \rho_{22} & 0 \\ 0 & 0 & \rho_{33} \end{pmatrix} = \rho \quad \text{com } \rho_{11} = \rho_{22} = \rho_{33} \quad (1.14)$$

A representação dos fatores que dão origem a resistividade conforme à equação 1.9 corresponde a Regra de Matthissen (MEADEN, 2013), postulada em 1860, onde definiu ρ como a resistividade total do material, ρ_0 a resistividade residual devido a presença de falhas, impurezas e tensões na rede, englobando arranjo atômico e estado químico, e $\rho_i(T)$ a resistividade intrínseca correspondente a fatores dependentes da temperatura. A forma como foi proposto um passo importante para a visualização do fenômeno pois permite observar onde se encontra a dependência da temperatura, isto é, o aumento da temperatura induz acréscimo das vibrações na rede aumentando a velocidade dos elétrons e consequentemente aumentando a probabilidade de colisões, pode também acarretar mudanças na cristalografia do material, causando transições de fase e transições magnéticas. Esses efeitos causam alterações fundamentais na resistividade, cujo comportamento depende do tipo do material a ser estudado.

Os metais em geral são bons condutores elétricos e apresentam um comportamento muito semelhante ao apresentado pela Regra de Matthissen. No inserto (a) da figura 1.2 temos a variação da resistividade em função da temperatura para um metal, em temperaturas baixas

apenas a resistividade residual é vista, conforme a regra, e o termo dependente da temperatura, dominante acima de uma certa temperatura, apresenta um comportamento linear. Essa consistência entre a regra e resultados experimentais permite a derivação da seguinte equação:

$$\rho = \rho_{297,15} [1 + \alpha(T - T_{297,15})] \quad (1.15)$$

Em que $\rho_{297,15}$ é a resistividade em temperatura ambiente, α o coeficiente de variação térmica da resistividade e $T_{297,15}$ a temperatura ambiente.

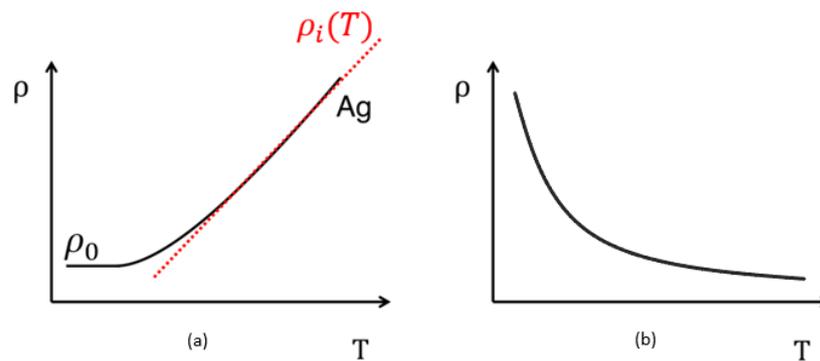


Figura 1.2: Resistividade em função da temperatura para um (a) condutor e um (b) semiconductor.

Entretanto isso não se aplica a outros tipos de sólidos, no inserto (b) da figura 1.2 temos que semicondutores apresentam um comportamento distinto ao dos condutores, com um aumento da resistividade conforme a temperatura diminui. Além disto, em temperatura ambiente a diferença entre a resistividade elétrica de um bom condutor elétrico e um semiconductor como, respectivamente, Prata, de $\rho_{Ag} = 1,59 \times 10^{-8} \Omega \cdot m$, e do Silício, de $\rho_{Si} \approx 60 \Omega \cdot m$, em temperatura ambiente (GIANCOLI, 2008) é da ordem de $10^9 \Omega \cdot m$. A causa desta grande diferença pode ser explicada utilizando a teoria de bandas (KITTEL, 2000). Na figura 1.3 apresentamos a ocupação das bandas de energia pelos elétrons, representado pelo hachurado, enquanto a área em branco representa o gap entre duas bandas. A camada preenchida e abaixo da energia de Fermi ϵ_F corresponde a camada de valência, enquanto acima desta energia temos a camada de condução.

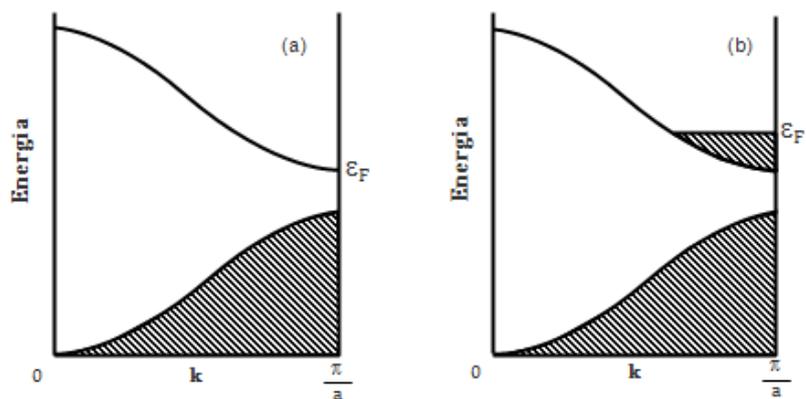


Figura 1.3: Composição das bandas de energia para um (a) semicondutor e um (b) condutor.

Para que um sólido seja um bom condutor é necessário que ele tenha elétrons na camada de condução e a forma de isto acontecer é conforme o inserto (b) da figura 1.3, em que a camada está incompleta e portanto os elétrons necessitam de pouca energia para serem excitados da camada de valência para a de condução. Um semicondutor entretanto tem a camada de valência totalmente preenchida, conforme inserto (a), em função disso existe um gap grande energia entre as duas camadas, ocasionando em poucos elétrons excitados, o que explica o incremento da resistividade em baixas temperaturas.

Esta distinção das camadas de energia dos sólidos permite diferenciar os termos dominantes da resistividade dada na equação 1.10. No caso de metais, a variação da temperatura causa mudanças na vibração dos íons da rede, portanto o fator importante da resistividade neste caso é o tempo τ . Enquanto para os semicondutores este fator não apresenta relevância pois os elétrons em sua grande maioria estão presos na camada de valência, disto o termo dominante é o número de elétron livres n .

Para a realização do estudo dessas propriedades em âmbito experimental é necessário o uso das devidas técnicas. O resultado dessas medidas é de onde serão feitas as caracterizações elétricas e validação das teorias propostas. O caso da descoberta da supercondutividade em 1908, em particular, expressa bem o interesse nessas medidas. Ao estudar o comportamento elétrico do mercúrio em temperaturas próximas ao zero absoluto, Onnes (VAN DELFT *et al.*, 2011) observou que os dados obtidos contrapunham a regra de Matthiessen, quando a temperatura do mercúrio atingiu $4,2K$ a resistividade passou por uma queda abrupta para um valor nulo, não exibindo a resistividade residual, e para esta transição deu-se o nome de fenômeno da supercondutividade.

1.3 – Técnicas de Medida de Resistividade Elétrica

Nesta seção são descritas algumas características de métodos de medida utilizados no estudo de propriedades elétricas.

1.3.1 – Método de Corrente Contínua (DC)

Este método consiste na aplicação de uma corrente elétrica contínua (I) e na medida da diferença de potencial (V) para determinação da resistência de uma amostra. Recorrendo a equação 1.13 podemos obter a resistência, portanto utilizamos a relação

$$\rho = R \frac{A}{L} \quad (1.16)$$

sabendo da geometria da amostra a seção-reta A e o comprimento L para a definição da resistividade.

Na figura 1.4 temos a configuração experimental para este método combinado com a técnica das duas pontas (GIROTTO *et al.*, 2002), em que a medida da diferença de potencial V é realizada sobre os contatos elétricos da corrente I . Este método é bastante utilizado devido a sua facilidade, pois requer apenas o conhecimento da resistência e da geometria da amostra para obtenção da resistividade, conforme a equação 1.16.

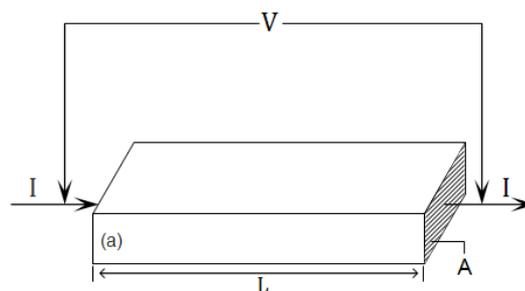


Figura 1.4: Esquematização experimental para aplicação do métodos de duas pontas.

Este método é válido para os casos em que a quantidade de portadores de carga na amostra do material analisado seja alta, ou seja, em condutores. Neste trabalho, este método será utilizado para a medida da resistência de um termômetro de Platina. A precisão desta técnica é significativa, porém caso a resistividade da amostra seja menor que a dos fios de contato haverá dificuldade na distinção do valor real. Neste caso, utiliza-se o método seção a seguir.

1.3.2 – Método de Corrente Alternada (AC)

Nos casos onde há a necessidade de medidas mais precisas devido à baixa resistividade do material utilizamos o método do espectro AC. Este método resulta na parte real do valor da impedância, conseqüentemente remove os acréscimos indesejados devido à resistência de contato elétrico e dos fios. Neste método, há uma maior complexidade quando comparado ao DC para a realização das medidas, isto surge da necessidade de um gerador de sinal que deixe em igualdade a frequência da corrente aplicada e da tensão medida. Disto define-se um zero de referência para a resistência e inicia-se o experimento. Conforme ocorre variação da temperatura, o valor do gerador de sinal deixará de ser zero, sabemos assim que a alteração na resistência, portanto como não há fase entre I e V pode-se, portanto, utilizar novamente a equação 1.16.

Para aplicação experimental do método AC utiliza-se o método das quatro pontas (SCHRODER, 2006) conforme a figura 1.5. Devido à alta impedância do multímetro ocorre a redução de corrente fluindo entre os dois pontos de contato V , tornando negligenciável a influência da resistência de contato introduzida pelos contatos elétricos e dos fios.

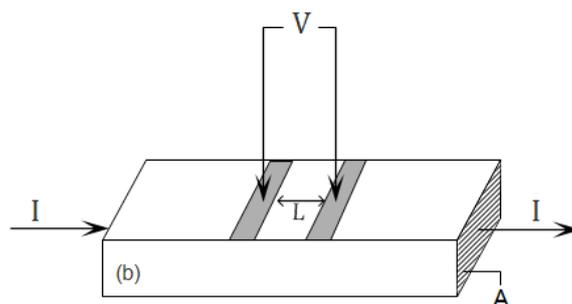


Figura 1.5: Esquematização experimental para aplicação do método das quatro pontas.

1.3.3 – Método das Curvas Características I-V (Corrente-Voltagem)

Outra técnica que utiliza o método das quatro pontas é para se obter as curvas características I-V (Corrente-Voltagem) (SCHRODER, 2006), e neste caso o interesse é compreender como ocorre o transporte de cargas pelo material. Diferente dos anteriores onde a tensão era uma propriedade a ser medida, quem toma esta posição neste é a corrente enquanto a tensão passa a ser aplicada com vários valores. Na figura 1.6, temos, no inserto (a), a curva I-V de comportamento linear para uma amostra metálica, da equação 1.12 obtem-se a inclinação e conseqüentemente com a variação do ângulo pode-se deduzir a grandeza da

resistência do dispositivo. No inserto (b) têm-se um comportamento bastante distinto, para o caso de uma amostra semicondutora, desta forma o método permite a identificação do tipo da amostra.

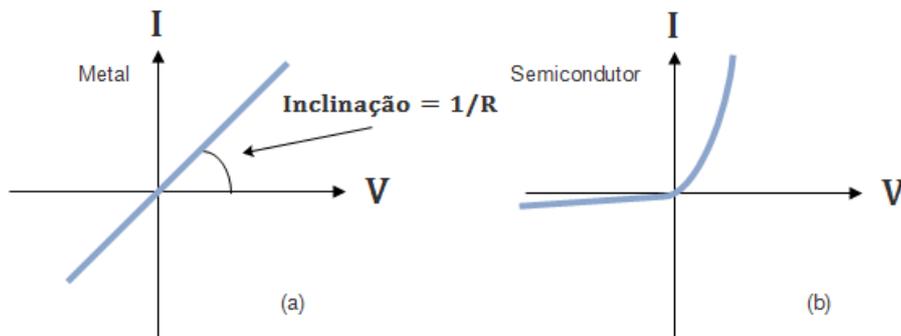


Figura 1.6: Curvas I-V características para duas um (a) metal e um (b) diodo.

Neste trabalho, para realizarmos as medidas de temperatura, será utilizado o método das duas pontas de corrente contínua em um termômetro, e para a medida da resistividade da amostra, o método de quatro pontas de corrente alternada. O uso de um computador com um software para a aquisição de dados minimiza a contribuição do erro associado a influência humana, visto que medidas de caracterização elétrica são realizadas por longos períodos de tempo. O *software* deve ser capaz de realizar as medidas de tensão da amostra e da temperatura com baixa latência; e ao fim da medida, deve reproduzir um arquivo de dados, no formato *.dat*, contendo os valores da temperatura, resistência e resistividade.

2 – Procedimentos Experimentais e Computacionais

Neste capítulo iremos apresentar os componentes de uma montagem para medidas de transporte elétrico, que tipo de controle podemos introduzir neste sistema e o desenvolvimento do *software* para automação deste procedimento.

2.1 – Sistema Térmico

Nesta seção apresentamos o funcionamento de criostatos e como o utilizamos para o experimento e para a aplicação de vácuo, iniciando o processo de obtenção de um ambiente de temperatura controlada.

2.1.1 – Criostato

Criostatos (JIRMANUS, 1990), também chamados de Dewars, são equipamentos confeccionados como um recipiente no qual é introduzido um material criogênico, usualmente sendo nitrogênio líquido ou hélio líquido devido a suas baixas temperaturas. Este material servirá para reduzir a temperatura do sistema em seu interior e por conseguinte reduzir a transferência de energia em forma de calor para o ambiente externo através uma série de barreiras. Essa transferência pode ocorrer de três modos diferentes: condução, convecção e radiação, sendo necessário que algumas precauções sejam tomadas de modo que todas sejam minimizadas. As duas primeiras, dadas por contato, tem grande influência pois o ambiente interno terá um gradiente de temperatura com relação ao externo, que causará uma rápida transferência para que se obtenha equilíbrio térmico. Para evitar isto, fazemos isolamento térmico com vácuo utilizando uma bomba de vácuo que dificultará a interação entre os dois ambientes. A terceira transferência, radiação, é impedida com a inserção de um material refletor, como papel alumínio, entre as paredes mais externas para que as ondas eletromagnéticas, que se propagam no vácuo, não façam contato com o sistema. Na ausência desta proteção é recomendado, ao menos, que o sistema interno seja revestido com o material, pois o efeito da radiação no aquecimento é bastante influente.

No inserto (a) da figura 2.1 temos a representação do criostato que utilizamos. Primeiramente temos a região 3, composta por dois vidros separados por alguns centímetros, esta mantém vácuo dificultando a interação entre o ar quente fora do criostato e o nitrogênio líquido. Supondo a ausência desta região seríamos incapazes de manter o nitrogênio em estado líquido por longos períodos de tempo e que, portanto, prejudicaria imensamente nosso experimento.

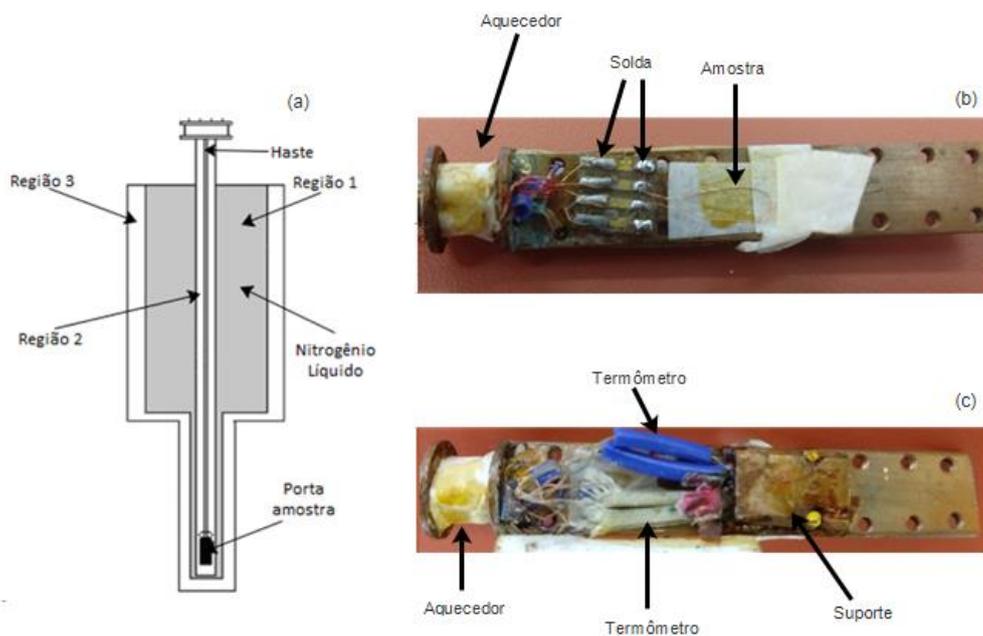


Figura 2.1: Diagrama do (a) criostato, (b) parte superior do porta amostras onde é fixada a amostra e (c) parte inferior do porta amostras em que dois termômetros distintos estão acoplados à um suporte de cobre para melhorar o contato térmico.

A região 1 é o reservatório do criostato, em que depositamos o nitrogênio líquido, separado por vácuo do ambiente externo e da haste de medidas.

A região 2 consiste de um vidro de quartzo onde faz-se vácuo e é onde inserimos a haste de medidas. A haste tem em sua extremidade inferior o porta amostra, composto por um aquecedor e uma chapa feita de cobre para obtermos todo o aparato em uma mesma temperatura. Por dentro da pequena tubulação desta haste são passados fios de cobre de AWG (escala americana de bitolas de fios) 36, pois são bons condutores de corrente e finos o suficiente para que não conduzam muita energia térmica a ponto de interferir nas medidas. Estes fio são soldados no cabeçote da cana para serem conectados aos instrumentos e na extremidade inferior são soldados nos componentes do porta amostra, sendo que todos os fios são trançados para evitar interferência magnética resultante da corrente conduzida.

Os componentes acoplados são posicionados de forma específica para garantir melhores resultados. No inserto (b) da figura 2.1 temos uma pequena peça também de cobre na qual são feitos contatos elétricos, neste lado são soldados os fios e do outro lado da peça faz-se a solda com os contatos da amostra, conforme a técnica das quatro pontas, para garantir que ambos permaneçam fixos utiliza-se graxa de vácuo no contato com a chapa. No inserto (c), o outro lado do porta amostra, é onde introduzimos termômetros para obter a temperatura da amostra.

Por uma questão de melhoria no contato com a chapa e estabilização, parafusamos um bloco de cobre com duas perfurações na chapa, com posição na parte oposta a amostra, onde os inserimos. Optamos pelo uso de dois termômetros em contrapartida ao uso de apenas um pela necessidade de melhor controle e precisão da temperatura.

O aquecedor é um resistor composto de um fio de níquel-cromo (NiCr), cuja resistência é de 27Ω , envolvido de forma a cobrir a maior área possível, com a função de controlar a temperatura da chapa de cobre e das componentes. Para garantir um bom contato térmico, o fio enrolado é colocado na superfície e assim passa-se uma camada de verniz, em seguida uma camada de papel de seda e novamente verniz, isto garante que o aquecedor permaneça em contato constante com a haste. O uso de NiCr atribui-se por ser resistente e flexível o suficiente para não quebrar facilmente ao ser enrolado, além de que sua resistência é bem conhecida.

