



**UNIVERSIDADE FEDERAL DE OURO PRETO**  
**ESCOLA DE MINAS**  
**COLEGIADO DO CURSO DE ENGENHARIA DE**  
**CONTROLE E AUTOMAÇÃO - CECAU**



**RAUL TORRES BERNARDO**

**DESENVOLVIMENTO DE UMA PLATAFORMA PARA APLICAÇÃO DE**  
**TÉCNICAS DE CONTROLE DE TEMPERATURA POR EFEITO PELTIER**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA**  
**DE CONTROLE E AUTOMAÇÃO**

**Ouro Preto, 2015**

RAUL TORRES BERNARDO

DESENVOLVIMENTO DE UMA PLATAFORMA PARA APLICAÇÃO DE TÉCNICAS  
DE CONTROLE DE TEMPERATURA POR EFEITO PELTIER

Monografia apresentada ao Curso de Engenharia de Controle e Automação como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

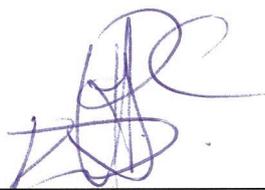
Orientador: Prof. Dr. Agnaldo José da Rocha Reis

Ouro Preto

Escola de Minas – UFOP

Julho/2015

Monografia defendida e aprovada, em 01 de julho de 2015, pela comissão avaliadora constituída pelos professores:



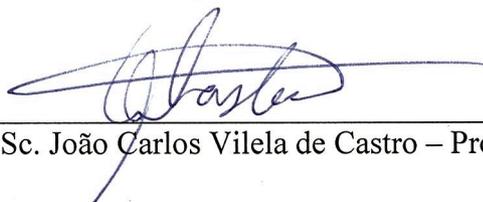
---

Prof. Dr. Agnaldo José da Rocha Reis - Orientador



---

Prof. Dr. Paulo Marcos de Barros Monteiro – Professor Convidado



---

Prof. MSc. João Carlos Vilela de Castro – Professor Convidado

## AGRADECIMENTOS

Agradeço primeiramente a Deus pelas bênçãos concedidas durante toda a minha vida. Agradeço aos meus pais Maria Aparecida e Carlos Fernando por serem os meus maiores exemplos e por sempre me apoiarem nas minhas decisões, mesmo que a consequência delas fosse a distância. Agradeço também a toda a minha família e em especial ao meu irmão Fernando pela amizade e apoio. Agradeço a minha querida Larissa pelo amor e apoio incondicional durante toda a minha jornada acadêmica. Não chegaria até aqui sem seu companheirismo. Agradeço a todos os meus irmãos da República Alambique pelo acolhimento, pelos aprendizados e por todos esses anos de amizade. É muito gratificante ir embora sabendo que sempre terei para onde voltar. Agradeço ao programa Ciência sem Fronteiras e a *HAN University of Applied Sciences* pela oportunidade de viver um ano na Holanda e pelo imenso ganho pessoal e profissional. Agradeço ao programa IAESTE e a *UTN – Facultad Regional Villa María* pela experiência adquirida no estágio e aos seus estudantes pelo acolhimento.

Por fim, agradeço a Universidade Federal de Ouro Preto e seus docentes por todos os ensinamentos e as oportunidades que me foram oferecidas. Em especial, agradeço à Escola de Minas pela formação profissional e por me tornar um engenheiro e ao professor Agnaldo José da Rocha Reis pela orientação e pelo grande apoio neste e em vários outros momentos da minha graduação.

Muito obrigado a todos.

## RESUMO

Explora-se neste trabalho a aplicação de estratégias de controle do tipo *On-Off*, P, PI e PID em uma plataforma para o controle de temperatura em malha fechada cujo elemento final de controle é uma pastilha Peltier. Neste caso, controlando-se a corrente elétrica fornecida à pastilha Peltier, se controla a temperatura de um pequeno bloco de alumínio. Para realizar o controle desse sistema e permitir uma interface com o usuário, foi desenvolvida uma *Graphical User Interface* (GUI) utilizando o *software* LabVIEW. Nessa interface é possível configurar a estratégia de controle desejada e definir a temperatura requerida (i.e., o *set point*). Além disso, também é possível observar graficamente a resposta de temperatura, o desvio e o sinal de controle aplicado ao longo do tempo. A interface física entre o ambiente gráfico e a planta é feita por meio de uma placa microcontroladora Arduino, que é responsável por coletar os dados de temperatura e aplicar o sinal de controle através de um sinal do tipo PWM (*Pulse Width Modulation*). O sistema utilizado neste trabalho permite observar a resposta às diferentes estratégias de controle listadas, bem como armazenar os dados para comparações com valores teóricos. Constata-se que o protótipo desenvolvido se mostra funcional e prático, e que apresenta uma resposta esperada.

Palavras-chave: Efeito Peltier, controle de temperatura, controle de processos, PID, LabVIEW, Arduino.

## **ABSTRACT**

*Is explored in this work the application of On-Off, P, PI and PID control strategies into a platform of closed loop temperature control whose final control element is a Peltier module. In this case, controlling the current applied to the Peltier module, the temperature of a small aluminum block is controlled. To perform the control of this system and to interface with the user, a Graphical User Interface (GUI) was developed using the LabVIEW software. In this interface is possible to configure the control strategy desired and the required temperature (i.e., the setpoint). Furthermore, it is also possible to graphically observe the temperature response, the deviation and the control signal applied over time. The physical interface between the graphical environment and the plant is made by means of an Arduino microcontroller board, which is responsible to collect the temperature data and to apply the control signal through a PWM (Pulse Width Modulation) signal. The system used in this study allows observing the response to the different control strategies listed as well as store the data for comparison with theoretical values. It is noted that the developed prototype shows itself to be functional and practical, and provides an expected response.*

*Keywords: Peltier effect, temperature control, process control, PID, LabVIEW, Arduino.*

## LISTA DE ABREVIACOES

A/D	Analgico/Digital
D/A	Digital/Analgico
DAQ	<i>Data Acquisition System</i>
FT	Funo de Transferncia
Gc	Ganho do controlador
GUI	<i>Graphical User Interface</i>
Kp	Ganho Proporcional
L	Tempo morto
MA	Malha Aberta
MF	Malha Fechada
MV	Varivel Manipulada
P	Proporcional
Pb	Banda Proporcional
PC	Computador pessoal
PD	Proporcional-Derivativo
PI	Proporcional-Integral
PID	Proporcional Integral e Derivativo
PV	Varivel de Processo ou Varivel Controlada
PWM	<i>Pulse-Width Modulation</i>
SP	<i>Setpoint</i>
T	Constante de Tempo
Td	Tempo Derivativo