2.1.2 – Procedimento de Vácuo

Para a conservação do material criogênico por um longo período de tempo é necessária a utilização de um equipamento confiável para realizar o vácuo no criostato. Para isto utilizamos o RV5 da Edwards (EDWARDS, 2013), capaz de prover vácuo de grau alto, isto é, $2 \cdot 10^{-3}$ mbar. Conforme aplicado na camada mais externa do criostato, conseguimos garantir que haja uma quantidade pequena de partículas presentes, dificultado a troca de calor e facilitando a manutenção da temperatura interna como de interesse. Essencialmente, o isolamento do sistema externo é o principal fator de uso do vácuo, mas conforme o tipo de criostato isso pode variar com mais ambientes propensos a conservar o material. No nosso caso, há um total de 3 regiões, conforme o inserto (a) da figura 2.1, em que fazemos vácuo, assim garantimos que a parte interna, onde se encontra a amostra, sofrerá em escala menor a interação com outro sistema de temperatura maior, contrário ao caso da região externa.

Para a instalação da bomba de vácuo, antes de sua utilização, seja no criostato ou em outra montagem, são necessários alguns procedimentos de segurança para garantir o funcionamento ideal. O procedimento para o primeiro uso de uma bomba de vácuo dá-se primeiramente abrindo-a, separando a região do motor e a do óleo. E manualmente aplicando 100ml de óleo na flange de entrada (NW25) e girando em sentido contrário o elemento de acoplamento 411 (EDWARDS, 2013). O óleo será sugado para dentro do equipamento, lubrificando-o e assim evitando problemas no primeiro uso.

Um procedimento necessário para todas medidas consiste no bombeamento do gás do criostato utilizando a bomba de vácuo antes de iniciar o experimento, primeiramente deve-se

abrir as válvulas 2 e 3, conforme figura 2.2, correspondentes as extremidades do criostato, verificando-se que atingiu valor próximo a $2 \cdot 10^{-3}$ mbar. Provavelmente, estará quase estável, é recomendável só então introduzir o nitrogênio líquido (*NL2*) pois assim se evita a perda de material criogênico durante o esfriamento. Após atingir a temperatura de aproximadamente 77K, se for de interesse, pode-se fechar as válvulas 2 e 3, em seguida abrir a 1, com isto será aumentada a pressão exatamente na área central do criostato, onde está depositado o *NL2*. Isso leva à queda da temperatura, para valor mínimo próximo a 57K, porém advém no gasto mais rápido do *NL2*. Ainda, ao retomar o aquecimento, após ser resfriado a temperaturas abaixo de 77K, deve-se fazer o procedimento inverso ao atingir temperaturas próximas a 77K para continuar o procedimento normalmente.



Figura 2.2: Válvulas do criostato.

2.2 – Eletrônica Associada

Nessa seção apresentamos a eletrônica da montagem, composta de duas configurações conforme inserto (a) e (b) da figura 2.3, e as ações de controle no aquecedor que permitem o controle da variação da temperatura. Disto apresentamos os métodos, manuais e automáticos, de como se ajusta a ação controladora PID.

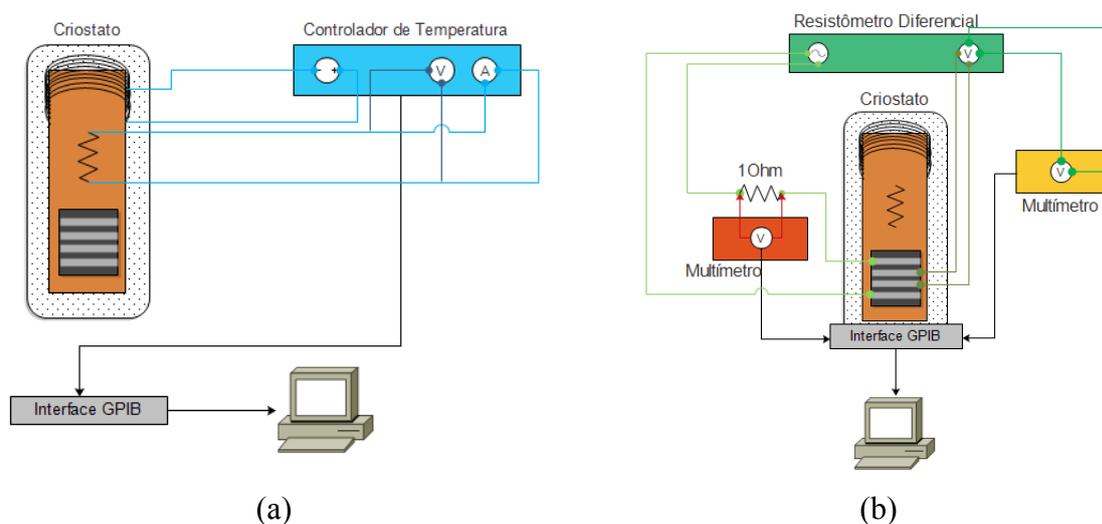


Figura 2.3: Esquema das duas montagem eletrônicas utilizadas no experimento deste trabalho.

2.2.1 – Resistômetro Diferencial

Para a realização da medida utilizando o método da quatro pontas é necessária a aplicação de uma corrente, em que usualmente se utiliza uma fonte confiável, e medir a diferença de potencial, normalmente com o uso de um multímetro. Entretanto, em nosso laboratório temos a nossa disposição o resistômetro diferencial da figura 2.4, confeccionado na UFRGS e doado pela mesma à FURG, portanto o utilizaremos para tais medida com um multímetro lendo seus valores de saída.



Figura 2.4: Vista frontal do resistômetro diferencial.

Este equipamento permite que seja feita a escolha da corrente aplicada com opções entre 0,1 e 100mA, tal valor deve ser escolhido baseado no tipo de amostra que está sendo medida, pois se for um valor elevado pode danificar a amostra. Montamos um pequeno circuito no qual a corrente, após percorrer a amostra e sair do criostato, deve passar por uma resistência padrão de 1 Ω e em seguida um multímetro mede a diferença de potencial, após isto a corrente segue normalmente retornando ao resistômetro. Isto foi montado para que possamos calcular a corrente com maior precisão de sua medida.

A medida da tensão da amostra é feita através de um multímetro ligado aos conectores presentes na parte frontal do resistômetro, conforme figura 2.4. Como pode ocorrer de o valor da resistência ser muito pequeno, o multímetro em alguns casos não conseguirá registrar tais valores, pois está fora de seu alcance. Para tanto, o resistômetro disponibiliza algumas ferramentas para solução deste problema, primeiramente o que se faz é definir um zero da diferença entre a resistência da amostra e de referência do instrumento, com isso o que será medido será a diferença além do zero. Novamente, isto provavelmente estará muito pequeno, portanto, pode-se definir um ganho de saída para ampliar o valor e permitir a leitura pelo multímetro. Caso seja utilizado qualquer tipo de ganho é importante registrar isto na medida,

pois eventualmente quando os dados forem obtidos deve-se normalizá-los e tirar qualquer ganho, dado que são necessários apenas para registro dos instrumentos.

2.2.2 – Controlador de Temperatura

O controlador de temperatura é o equipamento que permite medir a temperatura do sistema ou introduz corrente no aquecedor para aquecê-lo ou esfria-lo com determinado controle. Utilizamos o modelo 335 da Lakeshore[®] (LAKESHORE, 2012), conforme a figura 2.5, que com a configuração apropriada é capaz de operar em temperaturas de até 300mK com resolução de até 0,5mK, para o caso de um termômetro de Platina.

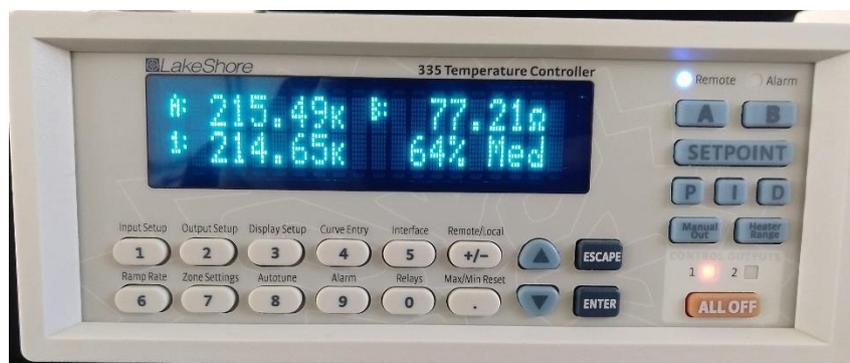


Figura 2.5: Vista frontal do controlador de temperatura Lakeshore[®] 225.

O instrumento é incapaz de medir temperatura por si só, para tanto é utilizado um termômetro, ou termoresistor (BALSHAW, 1996). Este sensor permite obter a temperatura a partir da medida de sua resistência, entretanto, essa relação pode ou não ser fornecida pelo fabricante. Caso se opte por comprar um sensor cuja calibração já foi realizada, o termômetro terá um custo para compra muito superior ao sem calibração, pois este não é um processo simples de ser feito. Para ser obtida essa calibração é necessária uma comparação de um sensor cujo comportamento seja bem conhecido com o de interesse, portanto é fundamental uma montagem experimental bastante confiável, mão de obra, uso de material criogênico e acesso a um sensor calibrado. O material com o qual o sensor é feito também introduz diferenças significativas, como a região de temperatura na qual tem melhor resolução e se suporta campos magnéticos, além de grande variação em custo para aquisição. Ainda, para medidas que cobrem uma grande região, como por exemplo de 1 a 300K, são necessários o uso de mais de um sensor, cada um com alta resolução para cada região em específico.

Em nosso experimento utilizamos dois termômetro de Platina, ambos sendo PT-103 (RTD – Resistor Temperature Detector), sendo um calibrado de fábrica e o outro não. As

informações que temos do calibrado são um polinômio de Tchebychev com seus determinados coeficientes, em três faixas de temperatura, fitando seu comportamento: 14,0K a 20,1K; 20,0K a 100,0K; 100,0K a 325K, sendo está última a de interesse por compreender a região do nitrogênio líquido. Além do polinômio temos também duas curvas de interpolação, em que a menor destas é chamada curva de resposta da temperatura, cujos valores introduzimos no controlador de temperatura para, ao realizarmos as medidas, que ele mostre o valor da temperatura com a precisão da calibração, caso contrário ele apresentaria o valor utilizando uma curva padrão dele ou apenas a resistência do sensor. Tendo em mãos este sensor já calibrado, podemos realizar o procedimento de calibração, isso garante a disponibilidade de um sensor com ótima precisão, porém valor comercial muito mais baixo. Além disto, possíveis danos a um deles não nos impossibilita de continuar com o experimento e ainda nos permite a consequente calibração de outros termômetros.

A função do controle do equipamento funciona através do fornecimento de corrente ao aquecedor (fio de NiCr com 27Ω). Sua utilização pode ser para se obter uma variação da temperatura A até temperatura B com a uma taxa constante.

Isto permite o estudo de uma propriedade do material em função da temperatura, como é o caso da resistividade, em que a variação pode levar a uma transição de fase. Para tal, é feito o uso de uma rampa até um certo *set-point*, na qual o controlador permanece aquecendo o sistema com o intuito de permanecer constante em um determinada taxa de variação de temperatura, que pode ser definida de 0,1 a 100K/min. Outro uso de controle permite a estabilização de uma temperatura fixa para o monitoramento de uma outra propriedade de controle. Um exemplo no qual isto pode er utilizado é o do estudo do efeito Hall (ARAÚJO, 2013) de um material, no qual consiste na aplicação de um campo magnético, variando a intensidade em uma temperatura fixa, outros usos incluem estudo de magnetorresistência e relaxação. E, neste caso, é definido apenas o *set-point* a ser atingido, portanto o instrumento deverá levar o sistema a tal temperatura com a mais rápida velocidade possível, sem preocupação com a taxa, ao atingi-la mantemos o sistema nesta temperatura.

2.2.3 – Ações de Controle do Aquecedor

Independentemente do método pelo qual se tem interesse em utilizar o instrumento para o controle, não basta apenas ele fornecer corrente para o aquecedor se não houver um controle preciso sobre tal variável. Para tanto, o que se utiliza são operações sobre a variável de controle, que são as chamadas ações de controle (KING, 2016), no qual um sinal produzido pelo

equipamento sobre um valor de referência deve levar o sistema ao valor desejado. Os sistemas que utilizam essas ações são de malha fechada, o que significa que tais ações são baseadas em um *feedback* do sistema, como podemos observar na figura 2.6, da qual retiramos importantes nomenclaturas:

- **Erro ou $e(t)$** : sinal de erro, é a diferença entre o valor desejado e o valor medido;
- **U ou $u(t)$** : sinal de saída, gerado para diminuir o erro;
- **SP**: *set-point*, o valor desejado;
- **$G_c(s)$ e $G_p(s)$** : função transferência, respectivamente, do controlador e do processo;
- **$Y(t)$** : valor de saída do processo.

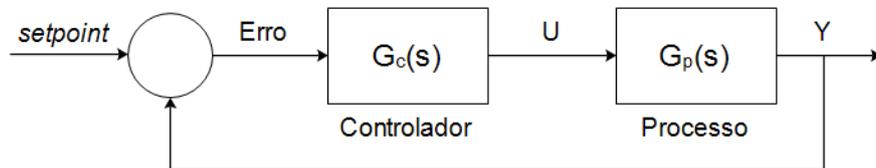


Figura 2.6: Diagrama de blocos para o controle de um processo.

Assim, a decisão da ação do controle do sistema decorre de uma análise do erro, de tal forma que o sinal aplicado pelo controlador gere um sinal que leve o processo ao estado desejado, de erro zero. Para um exemplo em controle de temperatura, o que teríamos é caso de um sistema que possui uma diferença – erro – entre a temperatura atual e a desejada – *set-point* –, então através de uma *função* ele irá modificar a corrente – $u(t)$ – aplicada ao sistema com o intuito de que o valor da temperatura – $Y(t)$ – esteja mais próxima, ou corresponda, ao valor desejado.

Podemos separar em 4 tipos de ações que o controlador pode realizar sobre o processo, isto é, o tipo de função que é aplicada com o intuito de modificar a variável de controle, elas são:

- Ação de Duas Posições (liga-desliga),
- Ação Proporcional (P),
- Ação Integral (I),
- Ação Derivativa (D).

Veremos nas próximas seções que é possível ainda utilizar mais de uma ação em um sistema, geralmente para auxiliar em pontos onde uma ação não atua.

2.2.3.1 – Controle liga-desliga

A primeira ação citada, liga-desliga (KING, 2016), pode ser representada através do seguinte sistema de duas equações:

$$U = \begin{cases} M_1 \text{ (ligado)} & E > 0 \\ M_2 \text{ (desligado)} & E < 0 \end{cases} \quad (2.1)$$

Esse tipo de ação costuma ser aplicado em sistemas que não precisem de alta precisão, pois haverá uma oscilação contínua em relação ao *set-point*, pela falta de controle de intensidade da variável de controle, cujos valores serão apenas 0%(ligado) e 100%(desligado) assim gerando uma variação de ação instantânea brusca. Entretanto, esse é também o motivo de seu baixo custo e comum aplicação, visto que não há a necessidade de instalação de um sistema de controle complexo.

Na figura 2.7 temos duas representações gráficas para esse tipo de controle. Na parte superior da figura apresentamos a relação da variável controlada, que representa valores reais, e na inferior a variável manipulada, que conforme a equação acima só tem dois valores: ligado e desligado.

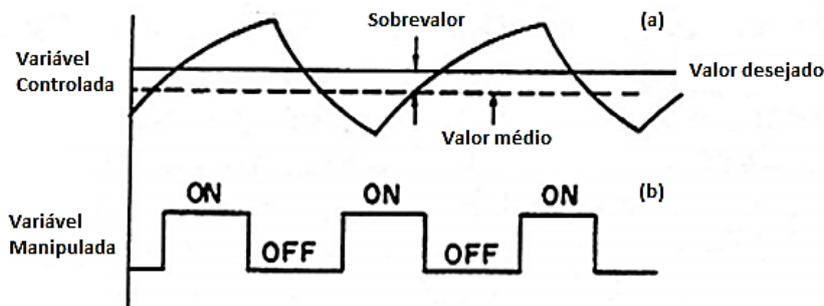


Figura 2.7: Resposta da (a) variável controlada conforme variação da (b) variável manipulada em relação ao tempo.

Tomando como exemplo um caso onde se quer obter o valor desejado conforme o da figura acima sendo uma determinada temperatura, ao ligar o sistema ele passará a aumentar a temperatura, ao ultrapassar o valor desejado ele se desliga, porém ele não sofre instantaneamente uma variação passando a ter decréscimo de temperatura pois há uma inércia do sistema. Por isso, é natural que haja um sobrevalor (*off-set*) do valor médio obtido por esse método em relação ao valor desejado durante todo o tempo em que o sistema se encontra tentando atingir o *set-point*. Por usualmente ser composto por um relé (componente elétrico), podemos notar que esse tipo de ação pode requisitar de bastante manutenção, especialmente para casos onde se necessita de estabilidade da variável controlada por período longos de tempo, pois o componente estará em constante liga-desliga.

2.2.3.2 – Controle proporcional

A segunda ação, proporcional (KING, 2016), é a única que, além da anterior, pode ser aplicada sozinha. Define-se que o sinal de saída deve ser proporcional a amplitude do sinal erro conforme a equação (ASTRÖM ANDT *et al.*, 1995):

$$u(t) = K_p e(t) \quad (2.2)$$

em que K_p é o ganho proporcional. O uso desta ação implica que o problema de ação brusca da anterior não retorna a ocorrer, havendo opção de uma ação mais suave. Porém, o obstáculo que é levar o sistema ao exato ponto desejado e manter o sistema estável ainda não é suprimido. Tomando uma análise da equação acima, vemos que ao atingir o *set-point*, situação na qual o erro é zero, o controlador deve enviar um sinal de controle à variável como também zero. Como isto representa nenhuma ação externa aplicada ao sistema, é esperado então que aja apenas ação da inércia, o que nos leva diretamente ao que observamos da seção anterior, que é sair da estabilidade e retornando o sistema para um estado de erro.

Na figura 2.8 temos três respostas de um sistema para três ganhos diferentes. Com o menor valor, temos que o ganho não é o suficiente para levar o sistema até o ponto desejado, mas é capaz de estabilizar o sistema com facilidade. Conforme o ganho é elevado, a ação passa a forçar o sistema a ter uma resposta mais intensa, levando o sistema além do ponto desejado. Nesse período ocorrem oscilações com pico alto e, devido ao *offset*, o erro inverte de sinal implicando também na inversão do sinal de resposta, isso ocorre repetidamente porém sempre com menor intensidade conforme o tempo passa até atingir um ponto de estabilidade, acima ou abaixo do *set-point*. Isso nos leva ao que é definido usualmente para resposta de um sistema que apresenta uma resposta lenta: elevar o valor do ganho proporcional aumenta a reação do sistema.