Ti            Tempo Integral

UART        *Universal Asynchronous Receiver/Transmitter*

USB          *Universal Serial Bus*

## LISTA DE FIGURAS

Figura 2.1- Efeito Peltier .....	19
Figura 2.2 – Módulo Termoelétrico .....	20
Figura 2.3 - Condução de Calor no Módulo Termoelétrico .....	21
Figura 2.4- Arduino Mega 2560 .....	23
Figura 2.5- Exemplo de Programação do Arduino no software LabVIEW (Diagrama de Blocos).....	25
Figura 2.6- Exemplo de Programação do Arduino no software LabVIEW (Painel Frontal) ...	25
Figura 2.7 - Diagrama de blocos genérico.....	27
Figura 2.8 - Diagrama de blocos do controle de temperatura proposto .....	28
Figura 2.9 - Controle On-Off.....	29
Figura 2.10– Histerese .....	30
Figura 2.11 - Controle On-Off com e sem histerese.....	31
Figura 2.12 - Efeito da redução de $P_b$ no comportamento de PV .....	32
Figura 2.13 – Resposta ao degrau unitário com <i>overshoot</i> de 25% .....	35
Figura 2.14 – Curva de resposta ao degrau em MA .....	36
Figura 2.15 – Resposta ao degrau unitário de um sistema de primeira ordem com tempo morto .....	37
Figura 3.1 – Exemplo de montagem.....	39
Figura 3.2 – Estrutura desenvolvida para o projeto.....	39
Figura 3.3 – <i>Driver</i> de corrente bidirecional controlado por tensão .....	40
Figura 3.4 - <i>Driver</i> bidirecional modificado .....	41

Figura 3.5 - <i>Driver</i> bidirecional modificado (segunda versão) .....	42
Figura 3.6 – <i>Driver</i> unidirecional (terceira versão) .....	44
Figura 3.7 - Hardware para medição de temperatura .....	44
Figura 3.8 – Foto do sensor LM35 .....	45
Figura 3.9 – Visão geral da Planta.....	46
Figura 3.10 – Circuito do Driver de Potência Unidirecional.....	46
Figura 3.11 - Blocos de interface com Arduino .....	47
Figura 3.12 - Conexão com o Arduino .....	48
Figura 3.13 - Configurações do Arduino na interface virtual.....	48
Figura 3.14 - Cálculo do valor médio de temperatura.....	49
Figura 3.15 - Display de leitura das variáveis .....	50
Figura 3.16 - Menu de Controles.....	51
Figura 3.17 - <i>Case Structure</i> do modo Manual .....	51
Figura 3.18 - Menu de controle On-Off .....	52
Figura 3.19 - <i>Case Structure</i> do modo <i>On-Off</i> .....	53
Figura 3.20 - Menu de controle PID .....	53
Figura 3.21 - <i>Case Structure</i> do modo PID .....	54
Figura 3.22 - Interface de usuário desenvolvida.....	55
Figura 3.23 - Resposta em Malha Aberta ao degrau de resfriamento de 50% .....	56
Figura 3.24 - Degrau de resfriamento de 50% normalizado.....	57
Figura 3.25 - Simulação com a FT obtida .....	58

Figura 3.26 - Comparação entre o degrau de resfriamento real e simulado .....	58
Figura 3.27 – Resposta ao degrau de 50% no aquecimento .....	60
Figura 4.1 - Resposta ao controle <i>On-Off</i> no resfriamento.....	61
Figura 4.2 - Saída do controlador e erro na resposta ao controlador <i>On-Off</i> no resfriamento .	62
Figura 4.3- Resposta ao controle <i>On-Off</i> com histerese no resfriamento.....	63
Figura 4.4 - Saída do controlador e Erro na resposta ao controle <i>On-Off</i> com histerese no resfriamento .....	63
Figura 4.5 - Configuração do controle P no LabVIEW.....	64
Figura 4.6 - Resposta ao controle Proporcional no resfriamento .....	64
Figura 4.7 - Saída do controlador e Erro na resposta ao controlador P no resfriamento.....	65
Figura 4.8 - Diagrama de blocos do controle P .....	65
Figura 4.9 - Comparação entre as respostas ao controle P, real e simulada.....	66
Figura 4.10 - Configuração do controle PI no LabVIEW .....	66
Figura 4.11 - Resposta ao controle PI no resfriamento .....	67
Figura 4.12 - Saída do controlador e Erro na resposta ao controlador PI no resfriamento .....	67
Figura 4.13 - Diagrama de blocos do controle PI.....	68
Figura 4.14 - Comparação entre as respostas ao controle PI real e simulada .....	68
Figura 4.15 - Configuração do controle PID no LabVIEW .....	69
Figura 4.16 - Resposta ao controle PID no resfriamento .....	69
Figura 4.17 - Saída do controlador e Erro na resposta ao PID no resfriamento.....	70
Figura 4.18 - Diagrama de blocos do controle PID.....	70
Figura 4.19 - Comparação entre as respostas ao controle PID real e simulada.....	71

Figura 4.20 - Resposta ao controle PID no aquecimento .....	72
Figura 4.21 – Resposta ao controle PID bidirecional no aquecimento .....	72
Figura 4.22 - Saída do controlador e Erro na resposta ao PID bidirecional no aquecimento...	73
Figura A1.1 - Circuito do driver unidirecional.....	79
Figura A1.2 - Placa de circuito impresso do driver unidirecional.....	80
Figura A2.1 – Visão geral do diagrama de blocos .....	81
Figura A2.2 – Abertura da conexão com o Arduino .....	81
Figura A2.3 – <i>Case Structure</i> dos Métodos de Controle e Tempo de Execução do <i>Loop</i> .....	82
Figura A2.4 – Pino de saída PWM e Seleção dos LS e LI da Histerese .....	83
Figura A2.5 – Loop de aferição de temperatura.....	83
Figura A2.6 – Sinais mostrados nos gráficos .....	84
Figura A2.7 – Botão ‘Zerar gráficos’ e Botão ‘PARAR’ .....	85
Figura A2.8 – Reinicialização das variáveis e encerramento da conexão com o Arduino.....	86

## LISTA DE TABELAS

Tabela 2.1- Especificações técnicas do módulo TEC1-12706 .....	21
Tabela 2.2 – Método Ziegler-Nichols para sintonia de PID, baseado na resposta ao degrau unitário em MA .....	37
Tabela 3.1 - Ganhos do Controlador para o resfriamento .....	59
Tabela 3.2 - Ganhos do Controlador para o aquecimento .....	60