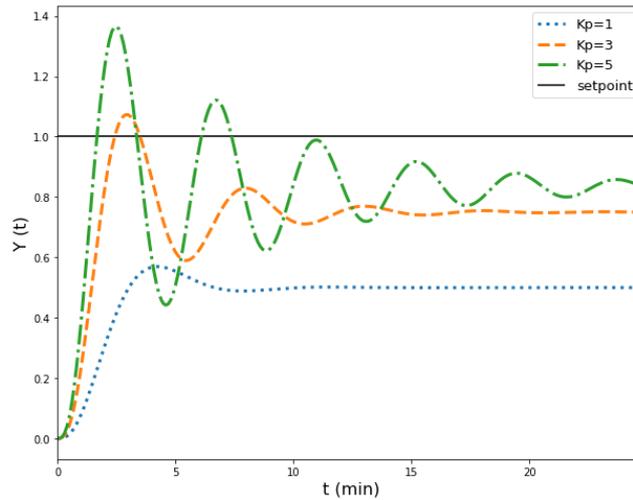


Figura 2.8: Resposta da variável $y(t)$ de um processo para três diferentes ganhos proporcionais.

2.2.3.3 – Controle proporcional-integral

O terceiro tipo de controle, proporcional-integral (PI) (KING, 2016), composto por duas ações é dado pela seguinte equação (ASTRÖM ANDT *et al.*, 1995):

$$u(t) = K_P(e(t) + K_I \int_0^t e(t) dt) \quad (2.3)$$

em que K_I é o ganho integral. Além da ação anterior, agora há outra referente a uma integral do erro, que permanece enquanto houver erro. Diferente da proporcional, que há uma atuação instantânea, essa tem uma ação prolongada com o tempo, agindo de forma a suavizar a resposta do sistema lentamente até o erro se tornar zero, que é possível nesse tipo.

Para garantir que o sistema não leve uma grande quantidade de tempo para se estabilizar é necessário o devido cuidado, pois com um grande valor de ganho integral, o fator referente a tal ganho passa a controlar o sinal, como vemos no caso da figura 2.9. Isso pode ocasionalmente implicar em uma desestabilização total do sistema. Conforme maior o ganho integral menos susceptível a oscilações é o sistema, que continua em direção ao ponto desejado porém, como dito antes, lentamente.

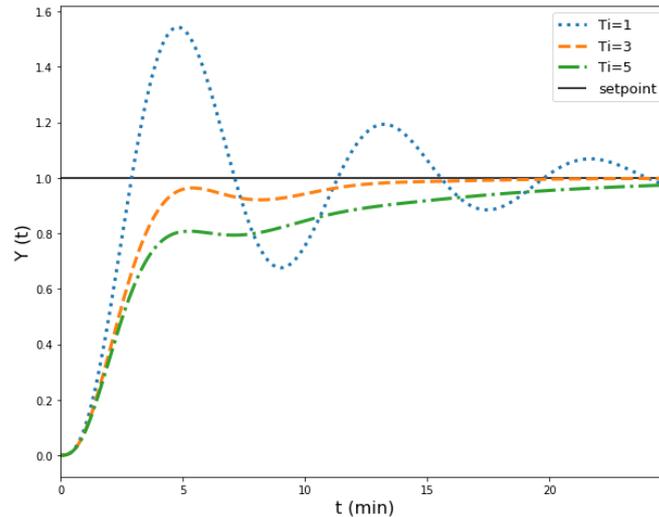


Figura 2.9: Resposta da variável $y(t)$ de um processo para três diferentes tempos integrais e um ganho proporcional igual.

2.2.3.4 – Controle proporcional-derivativo

O quarto controle, proporcional-derivativo (PD) (KING, 2016), também composto por duas ações, gera um sinal além de proporcional ao erro mas também com relação a sua variação, dado pela equação (ASTRÖM ANDT *et al.*, 1995):

$$u(t) = K_P(e(t) + K_D \frac{d}{dt} e(t)) \quad (2.4)$$

em que K_D é o ganho derivativo. Sua aplicação surgiu da necessidade de evitar a manifestação de um desvio, se há variação do erro isso deve ser compensado para o sistema.

Por ser relativa a variação do erro, a ação derivativa é por vezes denominada como sendo preditiva, verificando se o sistema está variando muito rapidamente e gerando um amortecimento. Essencial para casos em que possam surgir pequenas flutuações na saída, mesmo após a estabilidade, pois garante uma menor oscilação.

Na figura 2.10 temos duas situações, com a devida resposta da ação derivativa. No caso do inserto a temos um desvio em degrau, ocasionando uma variação do erro muito rápida e com isso a ação introduz uma reação brusca no sistema, pois é forçada a gerar um sinal forte como resultado da derivada. Entretanto, quando a variação é de forma mais lenta, em especial para uma rampa, inserto b, uma resposta linear da ação é dada garantindo um controle até atingir o ponto desejado, o que explica o fato de diversos controladores de temperaturas utilizarem apenas PD quando se deseja obter uma rampa de variação.

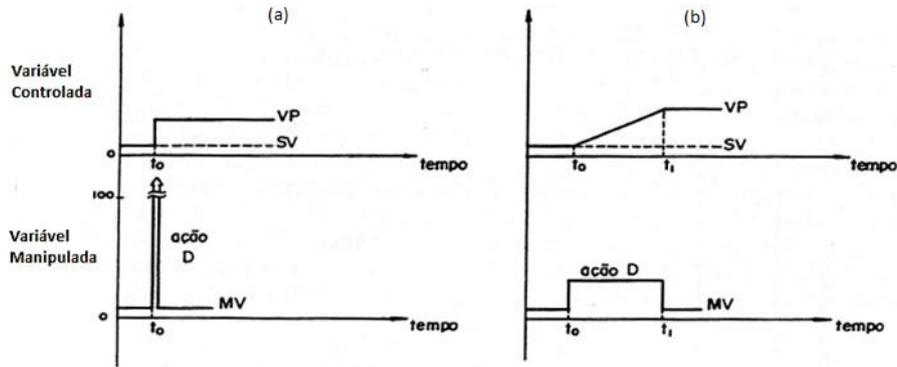


Figura 2.10: Comportamento da variável controlada para ação manipulada em forma de (a) degrau e (b) rampa.

2.2.4 – Controlador PID

Os controladores PI e PD possuem suas próprias características e benefícios, mas além disso é possível combiná-los para se obter o controlador proporcional-integral-derivativo (PID) (KING, 2016), de recorrente uso para diversos tipos de sistemas, especialmente nos mais complexos. Através da combinação das três ações, define-se a sua equação como sendo (LAKESHORE, 2012):

$$\begin{aligned}
 u(t) &= K_p \left(e(t) + K_I \int_0^t e(t) dt + K_D \frac{d}{dt} e(t) \right) \\
 &= K_p \left(e(t) + \frac{1000}{T_I} \int e(t) dt + \frac{1000}{T_D} \frac{d}{dt} e(t) \right)
 \end{aligned}
 \tag{2.5}$$

Em que introduzimos duas novas variáveis, T_I e T_D , que representam os respectivos ganhos, porém com relação ao tempo em segundos, assim fica explícita a conversão realizada no controlador de temperatura Lakeshore® 335 (LS-335). O uso do controlador PID garante que os erros provenientes de cada ação sejam compensados pela das outras duas, garantindo o melhor controle possível, evitando oscilações, altas amplitudes e possíveis desvios. Na figura 2.11 mostramos três situações para um mesmo sistema, a princípio há a situação em que existe apenas controle proporcional com alto ganho, gerando oscilações de grande amplitude. Em seguida é inserido a ação integral, se tornando o controle PI, passando assim a convergir ao ponto desejado. Na terceira etapa, a inclusão da ação derivativa nos leva ao controle PID, reduzindo os desvios, garantindo que o sistema atinja o ponto desejado com maior facilidade e com apenas um pequeno sobrevalor logo no início, porém antes da primeira oscilação observada nos outros dois sistemas.

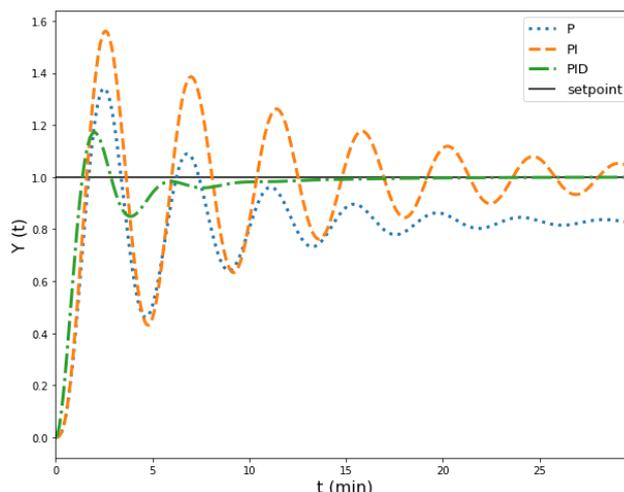


Figura 2.11: Comparativo entre resposta do sistema para ações de controle P, PI e PID para um mesmo sistema em função do tempo.

2.2.5 – Métodos de Sintonia

Apenas o equipamento não garante um controle da temperatura na taxa desejada, pois costuma vir com um PID determinado de fábrica, como é necessário que seja definido o PID para cada sistema, não podemos confiar no valor pré-definido. Nesta seção, portanto, apresentaremos com o uso de quais métodos podemos realizar o ajuste deste valor.

2.2.5.1 – Método de Auto Sintonia

O primeiro método costuma ser o mais utilizado nos casos no qual o próprio equipamento o fornece, como é o caso do nosso LS-335. Consequentemente, é interessante o utilizarmos para análise, visto que além da facilidade de seu uso, como veremos, há também a questão de ser amplamente utilizado, portanto, sua implementação já passou por diversos testes reais e otimizações por parte do fabricante.

Apesar de ser um método automático, ainda é necessário a interação manual para algumas definições, sendo a primeira e mais importante a de definição da potência do aquecedor. O equipamento não faz este processo automaticamente, pois tem de se tomar muito cuidado neste procedimento, visto que uma escolha ruim de valores pode causar danos ao aquecedor do sistema. Para a escolha mais viável, deve-se definir a saída manual para 50% no controle de malha aberta com o menor valor da corrente disponível na configuração do equipamento, em seguida escolhe-se a menor faixa de aquecimento, sendo que o LS-335 possui três faixas: LOW, MEDIUM e HIGH, cujos valores são:

$$HIGH = Potência Máxima = I^2 R \quad MEDIUM = \frac{HIGH}{10} \quad LOW = \frac{HIGH}{100}$$

Após esta escolha observa-se se isso foi o suficiente para atingir a temperatura desejada, em nosso caso 300K. Caso positivo temos os valores da corrente para o experimento, caso contrário repetimos o aquecimento para a próxima faixa. Alteramos o valor da saída da corrente apenas após as três faixas não satisfazerem nosso objetivo.

Além disso é importante já ter sido inserida a curva do termômetro no controlador de temperatura, pois apenas assim vai ser possível realizar o passo anterior e todos os outros. Essa curva de modo geral costuma ser dada pelo fabricante definindo diversos pontos e garantindo maior precisão nas medidas. Porém, é possível também realizar a utilização do método de *SoftCal* para a geração da curva, neste caso considerando que é utilizado um sensor de platina são introduzidos três valores nas seguintes faixas, conforme a figura 2.12:

- Primeiro em temperaturas próximas a da liquefação do nitrogênio, 77,35K;
- Segundo com próximas a temperatura ambiente, 305K e
- Terceira próximas a temperaturas altas, 480K.

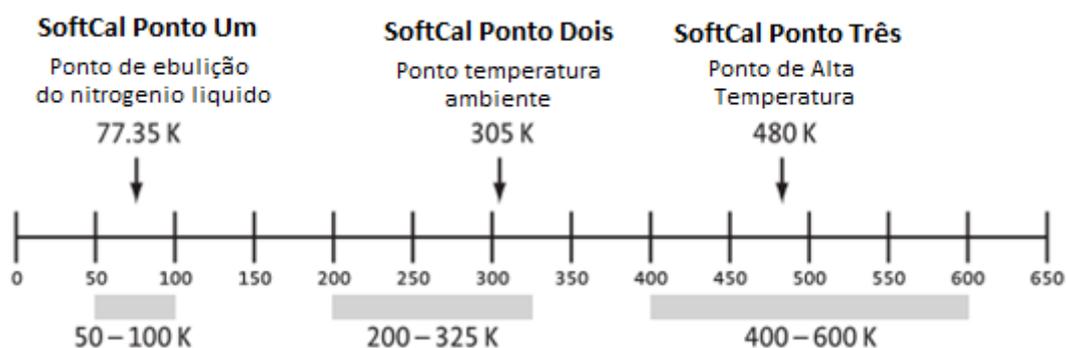


Figura 2.12: Regiões de temperatura aceita por sensores de platina. Adaptado de (LAKESHORE, 2012).

Apesar de se tratar de uma aproximação o método apresenta uma boa precisão, sendo de aproximadamente 250mK para o intervalo de 70K até 325K e de 500mK para 325K até 480K. No caso do presente trabalho, o fabricante nos forneceu uma tabela com 65 valores, com uma precisão de ± 10 mK. Sendo assim, é mais interessante a utilização destes visto que correspondem perfeitamente ao nosso termômetro, entretanto para o não calibrado é um método suficientemente bom para situações com pouco tempo para um processo lento de calibragem.

2.2.5.2 – Método de Ziegler-Nichols (ZN)

Este é um dos métodos de malha fechada mais populares para sintonia de PID devido a sua praticidade, foi desenvolvido por J. G. Ziegler e N. B. Nichols em 1942 (ZIEGLER *et al.*, 1942). Outro fator interessante para sua relevância ao longo do tempo é de que surgiram diversas modificações (HANG *et al.*, 1991) para melhorar sua precisão.

Para seu uso é necessário primeiramente zerar os parâmetros K_I e K_D . Em seguida se inicia uma série de tentativas com diversos valores de K_P para que se obtenha um comportamento oscilatório de amplitude constante do sistema, na figura 2.13 vemos que obteríamos isto para $K_P = 2,0$. Chamamos este valor encontrado de ganho crítico, K_U , e seu período de P_U , com estes dois valores podemos utilizar a tabela 2.1 para obter os respectivos ganhos para o controlador.

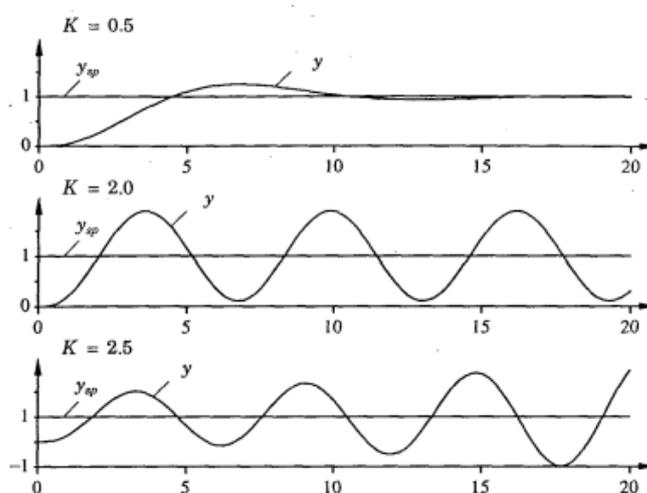


Figura 2.13: Método de tentativa e erro com três diferentes ganhos para procurar o ganho proporcional que produza oscilação de amplitude constante. Adaptado de (ASTRÖM ANDT, 1995).

Uma modificação sobre o método ZN foi desenvolvido pela Lakeshore® (LAKESHORE, 2012), com foco em experimentos com variação de temperatura. O procedimento permanece igual ao ZN, porém o termo referente ao período é medido em segundos e expresso como T_I . Para o cálculo dos ganhos segue-se conforme a tabela 2.1, sendo que o ganho diferencial deve ser definido manualmente, através de teste e erro.

Tabela 2.1: Valores para métodos de ajuste do controlador PID.

MÉTODO	K_p	K_I	K_D
ZIEGLER-NICHOLS (ZN)	$\frac{K_U}{1,7}$	$\frac{P_U}{2}$	$\frac{P_U}{8}$

ZN MODIFICADO	$\frac{K_U}{2}$	$\frac{1000}{T_I}$	Manual
COHEN-COON	$\frac{\tau}{K_p d} \left(\frac{4}{3} + \frac{d}{4\tau} \right)$	$d \frac{30 + \frac{3d}{\tau}}{13 + \frac{8d}{\tau}}$	$\frac{4d}{11 + \frac{2d}{\tau}}$
C-H-R COM SETPOINT TRACKING	$\frac{0,6 \tau}{K_p d}$	τ	$0,5d$

2.2.5.2 – Método de Cohen-Coon e método C-H-R

Diferente do método ZN, estes dois utilizam malha aberta para o ajuste da curva de reação do sistema com a aplicação de uma função degrau na variável manipulada, conforme a figura 2.14.

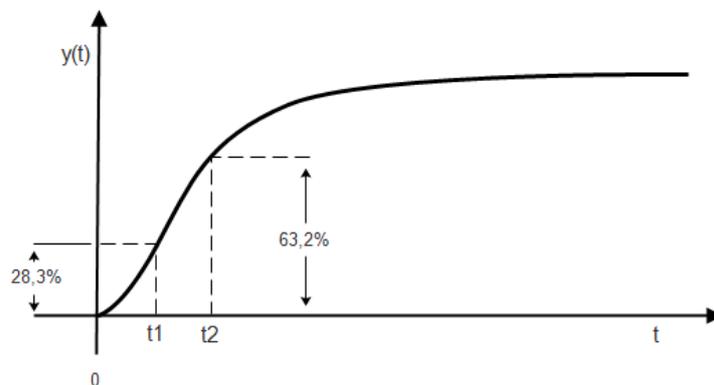


Figura 2.14: Curva de reação do processo para a aplicação de um degrau.

O método de Cohen-Coon foi proposto por C. H. Cohen e G. A. Coon em 1953 (ÅSTRÖM *et al.*, 1993), enquanto o método C-H-R por K. L. Chien, J. A. Hrones e J. B. Reswick em 1952 (SHAHROKHI *et al.*, 2013). Apesar de os dois serem diferentes, um com foco em estabilização rápido no setpoint e o outro em aplicação de rampa, ambos utilizam as mesmas variáveis para ajuste do PID: o tempo morto d , também escrito na literatura como θ , que é indicado quando a variável de controle (VC) passa a ser alterada e a constante de tempo τ , que indica quando a VC atingiu 63,2% do *setpoint* com redução do tempo morto. Podemos

obter essas variáveis através da figura utilizando as equações 2.6 e aplicando conforme definido na tabela 2.1 para o respectivo método.

$$\tau = 1,5 \cdot (t_2 - t_1) \quad (2.7)$$

$$d = t_2 - \theta \quad (2.8)$$

2.2.5.4 – Outros Métodos

Diversos outros métodos de ajuste estão disponíveis para uso, sua escolha deve ser feita com base no experimento. Em nosso caso, o constante *overshoot* no ajuste do método de ZN não prejudica nosso sistema, mas pode ser que não seja o mesmo para outros, uma alternativa é o método C-H-R sem *overshoot*, que consiste em uma modificação do ZN. Mas além desses métodos convencionais pode ser interessante o uso de métodos não convencionais, usualmente inteligentes para automação do ajuste do PID, como, por exemplo, rede neurais (ÜNAL, 2012) ou busca harmônica. Porém, isto deve ser feito com cautela visto que diferente dos métodos convencionais, que são amplamente utilizados e otimizados, os não convencionais ainda estão em fase experimental.

2.3 – Aquisição dos Dados/Instrumentação

O intuito dos experimentos se trata de obter dados que serão analisados com um determinado fim, para isso é necessário que esses dados sejam armazenados em algum lugar. Com computadores isso se tornou mais simples, basta possuir um *software* obtendo e armazenando os valores obtidos por um *hardware* através de uma interface de comunicação, mas ocorre uma dificuldade em como realizar a aquisição destes dados.