## SUMÁRIO

1	INTRODUÇÃO .....	15
1.1	Objetivo Geral.....	16
1.2	Objetivos Específicos .....	16
1.3	Justificativa do Trabalho.....	16
1.4	Metodologia Proposta .....	16
2	REVISÃO BIBLIOGRÁFICA.....	18
2.1	Efeito Peltier .....	18
2.1.1	Módulo Termoelétrico.....	19
2.2	Aquisição de Dados e Plataforma Arduino.....	22
2.3	Plataforma LabVIEW .....	23
2.4	Controle de Processos .....	25
2.4.1	Controle Liga-Desliga ( <i>On-Off</i> ) .....	28
2.4.2	Controle PID.....	31
2.4.3	Método Ziegler-Nichols para sintonia de controladores PID.....	35
3	DESENVOLVIMENTO DO PROJETO .....	38
3.1	Projeto da Planta .....	38
3.1.1	<i>Driver</i> para o controle da corrente.....	39
3.1.2	Medição de temperatura .....	44
3.1.3	Planta Final Utilizada .....	45
3.2	Desenvolvimento do Sistema em LabVIEW .....	47
3.2.1	Conexão com o Arduino.....	47
3.2.2	Leitura da temperatura.....	48

3.2.3	Métodos de Controle .....	50
3.2.4	Interface de Usuário.....	54
3.2.5	Tempo de Amostragem e Limitações da Interface .....	55
3.3	Estimação da Função de Transferência e Ganhos do Controlador .....	56
4	RESPOSTA DO SISTEMA E ANÁLISE DOS RESULTADOS .....	61
4.1	Resposta ao controle <i>On-Off</i> .....	61
4.2	Resposta ao controlador P, PI e PID no resfriamento.....	63
4.3	Resposta ao controle PID no aquecimento .....	71
4.4	Resultados prévios com o driver bidirecional.....	72
5	CONCLUSÕES E CONSIDERAÇÕES FINAIS .....	74
	REFERÊNCIAS .....	76
	APÊNDICE 1 .....	79
	APÊNDICE 2 .....	81

## 1 INTRODUÇÃO

A engenharia de controle e automação tem papel fundamental nos processos industriais modernos, tais como nos controles de pressão, temperatura, umidade e viscosidade. O controle de processos físico-químicos se dá através do monitoramento das variáveis de controle e da atuação dos equipamentos da planta (motores, bombas, pistões, aquecedores, etc.), sendo que essa atuação segue uma estratégia de controle previamente definida. O conceito de planta é definido como: “[...] uma parte de um equipamento, eventualmente um conjunto de itens de uma máquina, que funcionam conjuntamente, cuja finalidade é desempenhar uma dada operação.” (OGATA, 1985, P.2). O comportamento e atuação de uma planta definem a qualidade e a eficiência da mesma e só chegarão a um ponto ótimo de operação com a utilização da estratégia de controle correta e dos equipamentos adequados. No meio industrial, o alcance de um ponto ótimo de operação influencia diretamente a produtividade, a qualidade do produto e a lucratividade da empresa. A aplicabilidade da engenharia de controle e automação se mostra clara e, portanto, o estudo aprofundado de estratégias de controle é justificado e necessário para o constante aprimoramento das mesmas. Além das técnicas de controle é necessário também o estudo dos equipamentos que serão utilizados para garantir uma resposta satisfatória ao controle aplicado.

No presente trabalho, estratégias de controle de temperatura foram estudadas, visando aprimorar os conhecimentos a respeito do tema, bem como colocar em prática as metodologias aprendidas ao longo do curso. Uma das principais inspirações para a realização deste trabalho é um projeto proposto pelo autor John Essick nos capítulo 12 e apêndice 1 do livro *Hands-On Introduction to LabVIEW for Scientists and Engineers* (Essick, 2009), que explora o controle bidirecional de temperatura por efeito Peltier, utilizando o software LabVIEW. Como ferramenta principal do estudo proposto neste trabalho, foi utilizado um módulo termoelétrico, também conhecido como pastilha Peltier, que atua sobre um bloco de alumínio aquecendo-o e resfriando-o conforme solicitado pelo sistema de controle. Apresenta-se também uma interface de usuário desenvolvida a partir do software LabVIEW que exibe as informações de operação da planta proposta, bem como é capaz de executar as estratégias de controle estudadas.

## 1.1 Objetivo Geral

Desenvolver um sistema completo para o controle de temperatura que permita o estudo de diferentes estratégias de controle, bem como a aplicação das mesmas por meio de um controle de temperatura bidirecional por efeito Peltier. Para tanto, foram desenvolvidas uma bancada de testes e uma interface virtual capazes de realizar o controle, apresentando os dados de forma clara e permitindo a comparação entre diferentes métodos.

## 1.2 Objetivos Específicos

- Desenvolver uma bancada para controle de temperatura de um bloco de alumínio por efeito Peltier;
- Desenvolver uma interface em LabVIEW que realize a leitura dos valores de temperatura e execute diferentes estratégias de controle;
- Analisar a resposta do sistema frente às estratégias de controle On/Off , P, PI, PID, comparando-se os seus desempenhos.

## 1.3 Justificativa do Trabalho

O estudo prático da teoria de controle se faz necessário durante a formação de um engenheiro de controle e automação. Seguindo esta premissa, este trabalho tem como uma de suas justificativas suprir uma demanda pessoal do autor por atividades e experimentos nas áreas de Teoria de Controle e de Instrumentação Eletrônica.

## 1.4 Metodologia Proposta

Inicialmente é desenvolvida uma bancada de testes para realização do controle de temperatura em malha fechada. Essa bancada consta de um bloco de alumínio, cuja temperatura será controlada, uma pastilha termoelétrica Peltier, que faz o papel de atuador, um sistema para dissipar o calor, um sensor de temperatura, um sistema de aquisição de dados e uma interface com computador.

Mais especificamente, os valores de temperatura do bloco de alumínio são obtidos com um sensor LM35 e entregues a uma interface desenvolvida em LabVIEW. A temperatura do bloco deve ser controlada através da corrente aplicada à pastilha Peltier. Para controlar a corrente fornecida, um *driver* de potência bidirecional é desenvolvido, o qual aplica à pastilha uma corrente variável em módulo e sentido de acordo com um sinal de controle aplicado ao

*driver*. Também é desenvolvido um *driver* capaz de regular a corrente aplicada ao módulo em apenas um sentido, realizando o controle unidirecional da temperatura. A interface física entre a bancada e o software LabVIEW é realizada com o uso de uma placa microcontroladora Arduino, que é responsável por realizar a leitura do sensor de temperatura, converter os dados em sinal digital e aplicar o sinal de controle ao *driver* de potência. Os dados lidos pela plataforma Arduino são interpretados na interface desenvolvida em LabVIEW, onde se encontram as estratégias de controle estudadas. De acordo com a estratégia escolhida, os cálculos são realizados e o programa fornece um sinal de controle para ser entregue à planta por meio do Arduino e do *driver* de potência. De posse do hardware e do software, uma série de experimentos é realizada para verificar a resposta do sistema a duas estratégias básicas de controle, sendo estas o controle liga/desliga com e sem histerese e o controle PID com diferentes ganhos. Em seguida, um comparativo entre dados simulados e dados medidos é realizado.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Efeito Peltier

O sistema de controle de temperatura proposto neste trabalho utiliza como principal ferramenta um módulo termoeétrico, cujo funcionamento baseia-se em um fenômeno físico conhecido como efeito Peltier. Este fenômeno, descoberto pelo físico Jean Charles Athanase Peltier no início do século dezenove, consiste na absorção ou liberação de calor devido à passagem de corrente elétrica de uma substância para outra (junção) (FRADEN, 2004).

O calor absorvido ou produzido pela junção é função da corrente elétrica e depende do sentido da mesma, como é apresentado na equação 2.1.

$$dQ_p = \pm pidt \quad (2.1)$$

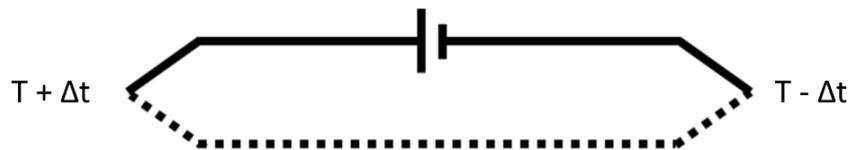
Onde  $i$  é a corrente,  $t$  o tempo,  $p$  um coeficiente medido em Volts que representa as propriedades termoeétricas do material e  $Q_p$  é o calor produzido ou absorvido por efeito Peltier.

O efeito Peltier pode ser utilizado de duas formas: para produzir ou absorver calor. A produção ou absorção do calor depende do sentido da corrente aplicada. Isso o torna muito útil para diversas aplicações, principalmente quando um controle térmico preciso é necessário (FRADEN, 2004).

O conceito de Efeito Peltier pode ser apresentado da seguinte forma:

“[...] Quando um circuito contendo duas junções, inicialmente à mesma temperatura, é percorrido por uma corrente, em decorrência da conexão de uma fonte externa, ocorre o aquecimento de uma junção e o resfriamento da outra [...] O circuito efetivamente bombeia o calor de uma junção para a outra.” (BEGA et al., 2006, P.223)

A figura 2.1 ilustra o efeito Peltier, onde uma junção é aquecida e a outra resfriada devido à adição de uma fonte de tensão.



**Figura 2.1- Efeito Peltier**

É importante diferenciar o efeito Peltier do efeito Joule, pois, apesar de ambos tratarem do calor produzido por um material condutor devido à passagem de corrente elétrica, possuem características distintas. O efeito Joule corresponde ao calor produzido pela passagem de corrente elétrica, em qualquer direção, por um material condutor com resistência elétrica finita, onde a potência térmica liberada é proporcional a sua resistência e ao quadrado da corrente aplicada. Esta relação é apresentada na equação 2.2. (FRADEN, 2004).

$$P = R \times i^2 \quad (2.2)$$

Já no efeito Peltier, o calor produzido ou absorvido depende linearmente da magnitude da corrente, equação 2.1. Esta diferença de efeitos é levada em consideração na utilização dos Módulos Termoelétricos, onde os fabricantes recomendam que os mesmos sejam utilizados com cerca de 70% dos valores máximos de corrente e tensão declarados nas especificações do produto. Esta recomendação visa uma maior eficiência no funcionamento dos módulos, pois, a partir destes valores de operação, a geração de calor por efeito Joule passa a influenciar de forma significativa, diminuindo a absorção de calor pelo módulo e gerando resultados adversos na solda de conexão dos semicondutores que compõem o módulo, podendo danificar o mesmo. (CUI Inc, 2015)

### 2.1.1 Módulo Termoelétrico

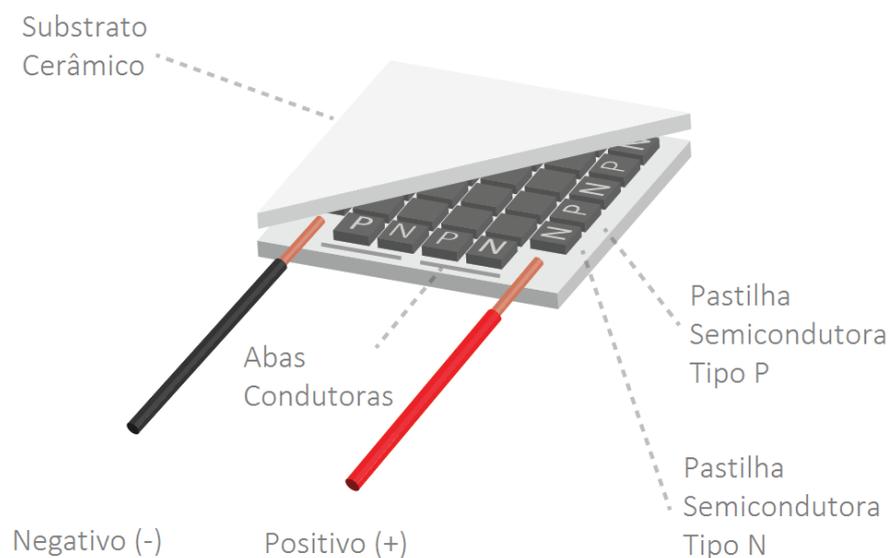
Um sistema de controle, da forma proposta neste trabalho, faz uso de um atuador para forçar uma variação de temperatura. Este atuador consiste de um equipamento capaz de converter energia elétrica em fluxo de calor, aquecendo ou resfriando o objeto proposto, no caso um bloco de alumínio.