Neste trabalho em particular, estamos interessados nos dados que são obtidos de equipamentos de medida de propriedades físicas. Estes são os instrumentos, ou *hardwares*, que utilizaremos nos experimentos. Para que o *software* seja capaz de interagir com o instrumento é necessário passar por diversos processos, como vemos figura 2.15, sendo a camada inferior a de nível mais baixo, referida assim por se tratar de interação direta com a arquitetura do *hardware*. Enquanto a superior é a de nível mais alto, apresentada ao usuário através de uma interface gráfica ou, tratando de programação, é onde os comandos são mais próximas da linguagem falada. Não é necessário que o usuário saiba utilizar ou como funciona cada uma dessas camadas, entretanto, sua compreensão permite que sejam desenvolvidas aplicações que utilizem o máximo de potencial dos instrumentos através de otimizações.

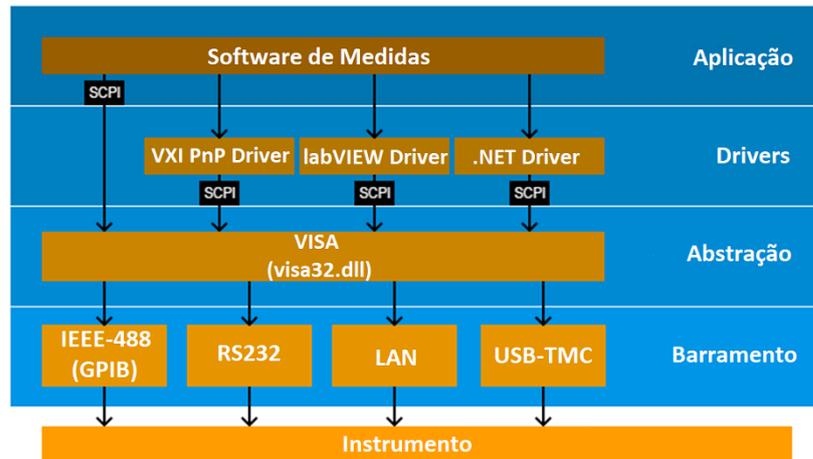


Figura 2.15: Diagrama das camadas em comunicação com equipamentos.

A esquematização representada pela figura 2.16 é dada pela fundação IVI (VXIplug&play, 2010). Este padrão foi estabelecido com o intuito de que os diversos instrumentos de medida disponíveis no mercado tenham entre si algumas compatibilidades, mesmo aqueles introduzidos antes deste padrão e que possam vir a utilizar outras lógicas. Sua introdução facilitou o uso por parte do usuário, podendo reutilizar aplicações já desenvolvidas para outros instrumentos, além de definir um padrão aos protocolos de comunicação. Também foi definido como e com quais materiais devem ser confeccionados os instrumentos por parte dos fabricantes.

Dos protocolos de comunicação temos os barramentos: LAN, RS-232, IEEE.488 e outros. Porém, conforme são utilizados vários equipamentos e com isso diversos barramentos em um mesmo experimento, ocasiona em diferentes processos de comunicação necessários simultaneamente, mas pela padronização da fundação IVI definiu-se uma camada de leitura e saída denominada VISA (*Virtual Instrument Software Architecture*) (OBRENOVIĆ, 2006), que diminuiu a complicação nestes casos.

Apesar da padronização na comunicação, ainda havia uma grande dispersão referente aos comandos utilizados pelos instrumentos, de modo a estruturar a forma que os comandos deveriam ter e quais se tornariam obrigatórios. Foi definido o padrão SCPI (SCPI CONSORTIUM *et al.*, 1999), todavia, nenhum comando novo foi definido, foram reestruturados com base nos já existentes e presentes nos instrumentos. Desta forma, programar instrumentos se tornou menos complexo, pois era comum um fabricante de multímetros, por exemplo, comercializar instrumentos cujas funções eram similares, apenas com diferença de uma versão para outra, porém com todos os comandos possuindo sintaxe diferente. Mesmo que não tenha resolvido o problema totalmente, já que alguns fabricantes não

seguem o padrão e apenas obedecem a introdução dos comandos obrigatórios, este tipo de problema se tornou muito mais raro de se encontrar.

Para a realização da comunicação temos três posições que englobam o funcionamento de interface que são: *Talker*, *Listener* e *Bus Controller*. Como o nome já deixa claro, o *Talker* é o responsável pela transferência de informação, enquanto o *Listener* a recebe. O *Bus Controller*, por sua vez, é quem vai ditar os comandos a serem executados, na grande maioria, e no nosso caso, essa posição é preenchida pelo computador, enquanto as outras duas são feitas pelos instrumentos de medida.

2.3.1 – Camada dos Barramentos

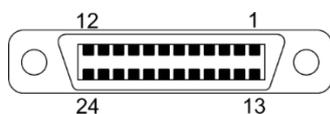
Atualmente temos a nossa disposição uma gama muito grande de interfaces com diversos tipos de barramentos disponíveis para interação com o computador, porém até o fim dos anos 60 o único disponível consistia no RS-232. Enquanto este era utilizado de modo geral para diversas funções, se tinha a necessidade de desenvolvimento de um novo barramento, cujo foco fosse exatamente para a automatização de experimentos, com características bem diferentes das que o RS-232 disponibilizava. A partir disso então foi criado o então chamado de HP-IB (*Hewlett-Packard Interface Bus*), atualmente conhecido como GPIB (*General Purpose Interface Bus*) ou pelo nome de protocolo IEEE-488. Porém, diferente dos anos 60, atualmente encontram-se no mercado interfaces de custo mais baixo, como USB e LAN, que já estão presentes em diversas aplicações do dia a dia, enquanto o RS-232 foi substituído ou retirado dos computadores.

Nesta seção iremos discutir sobre a funcionalidade e características de alguns desses barramentos e a partir de quais dessas informações se dá a escolha para um determinado experimento.

2.3.1.1 – IEEE-488

Após ser aceita pela IEEE (*Institute of Electrical and Electronics Engineers*), em 1975, e receber o nome de IEEE-488, esta interface possui barramento paralelo de 8 bits, como visto na figura 2.16. Possui 24 linhas de sinal sendo elas: 8 para transferência de dados, 3 para *handshake* e 5 para controle, além de 8 para o terra.

<i>Sinal</i>	<i>Pino</i>	<i>Função</i>		<i>Sinal</i>	<i>Pino</i>	<i>Função</i>
--------------	-------------	---------------	--	--------------	-------------	---------------



<i>DIO1</i>	1	<i>Dado/commando</i>	<i>SRQ</i>	10	<i>Requisição de serviço</i>
<i>DIO2</i>	2	<i>Dado/commando</i>	<i>ATN</i>	11	<i>Atenção</i>
<i>DIO3</i>	3	<i>Dado/commando</i>	<i>Shield</i>	12	<i>Cable shield</i>
<i>DIO4</i>	4	<i>Dado/commando</i>	<i>DIO5</i>	13	<i>Dado/commando</i>
<i>EOI</i>	5	<i>Fim ou identidade</i>	<i>DIO6</i>	14	<i>Dado/commando</i>
<i>DAV</i>	6	<i>Dado inválido</i>	<i>DIO7</i>	15	<i>Dado/commando</i>
<i>NRFD</i>	7	<i>Sem leitura pro dado</i>	<i>DIO8</i>	16	<i>Dado/commando</i>
<i>NDAC</i>	8	<i>Sem dado aceito</i>	<i>REN</i>	17	<i>Controle remoto</i>
<i>IFC</i>	9	<i>Interface limpa</i>	<i>Terra</i>	18...24	<i>Terra</i>

Figura 2.16: Barramento IEEE-488 com os 24 pinos nomeados na tabela ao lado.

Os 8 bits para transferência de dados podem ser manipulados diretamente, entretanto isso requer uma programação mais próxima ao nível de máquina. Porém, utilizando o padrão de programação VISA, é possível interagir através dos comandos definidos pela SCPI de modo mais simples. Os 3 bits para *handshake* servem para definir que a conexão entre o equipamento e o *software* foi estabelecida com sucesso ou falhou. Os 5 bits para controle estão presentes e bem definidos em todos equipamentos, pois eles possuem a sua identificação (ID), variando de 0 a 31 que são definidos no equipamento e em seguida no *software* para localizá-lo.

Neste tipo de interface, o *software* torna-se o controlador, através dele executamos os comandos que passam pelo barramento até o equipamento. Possuindo alguns comandos básicos, mas de vital importância, como “*Interface Clear*” (IFC) que limpa os dados que podem estar a espera para serem transferidos, ou “*Remote Enable*” (REN) que torna o equipamento acessível remotamente. O conjunto de comandos composto pelos dois citados acima e alguns outros, são chamados de eventos de interface, pois na sua execução são utilizados pinos especialmente reservados para eles. Podemos observar isso olhando a tabela da figura 2.17, onde o comando IFC tem o pino 9 reservado para si e o REN tem o pino 17.

A interface GPIB que possuímos no laboratório é conectada ao computador através da interface USB e aos até quinze equipamentos, que é capaz de permitir o controle. Podemos então realizar medidas em diversos equipamentos simultaneamente com baixa latência e, caso ocorram valores mais altos, o motivo deve estar diretamente ligado ao equipamento, mas não a interface em si. Isso porque a interface determina a taxa de transferência máxima com base no equipamento mais lento que estiver conectado, independente dos outros.

A vantagem de a interface ser conectada ao computador via USB consiste na praticidade de poder movê-la, se necessário, para outro laboratório, visto que as mais antigas eram ligadas ao conector PCI, portanto era necessário abrir o gabinete do computador sempre que fosse manipulá-la. Ainda que o barramento PCI seja mais rápido que o USB, é possível atingir com a GPIB via USB uma transferência de 8 MB/s, portanto mesmo que sejam utilizados vários equipamentos simultaneamente ela consegue satisfazer todas as necessidades, já que os valores usualmente transferidos nas interações consistem em valores de poucos bits, bem abaixo do disponível.

Para a utilização desta interface podemos conectá-la ao computador, como exibido na figura 2.17. Este é o modo em pilha, existem outras formas, mas isso não implica uma diferença na velocidade de transferência. O que restringe neste caso é apenas um limite máximo de 4 metros entre os equipamentos e um total de 20 metros de cabo, mantendo nesse regime se garante uso normal, ultrapassando isto podem surgir pequenos atrasos ou, no pior dos casos, interferência nas medidas.

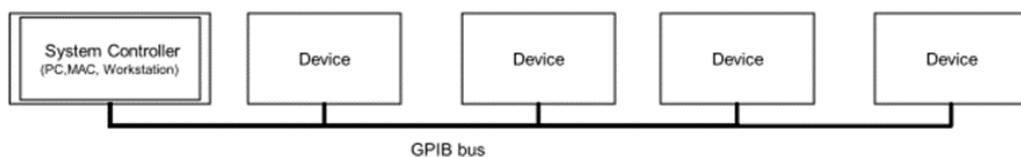


Figura 2.17: GPIB-via-USB ligada a diversos equipamento simultaneamente.

Além dos *eventos de interface* definidos logo em sua criação, com a definição da IEEE-488.2 foi definido que os comandos comuns da tabela 2.2 devem ser obedecidos. Ainda, este estabeleceu o importante Protocolo de Troca de Mensagem, pois havia a necessidade de se garantir uma conversa estável e efetiva, sendo necessário então que se definisse uma diferença clara entre apenas um comando e uma requisição. Sendo que no primeiro há apenas o entendimento de qual função executar, no segundo, além disso, há de se retornar uma resposta, cujo executar deve estar esperando, caso contrário a interface irá entrar em modo pendurado ou desligado e irá retornar um erro. Nesse protocolo fica também definido que, em nenhum momento além dos definidos, os instrumentos devem ficar conversando, deste modo quando é registrado erro não é enviada a mensagem com o erro, pois não corresponde ao comando enviado anterior ao erro. Para compreender o que aconteceu o controlador que deve requisitar o erro no momento seguinte.

Tabela 2.2: Conjunto de comandos comuns a todos instrumentos de medida.

Comando	Grupo	Descrição
*IDN?	System Data	Identification query
*RST	Internal Operations	Reset

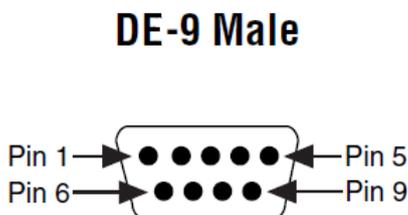
*TST?	Internal Operations	Self-test query
*OPC	Synchronization	Operation complete
*OPC?	Synchronization	Operation complete query
*WAI	Synchronization	Wait to complete
*CLS	Status and Event	Clear status
*ESE	Status and Event	Event status enable
*ESE?	Status and Event	Event status enable query
*ESR?	Status and Event	Event status register query
*SRE	Status and Event	Service request enable
*SRE?	Status and Event	Service request enable query
*STB?	Status and Event	Read status byte query

2.3.1.2 – RS-232

Esse barramento não teve seu desenvolvimento direcionado a equipamentos eletrônicos para medidas científicas em um determinado ponto. Este nunca foi seu foco. Inicialmente era utilizado essencialmente para comunicação em *modems*. Apesar de ter sido proposto como um padrão de comunicação serial digital, apenas em 1969, com a revisão RS-232C que ele passou a ser tratado desta forma, sendo então incluído nos computadores. Isso levou o seu uso a se tornar muito comum até recentemente, pois consistia num padrão relativamente simples e barato, sendo ainda possível encontrar em diversos laboratórios instrumentos que o utilizam, ou que ao menos possuem a porta disponível para uso.

Porém, diferentemente da IEEE-488, discutida anteriormente, essa não é considerada uma porta paralela, mas sim uma porta serial. Neste caso, o que se tem é uma transmissão de bit a bit, não simultânea como antes, isso implica que não se necessita dos cabos grossos e caros do barramento anterior, mas sim cabos simples. Todavia, há uma dificuldade introduzida em função disso, principalmente em experimentos mais complexos, que são de velocidades baixas de transmissão de dados e ligação única. Por exemplo, tomando um experimento onde se tem três equipamentos que precisam interagir simultaneamente, são obrigatórias três portas RS-232 no terminal central. Mas caso isso seja satisfeito, há então a grande vantagem de uso de ser capaz de utilizar cabos de até 15 metros ou, para cabos de alta qualidade, até mesmo 150 metros.

Temos na figura 2.18 a representação do barramento e a definição do pinos para um conector do tipo DE-9, porém o padrão RS-232 foi originalmente especificado como tendo 25-pinos. A redução na quantidade de pinos foi por uma questão de simplificação, visto que diversos eram poucas vezes utilizados e acabavam por apenas deixar o conector demasiadamente largo.



<i>Pino</i>	<i>Sinal</i>	<i>Função</i>
1	<i>DCD²</i>	<i>Data Carrier Detect</i>
2	<i>RXD</i>	<i>Received Data</i>
3	<i>TXD</i>	<i>Transmitted Data</i>
4	<i>DTR²</i>	<i>Data Terminal Ready</i>
5	<i>GND</i>	<i>Signal Ground</i>
6	<i>DSR²</i>	<i>Data Set Ready</i>
7	<i>RTS</i>	<i>Request to Send</i>
8	<i>CTS</i>	<i>Clear to Send</i>
9	<i>RI²</i>	<i>Ring Indicator</i>

Figura 2.18: Barramento RS-232 com os pinos nomeados na tabela ao lado.

Dos comandos listados temos dois de extrema importância, que são os referentes aos pinos 2 e 3, é através deles que é transmitido e enviado os dados. Conforme comentado anteriormente, eles seriam os dois necessários para diminuição do barramento. Além deles temos uma sequência de funcionamento interessante que podemos observar a partir da tabela, consiste em como é feito o início da interação do instrumento para o computador. Para isso, o instrumento acionaria o pino 7, Request to Send (RTS), informado que está pronto para receber dados e o computador, por sua vez, deve acionar o pino 8, Clear to Send (CTS) respondendo que está pronto para enviar os dados.

2.3.1.3 – Outros Barramentos

Como alguns barramentos se tornam muito populares e presentes na maioria dos computadores, fabricantes passaram a perceber que isso permitiria uma implementação de instrumentação de modo mais barato e fácil. Esse é o caso dos barramentos USB (*Universal Serial Bus*) e *Ethernet*, que tem sido introduzidos em quase todos os instrumentos de medida mais recentes e estão em todos computadores, desde os de alta performance até os pessoais.

Entretanto, por não terem seu desenvolvimento inicial direcionado a instrumentação, foram realizadas modificações destes e introdução de outros padrões com base em novos protocolos, respectivamente, USBTMC (*USB Test and Measurement Class* ou *USB Classe de Teste e Medida*) e LXI (*LAN eXtensions for Instrumentation* ou *LAN com eXtensão para Instrumentação*).

Porém, ainda é necessário cautela ao se escolher o barramento, visto que, por exemplo, enquanto o USBTMC permite uso *plug-and-play* ao emular comunicações do tipo GPIB, o LXI necessita de configurações a parte. Assim, este último se torna mais interessante para casos em que seja necessário acesso remoto, pois permite comunicações com distância ilimitada, com as devidas extensões. Porém, enquanto o primeiro é limitado em distância, sua capacidade de emulação é muito interessante, por permitir compatibilidade com todos instrumentos e comandos GPIB, pois atua como se o fosse.

2.3.1.4 – Largura de Banda e Latência

Na tabela 2.3 temos uma comparação básica entre os diversos barramentos, onde nos eixos se encontram a latência e a largura máxima da banda, além de outras características. Essas duas costumam ser as primeiras observadas quando se trata da escolha de um barramento, visto que usualmente o ponto de limitação de um sistema é dado por elas. Por exemplo, caso se saiba que o experimento vai gerar um fluxo de dados muito grande é interessante escolher um cuja largura de banda seja suficiente para sustentá-lo. Ou ainda pode ser importante, como é no caso de experimentos com diversos equipamentos interagindo, que as medidas sejam realizadas de forma simultâneas, neste caso o interesse é que se tenha uma latência baixa.

Tabela 2.3: Apresentação de alguns parâmetros importantes a serem considerados na escolha no barramento a ser utilizado no experimento.

	Largura de Banda (MB/s)	Latência (ms)	Distância (m) (sem extensores)	Instalação e Configuração	Robustez do Conector
GPIB	1,8 (488.1) 8 (HS488)	30	20	Bom	Melhor
USB	60 (Hi-Speed)	1000 (USB) 125 (Hi-Speed)	5	Melhor	Bom
PCI	132	0,7	Barramento Interno no PC	Ótimo	Melhor
PCI Express	250 (x1) 4000 (x16)	0,7 (x1) 0,7 (x4)	Barramento Interno no PC	Melhor	Melhor
LAN	12,5 (Fast) 125 (Gigabit)	1000(Fast) 1000 (Gigabit)	100 m	Bom	Bom

Da tabela podemos retirar que há barramentos como a LAN que permitem um maior fluxo de dados, porém com latência também maior, enquanto outros como a GPIB tem um fluxo menor mas uma latência baixa. Em nosso experimento, por termos uma necessidade de que as medidas sejam realizadas simultaneamente e cada único ponto obtido em intervalo de

tempo da aproximadamente 0,5 a 1s, temos exatamente o que a GPIB melhor proporciona e por isto a escolhemos para nossa montagem, apesar de quase todos os instrumentos disponibilizarem outros barramentos, como LAN e USB.

Como vemos, os diversos barramentos apresentam uma série de diferenças em propriedades, sendo, à primeira vista, necessário a desistência da qualidade de um parâmetro em função do outro. Entretanto, isso não necessariamente é um problema, pois não é obrigatório que o experimento seja composto por instrumentos de mesmo barramento, portanto tem-se a possibilidade de criação de sistemas híbridos, no qual o barramento pode ser escolhido de forma específica para cada instrumento, assim obtendo o máximo proveito de cada.

2.3.2 – Camada de Abstração de Hardware

Essa camada, também chamada de camada de comunicação, é a responsável por traduzir as mensagens do instrumento para o computador e vice-versa, já que o modo como os dois se comunicam é bem diferente. Isso é muito conveniente, pois permite que aplicações sejam capazes de interagir com o instrumento sem que seja necessário compreender como ele funciona nos mínimos detalhes, em vez disso define-se um certo conjunto padrão de comandos em comum que fica a cargo usualmente da CPU, ou no nosso caso do dispositivo GPIB, para então a aplicação poder se comunicar utilizando linguagens de programação de nível alto – como Python, C++, entre outras - sem se preocupar com dificuldades de uso em linguagens de nível baixo – como Assembly.

No caso de instrumentação, o padrão utilizado para esta camada é o VISA (*Virtual Instrument Software Architecture*), introduzido pela VXIplug&play, que consiste em uma biblioteca de entrada/saída (*I/O*), que possibilita ser uma ferramenta muito útil para comunicação com instrumentos de trabalho seja usando GPIB, RS-232(*serial*), USB, *Ethernet* ou outros. Isso é importante, pois permite a interoperabilidade entre equipamentos de diferentes fabricantes, visto que o padrão não é definido para um determinado mas sim de modo amplo.

Esta é a parte central de toda montagem, visto que ela faz a ligação entre o instrumento e o controlador. Assim como o padrão SCPI, comentado anteriormente, ela possui algumas funções que veremos que são aplicadas constantemente. Dentre elas, temos os comandos presentes na tabela 2.4, que são extremamente importante, visto que se fôssemos fazer uso direto destes comandos poderíamos compor diversas rotinas inteiras baseando-nos apenas neles. O modo como esses comandos podem ser acessados varia para o sistema operacional, no caso do Windows é utilizada a *DLL* (biblioteca de vínculo dinâmico) *visa32.dll* ou

visa64.dll, que engloba todos os comandos possíveis, formato dos dados, dos valores, tipos de respostas, erros permitidos e outras informações acerca da comunicação conforme definido pelo VPP-4.3. Para se obter tal biblioteca é necessária a aquisição através de algum fabricante, usualmente a diferença se dá em apenas alguns comandos, mas de modo geral o recomendado é de que se obtenha conforme o equipamento que se utiliza. No nosso caso, realizamos a instalação da *National Instruments VISA* (NI-VISA) para obtê-lo, mas também poderíamos o ter feito através da *Agilent IO Libraries*, ou qualquer um dos diversos outros disponíveis. O infortúnio no uso da biblioteca de tais fabricantes é que são proprietárias, caso houvesse interesse em comercializar uma aplicação que use a *DLL* da NI-VISA seria então necessário a aquisição de permissão para tanto, cujo valor comercial é alto e só vale para uma única versão da biblioteca. Para superar este problema alguns usuários iniciaram o desenvolvimento de bibliotecas abertas que implementam tais métodos e atributos e que, além disso, facilitem o uso em outros sistemas, como é o caso da LibreVISA e da PyVISA-py.

Tabela 2.4: Alguns comandos definido pelo padrão VISA.

Nome	Comando	Função
VISA Open Default RM	viOpenDefaultRM()	Abre um <i>Resource Manager</i> para as conexões
VISA Find Resource	ViFindRsrc()	Encontra uma determinada conexão
VISA Open / Close	viOpen()	Abre ou fecha a conexão
VISA Write	viWrite()	Envia um comando pela conexão para o instrumento
VISA Read	viRead()	Faz a leitura do instrumento

2.3.3 – SCPI (Standard Commands for Programmable Instruments)

Como já comentado anteriormente, devido à grande quantidade de instrumentos com comandos diferentes, foi-se definido o padrão SCPI para simplificar isto. Todos os comandos podem assim ser enviados, por exemplo, utilizando o *viWrite* definido na subseção anterior. É interessante portanto entendermos como é dada a estrutura destes comandos.

Os comandos tem duas formas em especial, os que terminam com interrogação e os que não. Isto tem um significado bem determinado pelo padrão SCPI, sendo que a sinalização de interrogação no final de qualquer comando, seja comum ou subsistêmico, indica que se trata de uma requisição (*query*) e portanto o instrumento deverá retornar uma resposta ao

controlador. Os que não possuem este sinal, indica apenas que deverá ser realizada alguma função, porém sem a necessidade de resposta. Além disto, temos em todos estes comandos a presença de um asterisco (*) no início, isso implica que se trata de um comando comum, sendo assim não deve estar presente em nenhum outro comando para não haver confusões. Como podemos ver, esses são bem simples e portanto de fácil utilização, podendo ser utilizado a qualquer momento e obrigatoriamente devem estar presente em todos instrumentos.

Os comandos de subsistema entretanto são bem diferentes, todos eles possuem uma estrutura hierárquica. Esta tem a primeira parte sendo o cabeçalho (*header*), que define a classe na qual os comandos vão seguir. Como exemplo, o *MEASure* é o cabeçalho utilizado por comandos cuja função é de realizar medidas, especificamente as que não necessitam de muitas configurações, por isso usualmente é utilizado com base no padrão do instrumento. Alguns outros entretanto não se limitam apenas a requisição ou escrita, como é o caso do *CONFig*, que define que medida será realizada pelo instrumento e em alguns casos parâmetros como *faixa* e *resolução*. Na tabela 2.5 temos a formação de comandos com base nos dois cabeçalhos citados, cujos comandos são utilizados no multímetro presente neste trabalho. Vemos com isso que, após o cabeçalho é introduzido um comando *composto*, separado por dois pontos (:), relativo a um subcomando que pode vir a ter, como é no caso de *MEASure:VOLTage*, um comando extra que é também separado pelos dois pontos. É relevante notar que, neste mesmo exemplo, temos várias opções de comandos extras, mas que apesar disto não dificultam sua compreensão e modificação, dado que expressam diretamente qual função vão cumprir. Do lado direito da tabela temos os subcomandos referentes ao *CONFig*, como neste caso pode se tratar de configuração do instrumentos, vemos que há uma nova coluna referente aos parâmetros. Isto introduz também outra nomenclatura especificada pelo SCPI que é do uso de chaves [] *na definição do comando* para especificar que se trata de algo opcional e de < > para identificar onde o valor do parâmetro deve ser inserido e suas possibilidades. Com base nessas especificações, poderíamos fazer os seguinte envios de comando:

Tabela 2.5: Possibilidades de uso de comando para configuração.

CONFig?	Requisita qual medida está configurada para ser realizada
CONFig:CURRent:DC	Configura para medir corrente DC
CONF:CURR	Configura para medir corrente DC
CONF:AC 1	Configura para medir tensão AC em 1 V range
CONF 1, 0.00001	Configura para medir tensão DC em 1 V range e 10 μ V resolução

Os comandos podem ser comunicados em forma extensa (*CONFig:CURRent:DC*) ou reduzida (*CONF:CURR*) sem que isso interfira em seu uso.

Apesar de citado apenas dois cabeçalhos, existem muitos outros, como: *CALibration*, *SYSTEM*, *DISPlay*, *MEMory*, *RADio*, *FREQuency* e etc. Obviamente nem todos estarão disponíveis para uso, pois sua disponibilidade depende da capacidade do instrumento, entretanto há dois em específico que são obrigatórios: *STatus* e *SYSTEM*, pois assim como nos comandos comuns há a necessidade básica de sempre se ter fácil acesso ao estado do instrumento, através do primeiro, e de saber se algum erro ocorreu e caso tenha ocorrido qual foi, com o uso do segundo.

Tabela 2.6: Comandos utilizados pelo Multímetro Digital Agilent 34401-A. Adaptado de (KEYSIGHT, 2014).

Comando Base	Composto	Extra	Comando Base	Composto	Extra	Parâmetro(s)
MEASURE			CONFig			
	:VOLTage	:DC?		?		
		:AC?		:CONTinuity		
		:DC:RATio?		:CURRent	:AC	[<range MIN MAX Auto>[,<resolution MIN MAX DEF>]]
	:CURRent	:DC?			[:DC]	[<range MIN MAX Auto>[,<resolution MIN MAX DEF>]]
		:AC?		:DIODE		
	:RESistance?			:FREQuency		[<range MIN MAX Auto>[,<resolution MIN MAX DEF>]]
	:FRESistance?			:FRESistance		[<range MIN MAX Auto>[,<resolution MIN MAX DEF>]]
	:FREQuency?			:RESistance		[<range MIN MAX Auto>[,<resolution MIN MAX DEF>]]
	PERiod?			[:VOLTAGE]	:AC	[<range MIN MAX Auto>[,<resolution MIN MAX DEF>]]
	CONTinuity?				[:DC]	[<range MIN MAX Auto>[,<resolution MIN MAX DEF>]]
	DIODE?					

Enquanto os comandos da tabela 2.6, definidos para o multímetro 34401-A são totalmente compatíveis com o padrão recomendado pelo SCPI, se olharmos a tabela 2.7 temos que aparentemente parte dos comandos para o controlador de temperatura, também referente ao utilizado neste trabalho, em parte não o segue. Isso acontece pois ainda há divergência entre fabricantes, especialmente por terem foco em instrumentos com diferentes funcionalidades. Porém, isso não implica que o dispositivo não respeite o padrão, olhando o lado esquerdo da tabela vemos que os comandos comuns obrigatórios estão presentes, a questão é apenas que esses comandos específicos não possuem uma hierarquia, mas continuam fazendo uso do símbolo de interrogação para requisição e a ausência para escrita.

Tabela 2.7: Comandos utilizados pelo Controlador de Temperatura Lakeshore 335. (LAKESHORE, 2012).

Comando	Função	Comando	Função	Parâmetro(s)
*CLS	Clear Interface Cmd	ALARM	Input Alarm Parameter Cmd	<input>,<off/on>,<high value>,<low value>,<deadband>,<latch enable>,<audible>,<visible>
*ESSE	Event Status Enable Register Cmd	ALARM?	Input Alarm Parameter Query	<input>
*ESE?	Event Status Enable Register Query	ALARMST?	Input Alarm Status Query	<input>
*ESR?	Standard Event Status Register Query	ALMRST	Reset Alarm Status Cmd	
*IDN?	Identification Query	ANALOG	Monitor Out Parameter Cmd	<output>,<input>,<units>,<high value>,<low value>,<polarity>
*OPC	Operation Complete Cmd	ANALOG?	Monitor Out Parameter Query	<output>
*OPC?	Operation Complete Query	ATUNE	Autotune Cmd	<output>,<mode>
*RST	Reset Instrument Cmd	BRIGT	Display Brightness Cmd	<brightness value>
*SER	Service Request Enable Register Cmd	BRIGT?	Display Brightness Query	
*SER?	Service Request Enable Register Query	CRDG?	Celsius Reading Query	<input>
*STB?	Status Byte Query	CRVDEL	Curve Delete Cmd	<curve>
*TST?	Self-Test Query	CRVHDR	Curve Header Cmd	<curve>,<name>,<SN>,<format>,<limit value>,<coefficient>

*WAI	Wait-to-Continue Cmd	CRVHDR?	Curve Header Query	<curve>
------	----------------------	---------	--------------------	---------

2.3.3 – Camada de Controle(Driver) do Instrumento

Essa camada configura a interação entre o aplicativo, ou rotina, desenvolvido para o experimento e os instrumentos. Costuma ser considerada opcional, dado que o *driver* compõe apenas as funções de um instrumento ou de uma classe de instrumentos em particular. É uma camada que facilita a interação com o equipamento, porém adiciona mais uma camada de abstração, cada vez mais distanciando o usuário do contato direto com o instrumento.

Um tipo interessante são os IVI *drivers*, que podem ser separados em duas esferas: *driver* IVI específico ou *driver* IVI de classe. Na primeira, o que se têm são as funções de um instrumento em particular, como o multímetro 34401-A, enquanto a segunda engloba todos os multímetros. Porém, foram definidos apenas para três tipos: *IVI-C*, *IVI-COM* e *IVI.NET*, sendo assim tem-se o mesmo problema que as bibliotecas proprietárias VISA, que tem seu uso limitado ao Windows por utilizar DLLs.

Com o *driver* criado, o modo como se conversa com o equipamento toma uma forma diferente da usual, onde normalmente para se obter algum valor o usuário deve escrever um *script* ou rotina no qual ele mesmo deve inserir os comandos SCPI a serem enviados. Ao se utilizar um *driver* tais comandos já estão definidos em uma *classe* – aqui esse termo é utilizado com referência a programação direcionada a objetos, diferente do uso no parágrafo anterior – a qual o usuário em sua rotina pode acessar através de atributos ou propriedades definidos a ela. Tomemos um exemplo básico, no qual queiramos a temperatura em Kelvin atual de um controlador de temperatura, no caso usual teríamos a rotina no inserto (a) do código 2.1 para interação via comandos SCPI e do inserto (b) para o uso de um *driver* criado pela biblioteca *slave* (HALDER, 2012). As diferenças nas linhas 1, 3 e 4 são dadas apenas pelo fato da biblioteca *slave* realizar algumas modificações sintáticas na biblioteca *visa*, mas na realidade podemos considera-las equivalentes já que em seu código a primeira importa a segunda. O ponto importante está na linha 2, onde se invoca o *driver* LS340 que não é feito do lado esquerdo, portanto resulta na diferença da requisição realizada na linha 6, como mencionado. Enquanto no caso usual é necessário que o usuário conheça o comando ‘KRDG? A’ (onde A é o canal de interesse) no outro é mais intuitivo, em que ele escolhe o canal A com `ls340.input[‘A’]` e, posteriormente, é possível obter diversas propriedades diretamente, no exemplo é usado `.kelvin` que está atribuído a temperatura em Kelvin do sensor no canal A. Outra característica interessante de *drivers* é que, em algumas bibliotecas, é fornecida a unidade de medida, assim caso sejam necessárias realizar operações entre valores entre medidas é

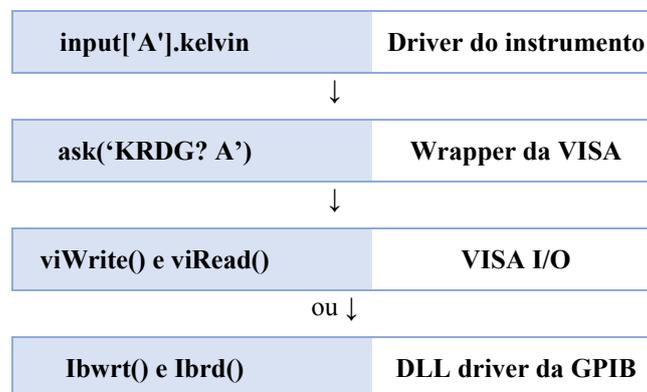
possível se certificar de que não estejam ocorrendo erros de grandeza, além de ser mais fácil posteriormente de tratar este tipo de dado.

Código 2.1: Comparação entre algoritmo para comunicação (a) sem e (b) com uso de driver.

<pre> 1 import visa 2 3 rm = visa.ResourceManager() 4 Ls340 = rm.get_instrument(' GPIB::08') 5 6 print(Ls340.ask('KRDG? A')) </pre>	(a)	<pre> from slave.transport import Visa from slave.Lakeshore import LS340 Ls340 = LS340(Visa(' GPIB::08')) print(Ls340.input['A'].kelvin) </pre>	(b)
---	-----	---	-----

Apesar da diferença de sintaxe e evidentemente da forma como opera, ao olharmos detalhadamente como é feita a execução do acesso ao atributo do *driver* para este caso, notamos que existe uma sequência de comandos, cujas duas linhas inferiores serão detalhadas na próxima seção, com operação similar a estrutura das camadas:

Tabela 2.8: Relação entre comando e camada.



Um problema recorrente é de que com a grande quantidade de instrumentos disponíveis no mercado, apenas os mais utilizados costumam possuir *drivers* e pode ser que os necessários para um experimento estejam disponíveis em bibliotecas separadas. Este é o caso deste trabalho, onde se quiséssemos utilizar para o multímetro Agilent 34401A teríamos que utilizar a *Python-IVI*, enquanto para o LS-335 usaríamos a *slave*, que apesar de não ter um driver para o 335 possui um para o 340 que compartilha grande parte de suas funções. Conseqüentemente, apesar de ser interessante poder utilizá-las, por se tratarem de bibliotecas separadas é possível que as nomenclaturas utilizadas sejam diferentes, funções sejam repetidas e até que hajam incompatibilidades. Tudo isso pode causar problemas ao realizar o experimento sendo assim

deve-se ter cautela ao escolher se usará *drivers* de instrumentos ou não, especialmente em casos onde mais de uma biblioteca é necessária.

2.3.4 – Camada da Aplicação

Nesta camada é desenvolvida a aplicação, com o uso de alguma linguagem de programação, que usualmente serve para automatizar o experimento através da escrita de uma rotina que fará a comunicação com os instrumentos. Com base apenas na *DLL VISA*, sem utilizar outro recurso, é possível o desenvolvimento de rotinas que utilizem linguagem C, BASIC ou G (linguagem do labView), conforme estabelecido oficialmente pela IVI Foundation. Entretanto, atualmente temos acesso a linguagens de programação cuja sintaxe está mais próxima a nossa escrita e portanto são mais fáceis de se compreender. Assim, diversos usuários começaram a desenvolver bibliotecas *wrappers*, cuja função é de expor funções de uma linguagem em outra. Com isto, ao realizar uma rápida pesquisa ao site *GitHub* é possível encontrar bibliotecas que permitem acesso a VISA em diversas linguagens, como: Lua, Perl, Python, Node.js, C#, Matlab, Go, Ruby, Julia, Octave e Rust.

2.4 – Automação

Nesta seção será detalhado como foi feito o processo para escolha da linguagem a ser utilizada para o desenvolvimento do *software* que realizará a automação do experimento, também será discutido como é dado o seu funcionamento, como se realiza medidas e quais rotinas ele é capaz de executar.

2.4.1 – A Linguagem de Programação

No trabalho de conclusão de curso (Santos, 2015), utilizamos a linguagem Python para o desenvolvimento da primeira versão do *software* que automatizou a realização do experimento. Com a proposta de introdução do controle de temperatura, decidimos atualizar o *software* como um todo, introduzindo novas rotinas, modificando a interface gráfica e introduzindo novas funções com base no *feedback* obtido do TCC e com base em relatos sobre o que seria interessante que integrasse a nova versão. Com isso, optamos por continuar utilizando Python, por ser uma linguagem de alto nível que tem uma sintaxe clara e objetiva, dado que o *software* será utilizado por outras pessoas que podem vir a modificá-lo.

Diversas outras linguagens poderiam ser utilizadas para o desenvolvimento do *software*, como C, C++, Visual Basic e outras, mas Python permite que o mesmo trecho de

código das outras linguagens seja escrito de forma mais simples. A forma como sua sintaxe é definida torna-a mais legível, principalmente em códigos mais extensos, pois tem um guia de estilo padrão que recomenda-se ser seguido, a PEP 8, o que limita a quantidade de variações que o código pode ter em relação a outros. Em linguagens como C, a liberdade para o desenvolvimento do código é maior, do ponto de vista de sintaxe. Entretanto, isto abre margem para a utilização de comandos obsoletos como o “goto”, que permite uma movimentação no código sem uma regra específica, quebrando fluxo de controle que deveria ser seguido. Obviamente, isso não significa que sejam linguagens ruins, aliás, em termos de performance, por exemplo, C sempre será mais rápida, pois Python foi desenvolvido utilizando-a, mas a curva para aprendizado é maior.