O atuador utilizado é um Módulo Termoelétrico ou Módulo Peltier, e consiste de em componente eletrônico baseado em semicondutores, o qual atua como uma pequena bomba de calor. Ao aplicar-se uma corrente contínua no módulo, calor é movido através dele de uma superfície para a outra. Então uma superfície do módulo é resfriada enquanto a superfície

oposta é aquecida. A temperatura é controlada pelo fluxo de calor produzido pelo módulo, que por sua vez é controlado pela Corrente Contínua aplicada a este módulo. A troca de calor é proporcional ao módulo da corrente aplicada e o sentido dessa troca dependerá do sentido da corrente. (ESSICK, 2009)

A tecnologia das pastilhas Peltier tem avançado nos últimos anos devido ao advento de materiais semicondutores adequadamente dopados contendo uma liga de bismuto e telúrio (MCKINNON, 2015). A dopagem eletrônica consiste “[...] num procedimento de adição de impurezas químicas a um elemento semiconductor para transformá-lo num elemento mais condutor, porém, de forma controlada” (PEREIRA, 2015). Assim são formados dois tipos de elementos. De forma simplificada, quando a dopagem favorece que cargas negativas sejam conduzidas (inserindo elétrons) o material é do tipo N, quando dopagem cria uma “lacuna” de elétrons, fazendo com que “cargas positivas” (ausências de elétrons) sejam propagadas, este elemento é do tipo P. (WESTE, et al, 2011). Estes pares de elementos P/N formam junções passíveis de sofrerem o Efeito Peltier. No módulo termoelétrico, os pares de pastilhas P/N são montados de forma que elas estejam eletricamente conectadas em série e termicamente em paralelo, somando o efeito termoelétrico de todas as junções. (CUI Inc, 2015)

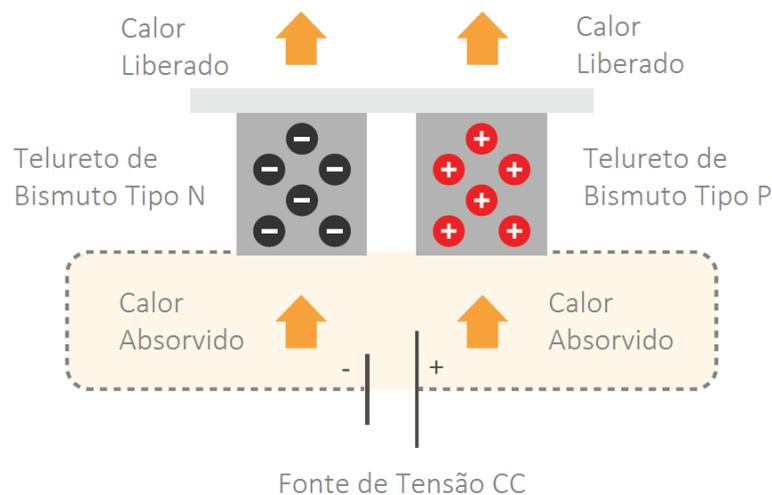
A figura 2.2 ilustra um módulo termoelétrico evidenciando os pares de elementos do tipo P e N.



**Figura 2.2 – Módulo Termoelétrico**

**Fonte: Adaptado de CUI Inc, 2015**

Placas de cerâmica metalizada formam os substratos e a estrutura física, onde são montadas as pastilhas semicondutoras e a lâmina de material condutor que as conecta. O fornecimento de uma tensão contínua ao módulo faz com que os portadores de cargas (positivas e negativas) nas pastilhas semicondutoras absorvam calor de uma superfície de substrato e o liberem na superfície oposta, conduzindo calor de uma superfície para a outra, como ilustrado na figura 2.3. A inversão na polaridade da tensão fornecida causa uma inversão no sentido de condução do calor. (CUI Inc, 2015).



**Figura 2.3 - Condução de Calor no Módulo Termoeletrico**

**Fonte: Adaptado de CUI Inc, 2015**

O módulo utilizado para realização dos experimentos é o TEC1-12706, cujas especificações são resumidas na tabela 2.1.

**Tabela 2.1- Especificações técnicas do módulo TEC1-12706**

Hot Side Temperature (°C)	25	50
Qmax (Watts)	50	57
Delta Tmax (°C)	66	75
I <sub>max</sub> (Amps)	6.4	6.4
V <sub>max</sub> (Volts) 14.4 16.4	14.4	16.4
Module Resistance (Ohms)	1.98	2.30

**Fonte: Hebei, 2015**

Os valores das especificações do módulo são valores teóricos que auxiliam na escolha do produto adquirido. A máxima diferença de temperatura (Delta Tmax) corresponde à diferença de temperatura entre as superfícies do módulo quando a absorção de calor é zero, já a máxima

absorção de calor ( $Q_{max}$ ) ocorre quando a diferença de temperatura entre os lados da pastilha Peltier é zero. (CUI Inc, 2015).

## 2.2 Aquisição de Dados e Plataforma Arduino

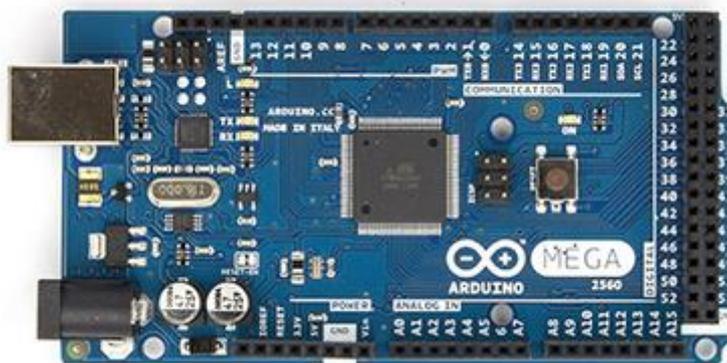
Para a realização do projeto se faz necessário um sistema de aquisição de dados para fazer a interface entre a planta e computador. Existem no mercado diversos modelos de DAQ (*Data Acquisition Systems*) com diferentes aplicabilidades e preços. Segundo a *National Instruments*,

“A aquisição de dados (DAQ) é o processo de medição de um fenômeno elétrico ou físico, como tensão, corrente, temperatura, pressão ou som, com o uso de um computador. Um sistema DAQ é formado por sensores, hardware de aquisição e medição de dados e um computador com software programável. Em comparação com os sistemas tradicionais de medição, os sistemas DAQ baseados em PC exploram a capacidade de processamento, produtividade, sistemas de visualização e recursos de conectividades dos computadores padrão da indústria. Com isso, temos uma solução de medição mais poderosa, flexível e de melhor custo-benefício.” (NATIONAL INSTRUMENTS, 2014 a)

“A plataforma Arduino é uma plataforma computacional de código aberto, baseada em uma simples placa micro controladora, e um ambiente para desenvolvimento de softwares para esta placa” (ARDUINO, 2014 a). Neste trabalho esta plataforma foi escolhida para atuar como sistema de aquisição de dados devido ao seu baixo custo e disponibilidade nos laboratórios da Escola de Minas.