Poderíamos ainda ao invés de desenvolver um código do ‘zero’, utilizar algum *software* proprietário, como o LabView da National Instrument que permite o desenvolvimento da rotina desejada através de uma interface gráfica, como na figura 2.19. Podemos citar pontos positivos para esta abordagem como a de que não iria falhar devido a pequenos detalhes, o que pode acontecer em linguagens de programação, além de compilar rapidamente sem necessidade de configurações adicionais. Porém, esse *software* precisaria primeiramente ser comprado e tem um alto custo. Além disso, a correção de erros é complicada, pois, quanto maior o número de funções, o estilo de “teia” em que cria as rotinas se torna confuso e estaríamos nos limitando a uma plataforma apenas, Windows. Desenvolvendo o *software* com o Python temos a dificuldade de escrever todo o código para a interface gráfica e para os equipamentos, o que é relativamente trabalhoso, com erros básicos de escrita podendo causar erros e a depuração para encontrar problemas requisitando muito tempo. Mas apesar disto tudo, não temos que pagar pelo *software*, o funcionamento independe da plataforma, a documentação dele é completa, não vai requisitar tanto poder de processamento do computador e observamos em detalhes o funcionamento da interface GPIB se comunicando com os equipamentos.

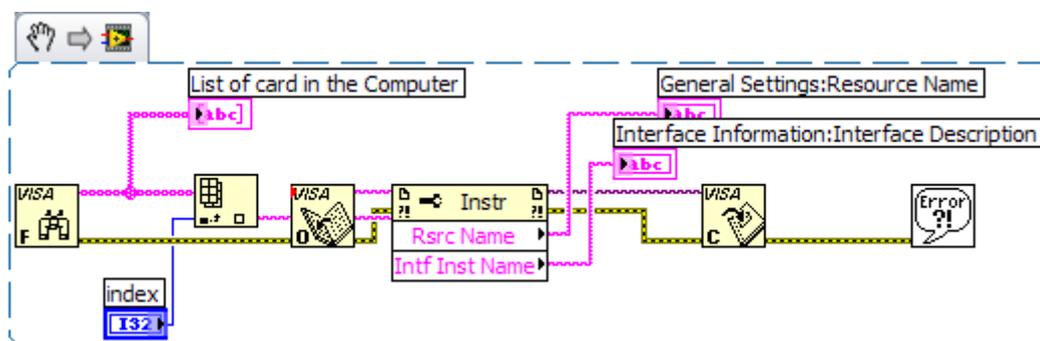


Figura 2.19: Exemplo de funcionamento de uma rotina no LabView.

2.4.2 – Interação Software-Equipamento

Por termos escolhido Python como a linguagem a ser utilizada, retomamos a questão de que não há oficialmente uma biblioteca VISA que permita a interação com os instrumentos para Python, somos levados assim a ter que escolher a biblioteca *wrapper* mais vantajosa para a nossa aplicação. Algumas propriedades da biblioteca são:

- **Número de Commits:** com base neste número de atualização do código, obtido do repositório de arquivos GitHub, temos qual biblioteca é atualizada com maior frequência e portanto introduz novos recursos e correções em menor tempo;

- **Interface Gráfica:** este ponto é importante visto que temos o interesse em desenvolver uma *GUI* (*Graphical User Interface* – Interface Gráfica para o Usuário);

- **Unidades (de medida):** garante facilidade no tratamento dos dados;

- **Camadas de Transporte:** é a parte vital na escolha por definir com que tipo de equipamentos vamos ser capazes de nos comunicar e se teremos que recorrer a outras bibliotecas complementares;

- **Número de Driver:** simplifica a sintaxe do código na comunicação com os equipamentos;

Curiosamente, ao olharmos a tabela 2.9 podemos ver que a biblioteca *PyVISA* não cumpre três pontos listados por nós: ter interface gráfica, exibir unidades e número de *drivers*, mesmo tendo uma quantidade grande de *commits* e ser capaz de se comunicar via todos barramentos. O motivo disto é que o desenvolvimento da *PyVISA* tem finalidade diferente das diversas outras, o que podemos notar do fato de que quase todas a utilizam como camada de transporte. O modo como tem-se arquitetada explicita sua funcionalidade, isto é, com os três seguintes níveis:

Baixo nível: neste nível a biblioteca define os tipos, respostas e conversões necessárias de cada função. **Médio nível:** funções da biblioteca VISA no Python, que são necessárias apenas quando certa funcionalidade ainda não foi implementada em alto nível. **Alto nível:** funções referentes ao controle de recursos, está diretamente ligada ao controle dos equipamentos e é parte essencial do *software*.

Tabela 2.8: Tópicos de comparação entre as bibliotecas disponíveis para instrumentação em Python.

Biblioteca	Nº Commits	Interface Gráfica	Unidades de medida	Camadas de Transporte	Nº Drivers
PyVISA	915	Não	Não	Todas PySerial ou PyVISA(opcional)	0
InstrumentKit	1152	Não	Sim		41

Lantz	306	Sim	Sim	PyVISA	25
Auspex	1925	Sim	Não	PyVISA	
Qcodes	2570	Sim	Sim	PyVISA	~80
Qudi	4371	Sim	Sim	PyVISA	
Crappy	909	Parcial	Não	PySerial	13
Instrumental	952	Sim	Sim	PyVISA	21
PLACE	697	Sim (web)	Não	Serial, Ethernet	~14

Todas: Serial, GPIB, Ethernet, USB, VXI e PXI

Nada, além dessas três capacidades, está introduzida na *PyVISA*, portanto é uma biblioteca cuja única função é de introduzir via Python a possibilidade do uso da VISA disponível no sistema, como dito anteriormente, é um *wrapper* da *DLL*. Sendo que a maior parte das rotinas que a utilizam costuma ser restringidas ao uso apenas das funções de alto nível, pois neste ponto temos todas as funções que nos interessam, que são de escrita e leitura de dados. As outras se tornam interessantes quando possuímos alguma opção básica como acesso remoto do equipamento que pode estar falhando, então acessamos funções de nível médio e nos comunicamos com o equipamento para modifica-lo internamente.

Da tabela 2.9 vemos ainda que, diferente da grande maioria, as bibliotecas *Crappy* e *PLACE* não utilizam de *PyVISA*, apesar disto ser um procedimento comum antes da criação desta, atualmente essa é uma decisão incomum. O motivo disto fica claro ao olharmos o exemplo do código 2.2, em uma situação via *visa* (abreviado de *PyVisa*), inserto (a), versus via *serial*, inserto (b), no qual iniciamos a comunicação com um mesmo instrumento e enviamos um comando SCPI requisitando a tensão atual. Do lado esquerdo temos um código muito mais enxuto que o direito, porém essencialmente tudo o que ocorre do lado direito está introduzido do lado esquerdo, mas há uma camada de abstração para facilitar o uso. Um exemplo para isso é dado pela função *ask* quando comparada com o contraponto também em vermelho, ao olharmos o código fonte da *visa* temos que *ask* é descrita como sendo uma combinação dos comandos *write(mensagem)* e *read*, com um *delay* entre a execução de cada, exatamente o que se tem do lado direito. Por isso, para a função de interação com os equipamento, *PyVISA* se tornou a escolha mais viável.

<pre> 1 import visa 2 3 4 rm = visa.ResourceManager() 5 dmm = rm.get_instrument('ASRL2') 6 7 8 9 10 11 12 13 volt = dmm.ask('MEAS:VOLT:DC?') 14 15 16 17 print(volt) </pre>	(a)	<pre> import serial 2 3 4 with serial.Serial(5 self._serial_port, 6 baudrate=9600, 7 bytesize=serial.EIGHTBITS, 8 parity=serial.PARITY_NONE, 9 stopbits=serial.STOPBITS_TWO 10) as connection: 11 12 cmd = 'MEAS:VOLT:DC?' 13 connection.write(bytes(cmd + '\r', 'ascii')) 14 sleep(0.1) 15 volt = connection.readline().decode('ascii').strip() 16 17 print(volt) </pre>	(b)
---	-----	---	-----

Código 2.2: Comparação entre uso da biblioteca (a) PyVISA e (b) PySerial para uma mesma rotina.

Das diversas outras nós temos funções, assim como as duas anteriores, direcionadas em específico para um determinado campo de pesquisa, como é de se esperar. Temos dentre essas *QCoDeS* para experimento em dispositivos nano eletrônicos, *Auspex* para medidas de *qubits* supercondutores e elementos de memória magnética, *PLACE* para experimentos de laser ultra sônicos e *Qudi* para microscopia confocal, ótica quântica e informação quântica. Entretanto, todos têm em alguma parte de sua documentação a descrição de serem aplicáveis a todos campos experimentais. Porém, por mais que isto seja um ideal que se tenta sempre atingir, e que espero que se seja atingido em certo ponto, não é o que temos visto no código delas. Há um direcionamento da sua arquitetura para a área inicial de aplicação e que raramente introduz facilidade para outras áreas. Com a escolha de realizar o desenvolvimento de um *software* sem recorrer a nenhuma destas bibliotecas além de *PyVISA*, devemos tomar precauções necessárias a não apenas introduzir uma ferramenta a mais no meio de milhares outras. O modo como fazemos isto é direcionar o nosso *software* para o foco em medidas de transporte elétrico.

2.4.3 – Desenvolvimento do software

Na primeira versão do *software* foi obtida a automação do processo experimental com o objetivo somente de monitorar uma propriedade física em função da temperatura. Fizemos isto de modo que os dados são salvos, podendo em sequência serem analisados e retiradas informações sobre a amostra de interesse. Tal monitoramento foi construído de maneira que a latência apresente o menor valor possível, assim garantindo que as medidas correspondem a um mesmo instante, ou seja, sejam síncronas.

Nesta nova versão introduzimos novos instrumentos na montagem e portanto queremos automatizar o manuseio deles através do *software*. Agora pretendemos ser capazes de

especificar uma temperatura no *software*, uma taxa e obter uma resposta do experimento de modo a chegar exatamente ao introduzido.

Para manter um desenvolvimento e futuras mudanças organizadas, foi reformulada a maneira como os arquivos se referenciam no *software*. O módulo principal do *software* é composto por dois arquivos: *app.py* e *app.ui*, respectivamente, a parte lógica e a parte gráfica (interface). Sendo que este inclui as lógicas *homefunc.py* e *dataexibit.py*. Estas duas foram inseridas nessa nova versão para ficar mais fácil o acesso aos dados das medidas realizadas, gravando-os em um banco de dados para caso o usuário perca o arquivo gerado da medida, e para a adição de novas rotinas diretamente pelo *software*. Podemos observar a aparência dessas duas telas nas figuras 2.20 e 2.21.

Os arquivos com extensão *.ui* são criados diretamente através do programa *QtCreator* ou *QtDesigner*, assim evitando a necessidade de escrever longos e complicados códigos para a exibição da interface. Enquanto os com extensão *.py* são os arquivos que nos referimos como tendo a lógica da interface, nele são desenvolvidos os comandos de cada ação da interface.

Os dados obtidos da medida são salvos em um arquivo *.dat*, sendo que as colunas vão depender de como foi pré-determinado no código, usualmente retornando apenas os valores mais importantes, para evitar confusão do usuário. Porém, para o banco de dados não há esta restrição, no caso dos módulos deste trabalho quase todos valores exibidos na interface e que se atualizam simultaneamente são salvos.

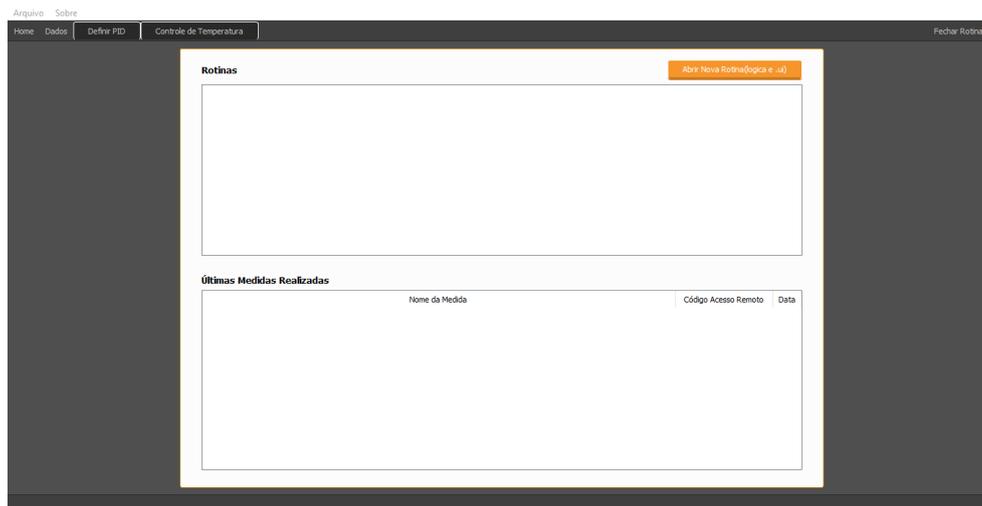


Figura 2.20: Interface Inicial (“Home”) do software, onde são exibidas as rotinas existentes e as últimas medidas realizadas.

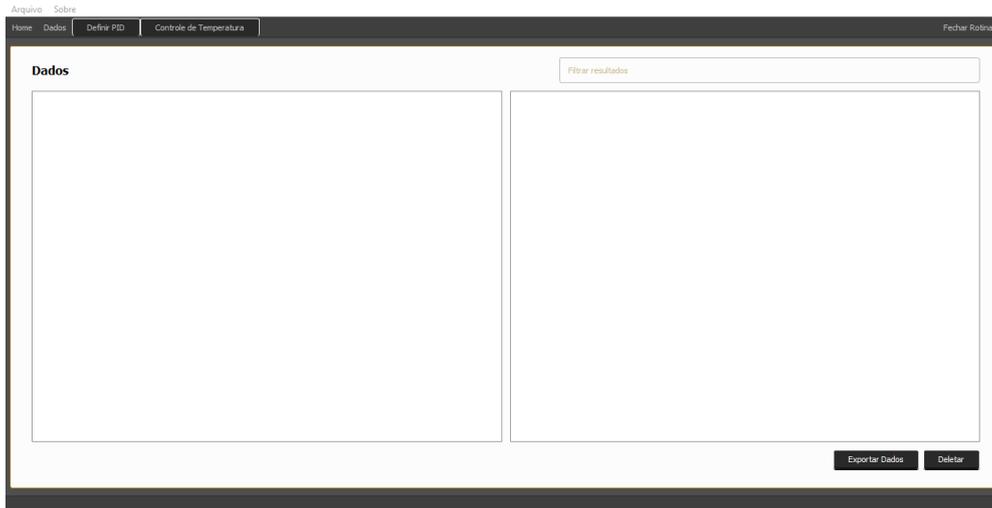


Figura 2.21: Segunda interface básica do software, onde são exibidas as medidas realizadas e os referentes dados obtidos.

2.4.5 – Experimentos

O primeiro módulo desenvolvimento para o *software* foi para auxiliar no ajuste do PID, este é composto pelos arquivos *SettingPID.py*, *SettingPID.ui* e *__MethodsPID.ui*. Diferente do mencionado anteriormente, aqui temos duas interfaces para apenas uma lógica, o motivo disto é de que na janela do ajuste é possível escolher entre outros métodos e portanto é necessário exibir uma pequena janela que necessita de um arquivo. Temos a exibição da janela deste módulo no *software* na figura 2.22.

Ao utilizar este módulo é possível escolher entre os seguintes módulos para o ajuste do PID: Manual, Automático, Ziegler-Nichols e Cohen e Coon. Tirando o método automático, todos os demais métodos exibem a resposta do sistema conforme ocorre a variação de temperatura e dos PIDs. Isto é interessante especialmente para o método manual, pois assim o usuário é capaz de tomar melhores decisões quanto aos novos valores a serem testados. Pelo modo como foi desenvolvido o *software*, é possível inserir facilmente novos métodos sendo necessário apenas criar ou importar a função na lógica e inserir a opção no arquivo da interface, obviamente, é apenas obrigatório que a função retorne os respectivos valores do P, I e D.



Figura 2.22: Interface para ajuste do PID.

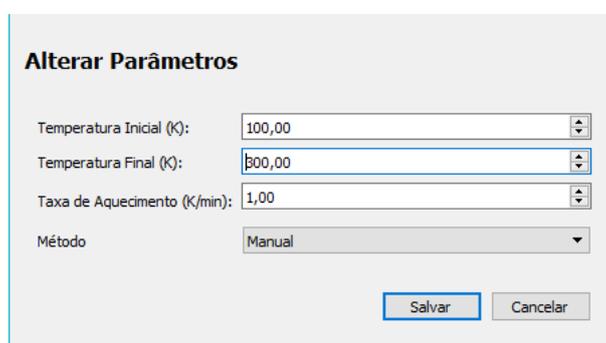


Figura 2.23: Interface de ajuste dos parâmetros do PID.

O segundo módulo é o responsável por realizar o controle da temperatura e a medida das propriedades, sendo composto pelos arquivos: *TemperatureControl.py*, *TemperatureControl.ui*, *__ParametersTemperatureControl.ui*. Na figura 2.24 podemos ver a interface deste módulo, em que podemos separar os valores exibidos em dois blocos: parâmetros e resultados, respectivamente, lado esquerdo – Temperatura Inicial, Temperatura Final, Taxa de Aquecimento, Passo, Ganho – e lado direito – Tensão, Corrente, Resistência, Resistividade, Temperatura, Resistência PT-103, dR/dt , dT/dt . A definição dos parâmetros é dada na interface da figura 2.25, onde se informa a geometria da amostra para o cálculo da resistividade, dadas as temperaturas o próprio *software* determina se está sendo realizado aquecimento ou resfriamento, temos o passo que corresponde quanto de variação de temperatura deve ocorrer para realizar as medidas e o ganho de saída do resistômetro, que requer atenção, pois caso mude as configurações do equipamento requer ajuste manual. Na parte superior da região em preto na figura o que temos ainda é opção para determinar os eixos que se deseja plotar, que por predefinição exibe a resistividade em função da temperatura, mas pode ser qualquer um dos valores do bloco da direita.

Para o uso deste módulo e início do experimento deve-se iniciar o procedimento realizando vácuo na cana de medidas até atingir pressão com valor da ordem de 10^{-2} mbar.

Apenas após isto que a colocamos em banho térmico, onde esperamos atingir temperatura próxima a 77K. Durante o processo de esfriamento verificamos se os cabos de conexão para as medidas estão devidamente conectados e que o resistômetro está com as configurações corretas, pois assim garantimos que ele produza um ganho suficiente para que o multímetro consiga obter valores de tensão que vão além do seu limite. Por fim, iniciamos o *software* na rotina desejada, introduzimos os parâmetros para o experimento – ou deixamos os valores padrão – e iniciamos a medida.