A sua viabilidade foi testada durante a execução do projeto, e como esperado, o seu funcionamento foi satisfatório. O modelo escolhido foi o Arduino Mega 2560,

“O Arduino Mega 2560 é uma placa micro controladora baseada no micro controlador ATmega2560. Ela possui 54 pinos de entrada/saída digitais (dos quais 15 podem ser utilizados como saída PWM), 16 entradas analógicas, 4 UARTs (porta para comunicação serial), um cristal oscilador de 16MHz, conexão USB, e um botão de reset. Ela contém todo o necessário para dar suporte ao micro controlador, basta conecta-lo ao computador por um cabo USB para começar a utiliza-lo.” (ARDUINO, 2014 b)



**Figura 2.4- Arduino Mega 2560**

**Fonte: ARDUINO, 2014 b**

No presente projeto, uma das funções do Arduino é receber os dados analógicos do sensor de temperatura e converte-los em valores digitais para serem enviados ao software que executará as ações de controle. Para isso, o Arduino faz uso de um conversor A/D de 10 bits com medição de 0 a 5V. Além disso, é função do Arduino fornecer à planta o sinal de controle, por meio das saídas PWM que possuem uma resolução padrão de 8 bits. Cada porta de saída é responsável por controlar a corrente fornecida ao módulo Peltier em uma direção, por meio do *driver* de potência explicado na seção 3.1.1.

A conexão com o computador é feita via USB por meio de comunicação serial, sendo possível a comunicação com diversos softwares de programação, como a própria plataforma de desenvolvimento do Arduino e o software de programação LabVIEW.

### **2.3 Plataforma LabVIEW**

Para a execução do projeto se faz necessário escolher qual software será utilizado para realizar o controle e a interface com o usuário. Esse software deve ser de simples manuseio e capaz de realizar os cálculos inerentes às estratégias de controle adotadas. Existem diversas possibilidades de softwares que podem ser utilizados, um deles é o próprio ambiente de programação do Arduino. No entanto, deve-se buscar um software que atenda da forma satisfatória as demandas do projeto, como por exemplo, a criação de uma interface gráfica.

O software LabVIEW da *National Instruments* é um ambiente de programação gráfico que permite o desenvolvimento de uma GUI de forma simples e intuitiva, além de facilitar a comunicação com a plataforma Arduino devido a um pacote de expansão (*Toolbox*)

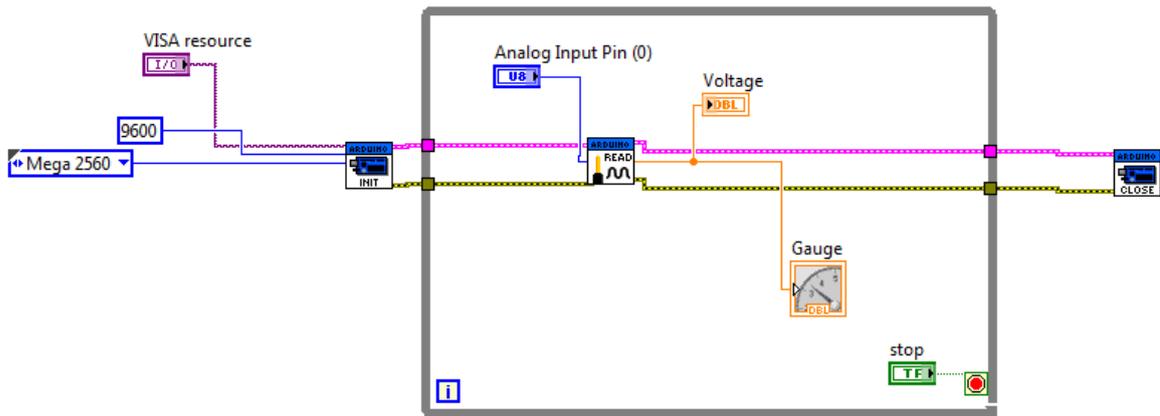
específico para isso. O LabVIEW é um *software* muito versátil, de fácil integração com *hardwares* e de ampla funcionalidade como descrito pelo fabricante,

“O LabVIEW é uma plataforma de programação gráfica que ajuda os engenheiros do projeto ao teste de sistemas de pequeno e grande porte. Ele oferece integração sem precedentes com *software*, IP e *hardware* já implementados e as vantagens das tecnologias mais recentes dos computadores. O LabVIEW oferece ferramentas para resolver os problemas — e a capacidade para inovação futura — com maior rapidez e eficiência.” (NATIONAL INSTRUMENTS, 2014 b).

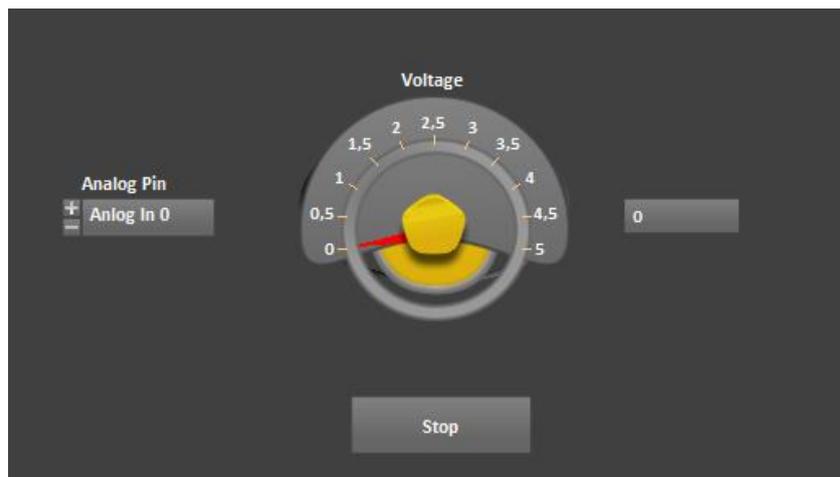
A sugestão para a utilização do LabVIEW parte do livro *Hands-on Introduction to LabVIEW for Scientists and Engineers* (ESSICK, 2009). No entanto foi adotado o Arduino como sistema de aquisição de dados ao invés de um sistema DAQ da *National Instruments* como o utilizado pelo autor do livro.

Um programa desenvolvido em LabVIEW possui duas partes principais, o Diagrama de Blocos, que contém o algoritmo do programa, e o Painel Frontal, onde a interface com o usuário acontece. O Diagrama de Blocos é onde algoritmo de programação é desenvolvido, incluído os blocos relativos à conexão com os periféricos utilizados, blocos de entrada e saída de dados, bem como cálculos e lógicas a serem realizados durante a execução do programa. No painel frontal estão presentes elementos gráficos, como botões, caixas de texto e chaves de seleção, que permitem a interação do usuário com o programa, além de apresentar gráficos e luzes contendo informações e dados numéricos durante a execução.

As figuras 2.5 e 2.6 ilustram o diagrama de blocos e o painel frontal de um exemplo simples de um programa em LabVIEW com comunicação com o Arduino. A primeira mostra o “código” do programa desenvolvido através de blocos de programação e a segunda mostra a interface gráfica desenvolvida. Este programa, que pertence à biblioteca de exemplos do *toolkit LabVIEW Interface for Arduino*, permite ao usuário escolher uma porta analógica do Arduino para ser lida e mostra o valor de tensão em um indicador gráfico.



**Figura 2.5- Exemplo de Programação do Arduino no software LabVIEW (Diagrama de Blocos)**



**Figura 2.6- Exemplo de Programação do Arduino no software LabVIEW (Painel Frontal)**

## 2.4 Controle de Processos

Após o desenvolvimento da planta e estabelecimento da comunicação com o computador, a próxima etapa é a aplicação de uma estratégia de controle para alcançar a temperatura desejada. Porém, antes do desenvolvimento do sistema, se faz necessário compreender alguns conceitos do controle de processos e o funcionamento da planta. Este capítulo apresenta alguns termos utilizados na automação e controle de processos e sua relação com o presente trabalho.

**Processo:** “Qualquer operação ou sequência de operações, envolvendo uma mudança de estado, de composição, de dimensão ou outras propriedades que possam ser definidas relativamente a um padrão” (OLIVEIRA, 1999, p. 9), ou um “conjunto de atividades ou passos que objetivam atingir uma meta. Utilizado para criar, inventar, projetar, transformar, produzir, controlar, manter e usar produtos ou sistemas” (BAYER & ARAÚJO, 2012, p.15). No caso do projeto proposto neste trabalho, o processo é a alteração da temperatura do bloco de alumínio, através da atuação do módulo termoeletrico por meio do *driver* de potência.

**Sistema:** É todo o conjunto de componentes e ferramentas utilizadas para realizar o objetivo proposto (OLIVEIRA, 1999). Para o controle de temperatura proposto, o sistema, engloba o computador, o programa desenvolvido em LabVIEW, a plataforma Arduino, o sensor de temperatura, o *driver* de potência, o módulo termoeletrico, o dissipador de calor e a fonte de alimentação.

**Variável de Processo ou Variável Controlada (PV):** É a variável medida sobre a qual o controle irá atuar para garantir um comportamento determinado do processo (BAYER & ARAÚJO, 2012). No caso deste trabalho, a Variável Controlada é a temperatura do bloco de alumínio, que deve ser mantida dentro de uma faixa de valores desejada.

**Variável Manipulada (MV):** É uma a variável do processo capaz de influenciar a Variável Controlada, é através dela que o controle atua no intuito de manter a Variável Controlada no valor desejado (OLIVEIRA, 1999) (BAYER & ARAÚJO, 2012). A Variável Manipulada do sistema desenvolvido neste trabalho é a corrente fornecida ao módulo termoeletrico pelo *driver* de potência. Esta corrente influencia diretamente na troca de calor realizada pelo módulo, alterando a temperatura do bloco. A corrente, por sua vez, é controlada através da tensão aplicada pelo Arduino ao *driver*.

**Valor desejado ou Setpoint (SP):** “Sinal de entrada que estabelece o valor desejado da variável controlada.” (Bayer e Araújo, 2012, p.16). É um valor expresso na mesma unidade que a Variável Controlada e serve de referência para controlar a variável de processo. Para o controle de temperatura proposto, o *setpoint* é a temperatura final desejada para o bloco de alumínio em graus Celsius.

**Distúrbio:** É um sinal externo que pode afetar o valor da variável controlada, por exemplo, a variação de temperatura ambiente que pode influenciar na temperatura do bloco de alumínio.

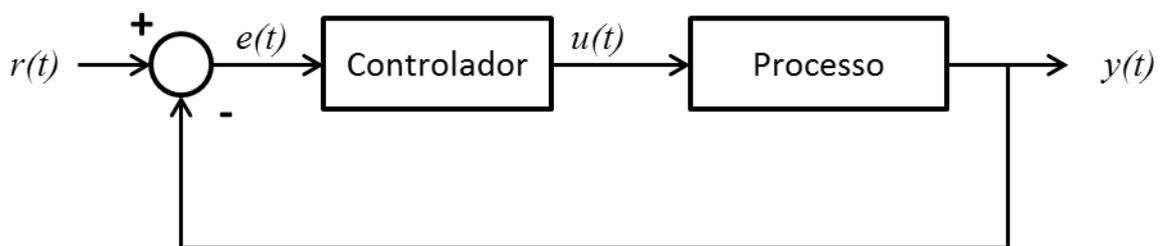
**Desvio (Erro):** Quando, em uma estratégia de controle, o valor desejado e o valor da variável controlada são comparados e apresentam uma diferença, esta diferença é chamada de Erro. É o quão distante do valor desejado a variável controlada se encontra em um determinado momento. O erro remanescente após a estabilização do sistema é chamado Erro em Regime Permanente.

**Ganho:** “[...] é calculado a partir do quociente entre a taxa de mudança na saída e a taxa de mudança na entrada” (BAYER & ARAÚJO, 2012, p.31). No presente trabalho é a relação entre a variação da temperatura no bloco de alumínio dado uma variação de tensão na saída do Arduino.

**Sobressinal, Pico de Resposta ou *Overshoot*:** “É o valor que a variável controlada ultrapassa o *setpoint* por ocasião da primeira oscilação. Normalmente é expresso em termos percentuais em relação ao *setpoint*” (BAYER & ARAÚJO, 2012, p.28).

**Tempo morto, atraso ou *delay*:** É o tempo que o processo leva para responder a uma alteração na variável manipulada. No caso do sistema proposto, é o tempo para que uma variação de temperatura seja detectada após o acionamento do módulo termoeletrônico.

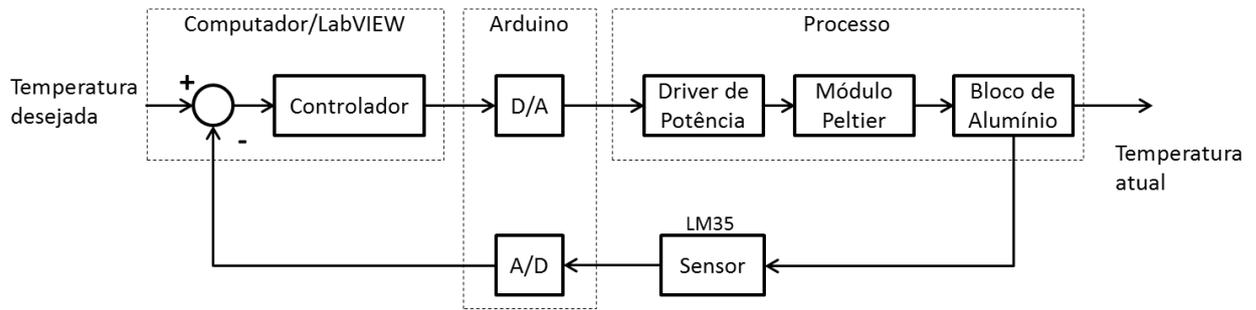
A figura 2.7 representa um diagrama de blocos de um controle em malha-fechada genérico, onde  $r(t)$  é o valor de referência (*setpoint*),  $y(t)$  a variável controlada,  $e(t)$  o desvio entre a variável controlada e o *setpoint* e  $u(t)$  a saída do controlador para o processo.