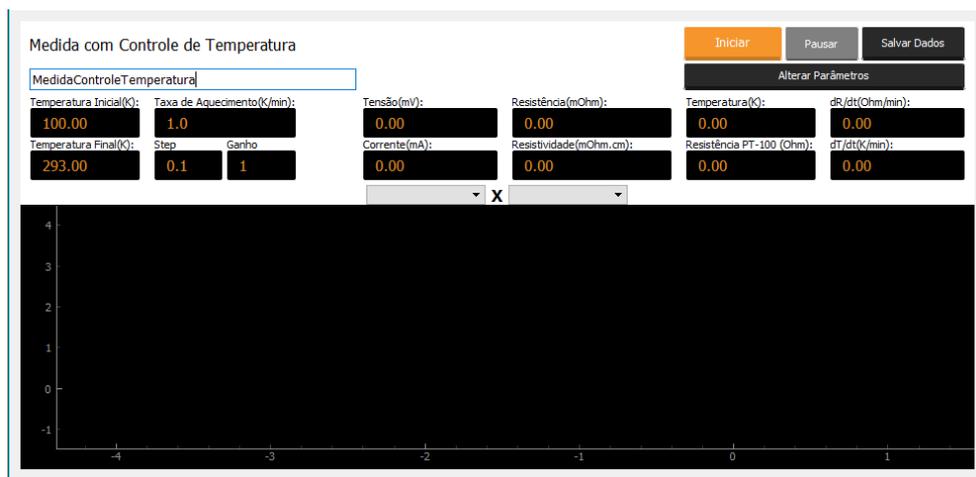


Figura 2.24: Interface para medida da resistividade de um material em função da temperatura.

Figura 2.25: Interface para ajuste do parâmetros da medida.

3 – Resultados

Neste capítulo iremos apresentar os resultados obtidos na calibração do PID, ajuste do controle de temperatura, calibração dos termômetros e da amostra de cobre.

3.1 – Calibração do PID

O método que utilizamos para a calibração do PID foi similar ao de Ziegler-Nichols, no qual primeiramente são zerados todos os parâmetros para em seguida ir ajustando o K_p até atingir uma oscilação quase constante em relação ao *setpoint*. Este é o método de ajuste manual presente na rotina de ajuste dos PID no *software*, o utilizamos para compreensão detalhada do procedimento. Porém, antes de começar o método em si, realizamos alguns testes com diferentes valores de corrente aplicada ao aquecedor. O primeiro valor foi de $i = 0,5A$, por ser muito menor que o valor máximo disponível e consequentemente seguro para o início do experimento, com ele notamos que o uso da faixa de aquecimento *LOW* e *MEDIUM*, de respectivas potência máximas $P = 0,0675W$ e $P = 0,675W$ com a saída manual definida como sendo 100% não foi capaz de aquecer o sistema para temperaturas acima de $T \sim 150K$. Tentamos em seguida aplicar a faixa *HIGH* com saída de 40%, resultando no limite de $P = 2,7W$, inicialmente por aplicar potência abaixo deste limite o aquecimento estava satisfatório, entretanto não foi capaz de atingir temperatura ambiente. Continuamos testando valores mais altos de corrente e diversas combinações de faixa e saída manual, com o último sendo a saída de corrente limite do instrumento, $i = 1,414A$. Neste observamos que, na faixa *MEDIUM* 65%, potência limite $P = 3,51W$, ele foi capaz de aquecer o sistema até $T = 310K$ com 99% de aplicação do limite da potência desta configuração. Com este resultado definimos que a nossa montagem o valor da corrente deve ser $i = 1,414A$ e o limite superior de escolha de potência *MEDIUM* 65%, assim asseguramos que não haverá problemas com o aquecedor, pois caso seja preciso repará-lo e sua resistência mude, será preciso realizar um novo ajuste do PID.

A figura 3.1 representa o comportamento oscilatório da temperatura em relação ao tempo em torno do *setpoint* para três valores de saída manual para um ganho proporcional de $K_p = 320$. O valor do ganho foi escolhido por ser o primeiro que apresentou um comportamento oscilatório com período estável. O uso dos três valores de saída foi para que obtivéssemos uma melhor precisão no ajuste do PID. Para o valor máximo o sistema tem uma dificuldade na precisão da potência e pode oscilar em $T \sim 116,5K$. Tomamos um valor próximo a sua metade para o teste seguinte, 35%. Neste obtivemos oscilação em $T \sim 115,3K$, no

seguinte, 25%, obtivemos o melhor comportamento oscilatório, em $T \sim 114,9K$, que corresponde ao nosso interesse.

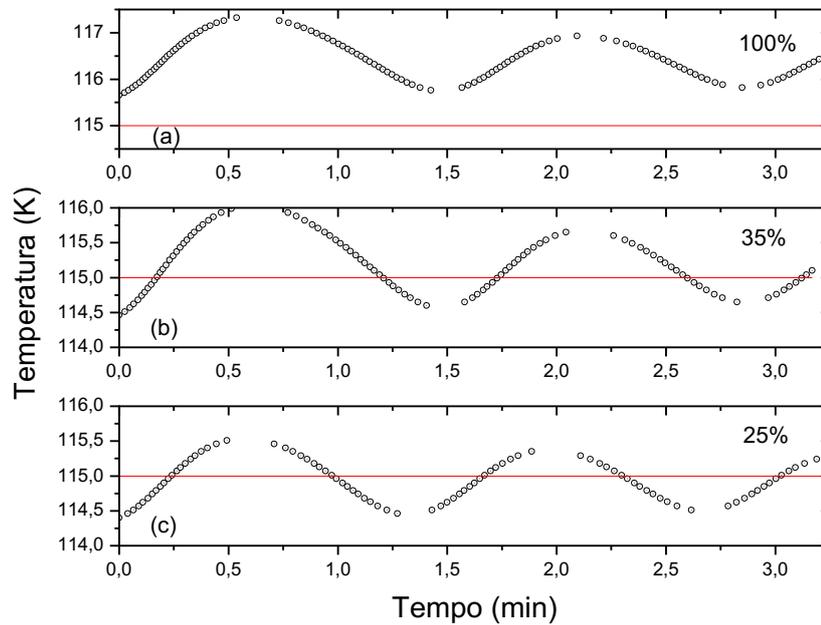


Figura 3.1: Oscilação da temperatura em função do tempo com o setpoint em $T = 115K$ para três variações de saída.

Porém, apesar do proporcional anterior introduzir uma resposta próxima ao *setpoint*, em um intervalo maior de tempo o que veríamos é uma queda de amplitude, enquanto o que queremos é uma estabilização dos picos. Por isto, continuamos aumentando o valor do ganho proporcional para $K_p = 420$ e realizamos duas medidas para o *setpoint* anterior com as saídas 25% e 35%. Na figura 3.2 temos que para 25% o sistema apresenta variação de amplitude, mas para 35% obtivemos amplitude quase constante em $T \sim 115,2K$, que conseguiremos melhorar ainda mais no ajuste fino ao final.

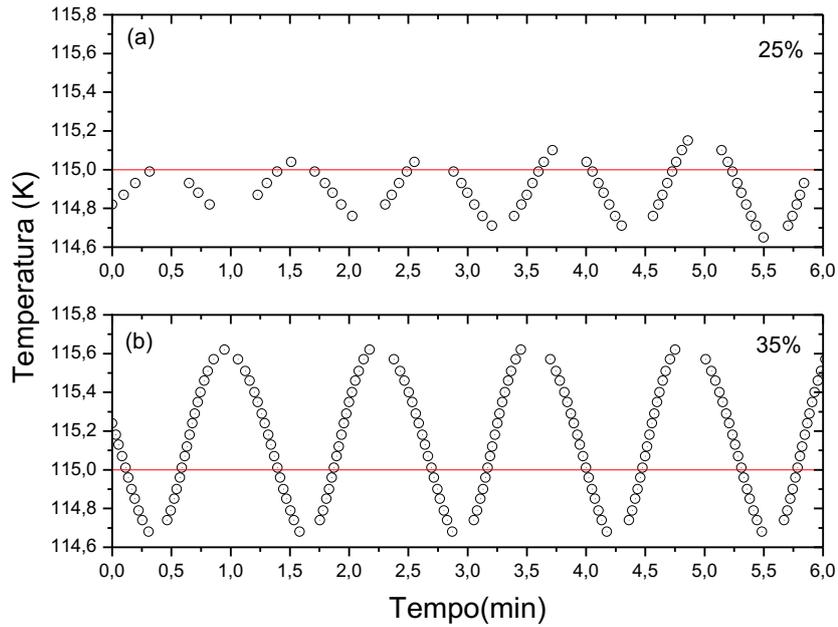


Figura 3.2: Variação da temperatura para dois valores de saída manual com $K_p = 420$ e setpoint em $T = 115K$.

Obtivemos até o momento que a saída manual da corrente deve ser de *MEDIUM* 35% e o ganho proporcional é $K_p = 420$. Calculamos o tempo que o sistema leva para oscilar entre dois picos seguintes para iniciarmos o cálculo do ganho integral T_i . Da figura acima obtemos o período $P = 75$ segundos, disto temos que:

$$T_i = \frac{1000}{75} = 13,3 \quad (3.1)$$

Além do fator integral, definimos o valor do novo ganho proporcional como sendo a metade do K_p utilizado anteriormente:

$$K_p = \frac{420}{2} = 210 \quad (3.2)$$

A partir deste ponto começamos a realizar o ajuste fino dos parâmetros (figura 3.3). Realizamos uma redução no ganho integral T_i , pois queremos a temperatura atinja a *setpoint* com pouca oscilação. Caso quiséssemos atingi-lo rapidamente deveríamos dobrar o ganho K_i , porém com isto obteríamos grandes oscilações. Observamos na figura 3.3 que, utilizando o valor de $T_i = 13$, o comportamento da temperatura atinge picos de até 2 kelvin acima do *setpoint*, quando definimos $T_i = 6,5$, metade do anterior, é notável a queda na amplitude das oscilações.

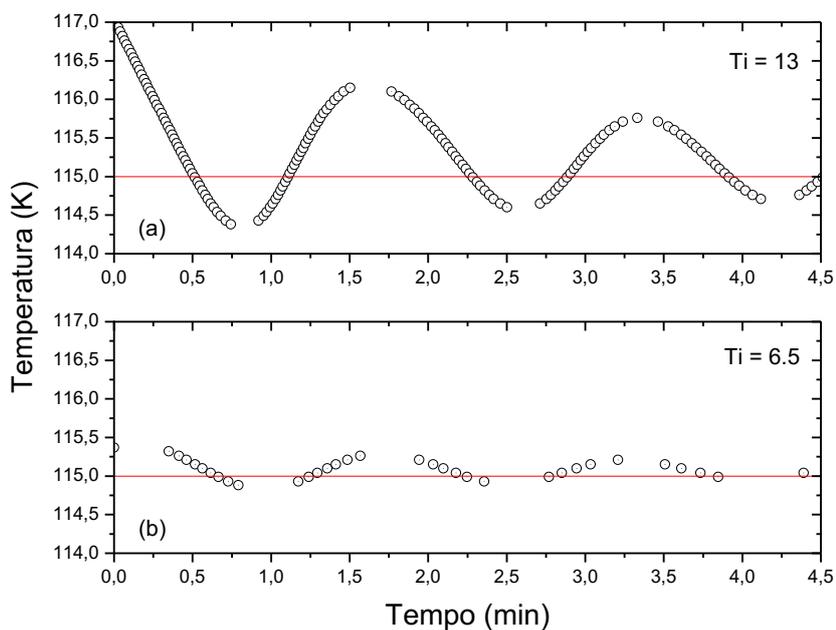


Figura 3.3: Temperatura em função do tempo para $K_P = 210$ com (a) $T_i = 13$ e (b) $T_i = 6,5$ com *setpoint* definido em $T = 115K$.

Para a realização do ganho derivativo o ajuste conforme o método ZN modificado deve ser manual. Portanto, para análise iniciamos o experimento sem ganho derivativo, isto é $T_d = 0\%$, e observamos como se comporta até atingir o *setpoint* em $T = 155\text{K}$. No inserto (a) da figura 3.4 observamos que o sistema apresenta um sobre valor, de aproximadamente 3K acima do *setpoint*, em sua primeira oscilação. Definimos, para uma segunda análise, o ganho proporcional como sendo $T_d = 200\%$, correspondente ao valor máximo do equipamento. Disto obtemos o comportamento apresentado no inserto (b). Observamos uma resposta mais rápida do sistema, que é esperado devido a introdução do ganho derivativo, com sobre valor de apenas 1K e estabilização no *setpoint* em apenas 4 minutos.

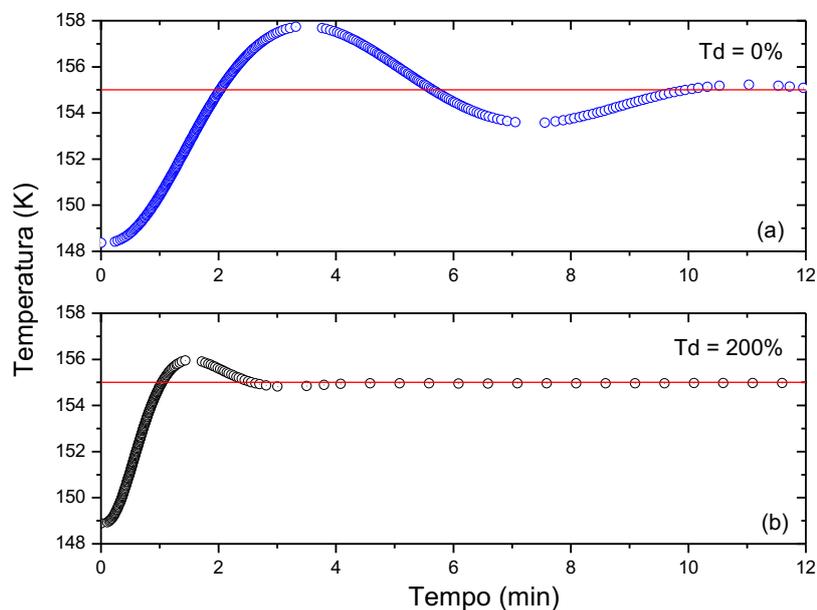


Figura 3.4: Temperatura em função do tempo para $K_P = 210$ e $T_i = 13$ para (a) $T_d = 0\%$ e (b) $T_d = 200\%$ com *setpoint* $T = 155\text{K}$.

Através de ajustes finos nos ganhos com base no resultados obtidos fomos capazes de ajustar o PID para o nosso sistema. Para obter uma melhor precisão nas taxas de aquecimentos obtivemos três regiões com dois PIDs diferentes, um para baixas e outro para intermediárias e altas temperaturas, conforme a tabela 3.1. Da região 2 pra 3 temos apenas a diferença na saída, visto que um valor de 33% não consegue atingir a temperatura ambiente.

Tabela 3.1: Configuração das zonas de controle de temperatura.

REGIÃO	TEMPERATURA LIMITE	P	I	D	SAÍDA MANUAL (MEDIUM E $i = 1,41$)
1	100K	320	10,8	50%	10%
2	250K	140	0,7	100%	33%
3	310K	140	0,7	100%	65%

3.2 – Resultados do PID

O parâmetro que qualifica a capacidade do PIDs, no nosso caso, consiste na função de manter uma taxa de aquecimento constante. Portanto definimos quatro medidas, para validação, a serem realizadas nos intervalos e taxas, conforme apresentado pela tabela 3.2. A última taxa, de 5 K/min, não é interessante para medidas com intuito de pesquisa por ser muito alta e não garantir que os sensores e a amostra estejam em equilíbrio térmico ao realizar as medidas, por isso a realizamos em um intervalo pequeno para testar a capacidade do nosso controle.

Tabela 3.2: Definição de temperatura inicial, final e taxa para os experimentos de validação do PID.

Nº	SET-POINT INICIAL (K)	SET-POINT FINAL (K)	TAXA RAMPA ($\frac{K}{min}$)	DA
1	80	100	0,5	
2	90	300	1,0	
3	100	200	1,5	
4	100	155	5,0	

As curvas de resposta dos nossos PIDs para as taxas e temperaturas obtidas no experimento realizado estão representadas na figura 3.5, sendo que as separamos para a região de $T = 100K$ a $T = 200K$ e de $T = 100K$ a $T = 150K$. Assim, podendo comparar seus comportamentos. No inserto (a) da figura obtemos a variação linear da temperatura, com as diferenças de tempo evidentemente devido a suas respectivas taxas de aquecimento. No inserto (b) as duas menores taxas apresentam o comportamento linear esperado, entretanto na taxa de 5 K/min temos uma variação no intervalo de $T = 100K$ a $T = 110K$, cujo motivo é devido ao controlador de temperatura variar o *setpoint* de acordo com a rampa, porém há um período de

tempo até que o aquecedor aplique potência o suficiente para elevar a temperatura inicial dos componentes no porta amostra até o respectivo *setpoint*.

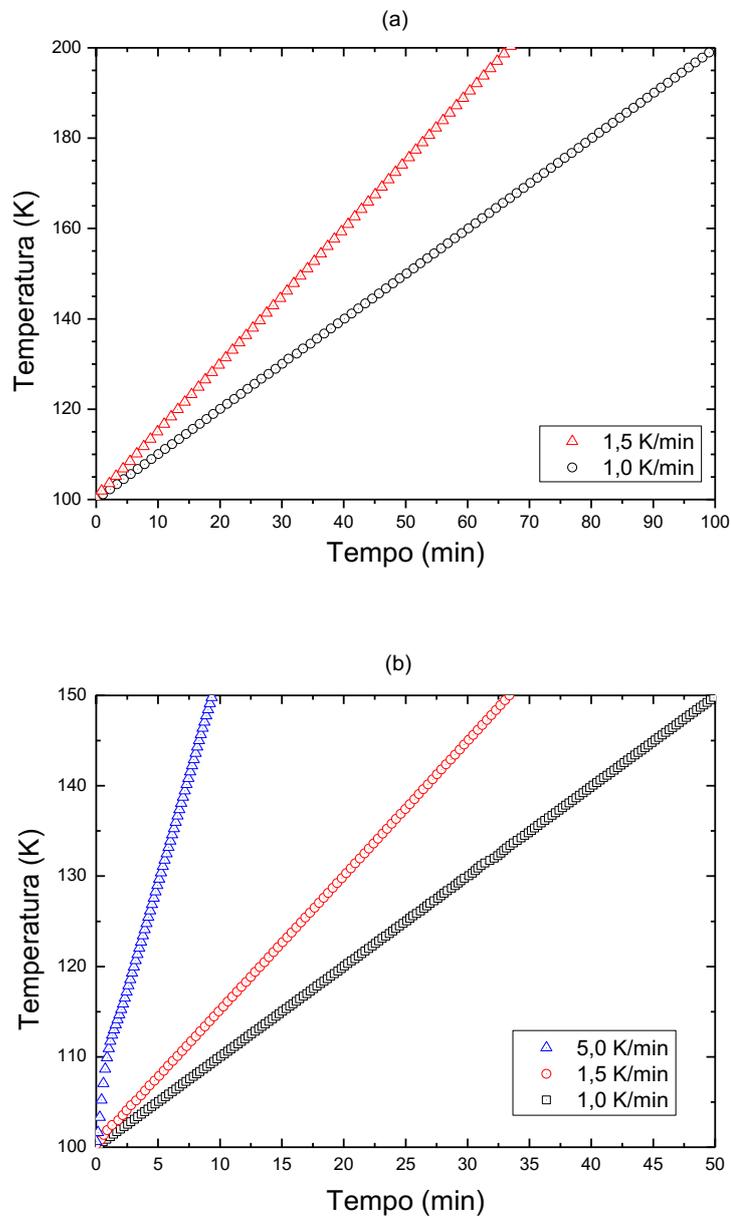


Figura 3.5: Temperatura em função do tempo para os PIDs ajustados com as determinadas taxas de aquecimento nos intervalos de (a) $T = 100\text{K}$ a $T = 200\text{K}$ e de (b) $T = 100\text{K}$ a $T = 150\text{K}$.

Realizando a derivada destas medidas e calculando o valor médio retiramos os valores tabela 3.3 que comparamos com os valores ideais que definimos no instrumento. Temos que, com exceção da taxa de 5 K/min, todas apresentaram valores muito próximos do definido, sendo obtido desvios padrões muito abaixo dos valores das taxas reais. Isso indica que os PIDs

estão bem ajustados para um controle de temperatura de 0,5 K/min até 1,5 K/min com baixa oscilação.

Tabela 3.3: Comparação entre a taxa obtida no experimento, o desvio padrão e a taxa definida.

TAXA DEFINIDA	TAXA REAL	DESVIO PADRÃO
5K/min	4,77K/min	1,82K/min
1,5K/min	1,49K/min	0,08K/min
1K/min	0,99K/min	0,10K/min
0,5K/min	0,50K/min	0,04K/min

3.3 – Amostra Padrão de Cobre

Os dados obtidos através dos multímetros estão exibidos na figura 3.6, sendo a parte superior a tensão medida proveniente do resistômetro e na parte inferior a corrente. Da tensão podemos observar que em dois momentos, $T \sim 125K$ e $T \sim 210K$, ocorre uma descontinuidade do valor medido. Realizando uma medida de curva característica I-V, representado no inserto (c), observamos que a nossa amostra é ôhmica. Portanto, essa descontinuidade pode ser que ocorre devido aos contatos elétricos, que, em virtude das baixas temperaturas, sofrem alterações. A corrente usualmente pode não necessitar ser medida, porém por desejo de uma maior precisão introduzimos um aparato no experimento onde a corrente, após percorrer a amostra, passa por uma resistência padrão de $R = 1\Omega$ e em seguida por dois multímetros. Dessa forma, conseguimos medir a sua tensão e realizar o cálculo preciso de seu valor. Isso resultou que a definição no equipamento de fornecimento de $i = 1mA$ teve uma adição de aproximadamente $i = 0,07mA$, apesar de ser um valor pequeno isto reduz um possível erro caso tomássemos o valor definido como exato.

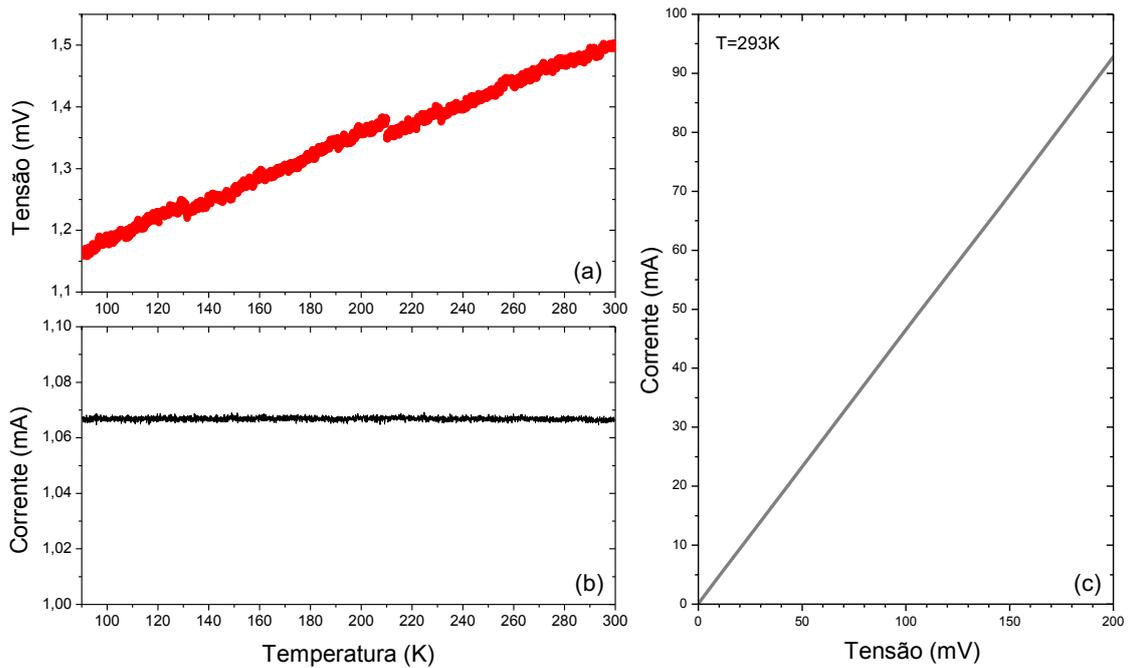


Figura 3.6: Variação da corrente aplicada na amostra em (a), da tensão medida em (b) e curva característica I-V em (c).

Com esses dois resultados em mãos utilizando a relação que descreve aproximadamente a resistência dos metais em função dessa propriedades, usualmente chamada de variação da lei de Ohm $R = \frac{V}{I}$, para assim obtermos a figura 3.7.

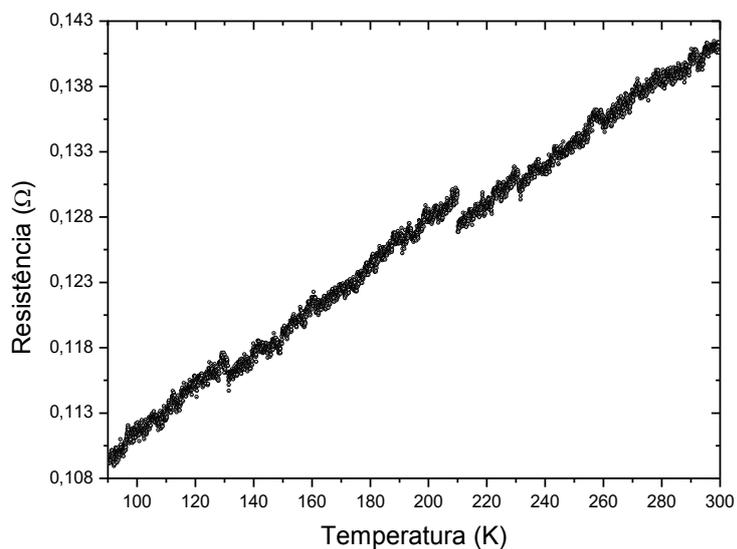


Figura 3.7: Resistência da amostra de Cu em função da temperatura.

Para o cálculo da resistividade da amostra é necessário conhecer sua geometria, como utilizamos um fio de cobre de número AWG 36 sabemos por convenção que isto equivale a

uma secção reta de $A = 0,0127\text{mm}^2$ e também que a distância entre os contatos é de $L = 0,42\text{cm}$. Com isto, chegamos ao resultado ilustrado pela figura 3.8, onde descrevemos o comportamento da resistividade. Em temperatura ambiente ($T = 293\text{K}$) a amostra exhibe resistividade $\rho \sim 1,7 \times 10^{-8} \Omega \cdot m$, próximo ao valor da literatura ($\rho \sim 1,68 \times 10^{-8} \Omega \cdot m$) (GIANCOLI, 2008), em temperaturas mais baixas temos a descontinuidade discutida ao início da seção. Entretanto, apesar disto, observa-se que a amostra apresenta o comportamento esperado para um metal, com dependência linear da resistividade com a temperatura.

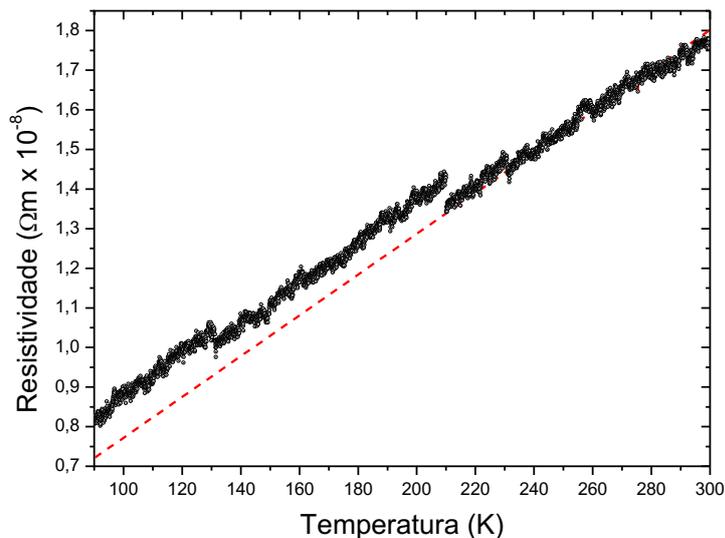


Figura 3.8: Resistividade em função da amostra de cobre em função da temperatura para a região de 90 a 300K com ajuste linear pontilhado em vermelho.

3.4 – Calibração do Termômetro

Com o intuito de calibrar um dos termômetros, é necessário obter a temperatura do termômetro previamente calibrado e a resistência do não calibrado. O ajuste da curva da temperatura em função da resistência, é feito o corte da curva em partes, no caso de nosso termômetro calibrado foi-se dividido em três partes: de 14 a 20,1K, de 20 a 100K e de 100 a 325K. Para este processo realizamos duas medidas, em paralelo com o experimento da seção 3.2, com regiões e taxas distintas, sendo de $T = 90\text{K}$ a $T = 310\text{K}$ com $1\text{K}/\text{min}$ e de $T = 80\text{K}$ a $T = 100\text{K}$ com $0,5\text{K}/\text{min}$.

Para o ajuste da curva é necessário a escolha de um polinômio de complexidade alta o suficiente para que tenhamos uma precisão da ordem de $m\text{K}$, que é o considerado ideal para uma calibração. Por isto, é inviável a realização de um ajuste simples de reta apesar do comportamento ser aparentemente linear. Portanto, nossa escolha foi de utilizar o polinômio

de Tchebychev, este é constante utilizado para ajustes onde há relação entre temperatura e resistência, cujo ajuste é dado pela forma:

$$Temperatura(R_{norm}) = \sum_i A_i \cos(\arccos(R_{norm})) \quad (4.1)$$

$$R_{norm} = \frac{(R - R_I) - (R_S - R)}{R_S - R_I} \quad (4.2)$$

Na primeira equação temos que A_i são os coeficientes de Tchebychev, que precisaremos determinar, e i é a ordem que especifica o grau do ajuste. Na segunda R_{norm} representa a resistência normalizada para $[-1, 1]$, Z a resistência medida, R_I e R_S são, respectivamente, o valor mínimo (inferior) e máximo (superior) da resistência no intervalo de ajuste.

Primeiramente utilizamos os dados correspondente à medida feita com uma rampa de 1K/min. O primeiro passo para a realização do ajuste é com a definição dos limites da resistência, que vão permitir normalizar a resistência:

$$R_I = 25,56\Omega \quad e \quad R_S = 110,29\Omega$$

Definimos que o grau de ordem do ajuste é de 5, realizando os respectivos cálculos obtive os seguintes coeficientes:

$$\begin{aligned} A_0 &= 193,719083 & A_3 &= -0,100151 \\ A_1 &= 104,755487 & A_4 &= 0,015632 \\ A_2 &= 1,205736 & A_5 &= -0,008132 \end{aligned}$$

Repetimos os passos realizados anteriormente e realizamos o ajuste para os dados obtidos com a rampa de 0,5K/min no intervalo. Disto temos os limites:

$$R_I = 21,42\Omega \quad e \quad R_S = 29,93\Omega$$

E os seguintes coeficientes:

$$\begin{aligned} A_0 &= 90,094447 & A_3 &= 0,0007117 \\ A_1 &= 9,984064 & A_4 &= -0,000032185 \\ A_2 &= -0,0004222 & A_5 &= 0,0002578 \end{aligned}$$

Podemos agrupar os dados obtidos anteriormente em forma da tabela 3.4, similar ao que o fabricante fornece ao realizar a compra de um termômetro calibrado.

Tabela 3.4: Coeficientes e limites obtidos através do uso do polinômio de Tchebychev.

80 a 100K		100 a 300K	
R = Resistência		R = Resistência	
$R_I = 21,42\Omega$		$R_I = 25,56\Omega$	
$R_S = 29,93\Omega$		$R_S = 110,29\Omega$	
ORDEM	COEFICIENTE	ORDEM	COEFICIENTE
0	90,094447	0	193,719083
1	0,984064	1	104,755487
2	-0,0004222	2	1,205736
3	0,0007117	3	-0,100151
4	-0,000032185	4	0,015632
5	0,0002578	5	-0,008132

A partir dos dados da tabela 3.4 construímos o gráfico ilustrado na figura 3.9, que permite correlacionar a resistência do termômetro com a temperatura, representando as duas regiões de calibração. Para a região de 80 a 100K nosso ajuste apresentou um erro $RMS = 4,11mK$ com variação máxima de 12mK, próximo ao erro $RMS = 1,00mK$ do fabricante. Para a seguinte região, de 100 a 30K, nosso ajuste apresentou um erro $RMS = 9,37mK$ com variação máxima de 30mK, também próximo ao erro $RMS = 4,00mK$ do fabricante.

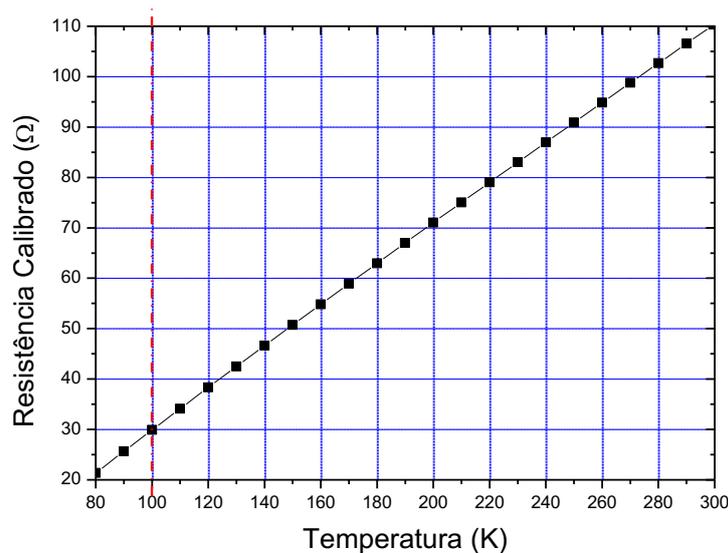


Figura 3.9: Linha em vermelho separando as regiões da calibração do termômetro.

4 – Conclusão e Perspectivas

A realização deste trabalho permitiu o desenvolvimento de um experimento automatizado para a realização de medidas de propriedades elétricas de materiais em sistemas criogênicos. Para a automatização foi desenvolvido uma nova versão, em relação a apresentada no trabalho de conclusão de curso, do *software* para facilitar a utilização pelo usuário e garantir a otimização do controle das medidas. Como forma de validar a montagem e o *software* foi realizada a medida da resistividade de uma amostra padrão de cobre, cujos resultados se mostraram compatíveis com a literatura.

Foi realizado o ajuste do PID do controlador de temperatura para obter uma taxa de variação de temperatura estável na região entre 100K e 300K. Isto foi feito com auxílio do *software* desenvolvido, que possui módulos para facilitar a escolha do melhor método de sintonia sem a necessidade de *expertise* do usuário na área. Disto foram definidas três regiões com distintos valores para PID e saída manual da potência, que foram capazes de realizar rampas de aquecimento com percentual de erro muito baixo (tabela 3.3) para 0,5 K/min, 1,0 K/min e 1,5 K/min. Para taxas maiores concluímos que é necessário o ajuste de um PID distinto.

Também foi realizada a calibração de um termômetro de platina (PT-100), que pode ser utilizado em outros sistemas criogênicos. O procedimento de calibração desse termômetro é importante pois diminui consideravelmente o custo do controle de temperatura, se comparado com a utilização de um termômetro calibrado de fábrica.

Essas etapas concluídas são de grande importância para o laboratório de Supercondutividade, Magnetismo e Materiais Avançados do IMEF/FURG, ao permitir que outras montagens experimentais possam ser desenvolvidas a partir dessa.

Referências

ARAÚJO, J. F. D. F. Construção de um magnetômetro Hall a baixas temperaturas para caracterização de nanopartículas magnéticas, Tese de Doutorado, PUC-Rio, 2013.

ÅSTRÖM, Karl Johan et al. Automatic tuning and adaptation for PID controllers-a survey. In: Adaptive Systems in Control and Signal Processing 1992. 1993. p. 371-376.

ÅSTRÖM ANDT, K. J. Hägglund, PID controllers: theory, design and tuning. 1995.

BALSHAW, N. Practical cryogenics. And introduction to laboratory cryogenics. 1996.

BLOCH, F. F. Bloch, Z. Phys. 52, 555 (1928). Z. Phys., v. 52, p. 555, 1928.

EDWARDS. Bombas de Palhetas Rotativas RV3, RV5, RV8 e RV12. Burgess Hill, 2013, 58p.

GIANCOLI, Douglas C. Physics for scientists and engineers. Pearson Education International, 2008.

GIROTTI, Emerson M.; SANTOS, Ivair A. Medidas de resistividade elétrica DC em sólidos: como efetuá-las corretamente. Química Nova, v. 25, n. 4, p. 639-647, 2002.

HALDER, Marco. Slave. Disponível em: <<https://slave.readthedocs.io/en/develop/>> Acesso em: 07 de novembro de 2018

HANG, Chang C.; ÅSTRÖM, Karl Johan; HO, Weng Khuen. Refinements of the Ziegler–Nichols tuning formula. In: IEE Proceedings D (Control Theory and Applications). IET Digital Library, 1991. p. 111-118.

JIRMANUS, M. N. Introduction to laboratory cryogenics. Janis Research Company, MA, EUA, 1990.

KEYSIGHT. 34401A 6½ Digit Multimeter User's Guide. Colorado, 2014, 246p.

KING, Myke. Process control: a practical approach. John Wiley & Sons, 2016.

KITTEL, Charles. Introdução À Física Do Estado Sólido. Grupo Gen-LTC, 2000.

KUMAR, G. S.; PRASAD, G.; POHL, R. O. Experimental determinations of the Lorenz number. Journal of materials science, v. 28, n. 16, p. 4261-4272, 1993.

LAKESHORE. User's Manual Model 335 Temperatura Controller. Westerville, 2012, 172p.

LORENTZ, H. A. The motion of electrons in metallic bodies I. In: KNAW, proceedings. 1905. p. 438-453.

MEADEN, George T. Electrical resistance of metals. Springer, 2013.

OBRENOVIĆ, Željko; STARČEVIĆ, Dušan; JOVANOVIĆ, Emil. Virtual Instrumentation. Wiley Encyclopedia of Biomedical Engineering, 2006.

SANTOS, L. H. Automação de um sistema de medidas de propriedades elétricas, Trabalho de Conclusão de Curso, FURG, 2015.

SCHRODER, Dieter K. Semiconductor material and device characterization. John Wiley & Sons, 2006.

SCPI CONSORTIUM et al. Standard Commands for Programmable Instruments (SCPI), Volume 1: Syntax and Style. USA, May, 1999.

SHAHROKHI, Mohammad; ZOMORRODI, Alireza. Comparison of PID controller tuning methods. Department of Chemical & Petroleum Engineering Sharif University of Technology, p. 1-2, 2013.

SOMMERFELD, Arnold. Zur elektronentheorie der metalle auf grund der fermischen statistik. Zeitschrift für Physik, v. 47, n. 1-2, p. 1-32, 1928.

SRIVASATAVA, J. P. Elements of solid state physics. PHI Learning Pvt. Ltd., 2014.

ÜNAL, Muhammet et al. Optimization of PID controllers using ant colony and genetic algorithms. Springer, 2012.

VAN DELFT, Dirk; KES, Peter. The discovery of superconductivity. Europhysics News, v. 42, n. 1, p. 21-25, 2011.

VXIplug&play. VPP-2: System Frameworks Specification. 2010, 36p.

WILSON, Alan Herries. The theory of metals. 1954.

ZIEGLER, John G.; NICHOLS, Nathaniel B. Optimum settings for automatic controllers. trans. ASME, v. 64, n. 11, 1942.