**Figura 2.7 - Diagrama de blocos genérico**

A figura 2.8 ilustra o sistema proposto neste trabalho através de um diagrama de controle em malha-fechada. Nele a temperatura atual é realimentada ao sistema de forma que exista uma comparação entre a temperatura atual e a desejada. O controlador, presente dentro do programa desenvolvido em LabVIEW, utiliza essa diferença (erro) para calcular o acionamento do módulo termoeletrônico. O Arduino atua como um conversor A/D e D/A, realizando a interface entre a planta e o computador. De acordo com o sinal enviado pelo

controlador o *driver* de potência entrega a corrente elétrica ao módulo termoeletrico que aquece ou resfria o bloco de alumínio.



**Figura 2.8 - Diagrama de blocos do controle de temperatura proposto**

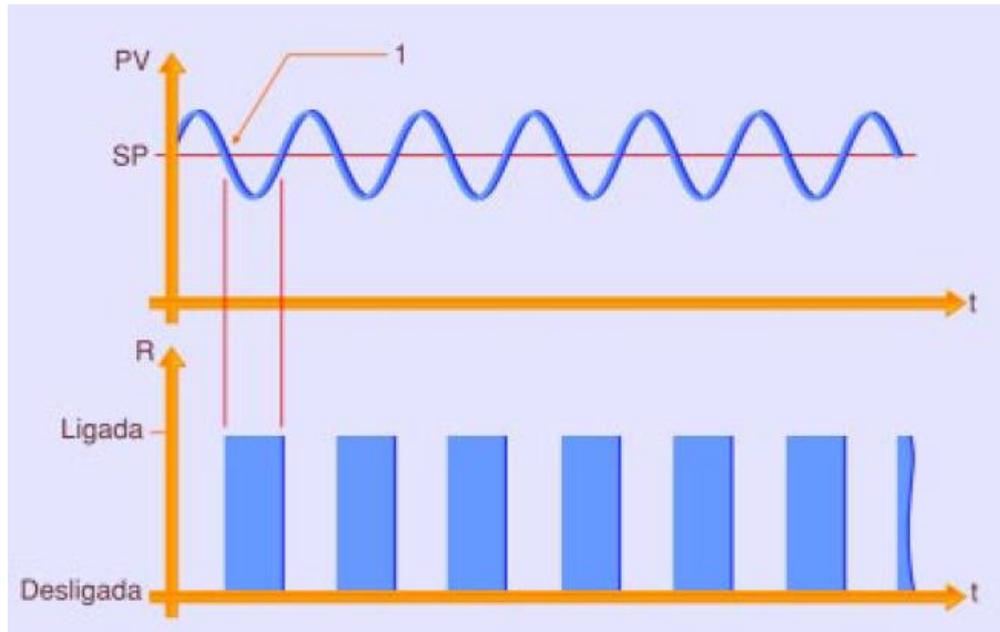
A relação entre o erro e a corrente fornecida ao módulo depende da estratégia de controle adotada, bem como das constantes utilizadas no controlador. Foram analisadas durante a execução deste projeto duas estratégias básicas de controle, o controle On-Off e o PID, com diferentes configurações.

#### 2.4.1 Controle Liga-Desliga (*On-Off*)

O controle *On-Off* é uma das formas mais simples de controle, tendo geralmente um custo mais baixo, sendo uma das formas de controle mais utilizadas. Como o próprio nome sugere, este controle possui apenas duas saídas, ligado ou desligado, ou seja, a variável manipulada é utilizada sempre em um valor máximo ou mínimo, não permitindo valores intermediários (MATIAS, 2002). Neste método o sistema compara a variável de processo com o *setpoint* e alterna a variável manipulada entre ligada e desligada, dependendo do resultado desta comparação, conforme apresentado na equação 2.3, onde  $u(t)$  é a saída do controlador que pode assumir os valores  $U_1$  ou  $U_2$  dependendo do sinal de erro  $e(t)$ .

$$u(t) = \begin{cases} U_1 & \text{se } e(t) > 0 \\ U_2 & \text{se } e(t) < 0 \end{cases} \quad (2.3)$$

Na figura 2.9 observa-se um exemplo de controle *On-Off*, onde um aquecedor por resistência elétrica é utilizado para controlar uma temperatura. Neste exemplo PV representa a variável de processo (temperatura), SP o valor desejado de temperatura e R o estado da resistência (ligada ou desligada). Quando a variável de processo tem valor inferior ao *setpoint*, a resistência é acionada, a partir do momento em que a temperatura ultrapassa o valor desejado, a resistência é desligada.



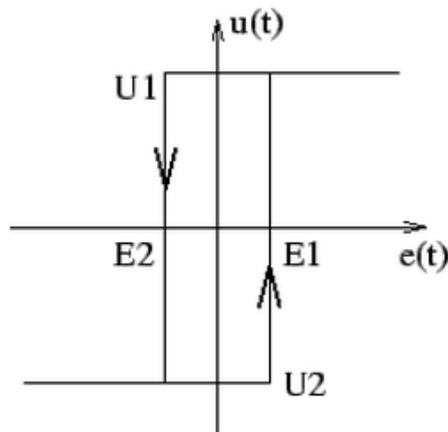
**Figura 2.9 - Controle On-Off**

**Fonte: MATIAS, 2002**

Este exemplo demonstra o funcionamento do controle *On-Off* e também evidencia as suas desvantagens. Pela figura percebe-se a constante oscilação da variável de processo e do acionamento da resistência, esta oscilação é devido à resistência estar totalmente ligada ou desligada. Como ela não pode assumir um valor intermediário de funcionamento a temperatura vai sempre oscilar em torno do *setpoint*.

Em uma aplicação prática, a oscilação do acionamento do atuador entre máximo e mínimo, ou ligado e desligado pode acarretar em danos ao sistema, desde ruídos a falhas mecânicas devido à alta frequência de acionamento.

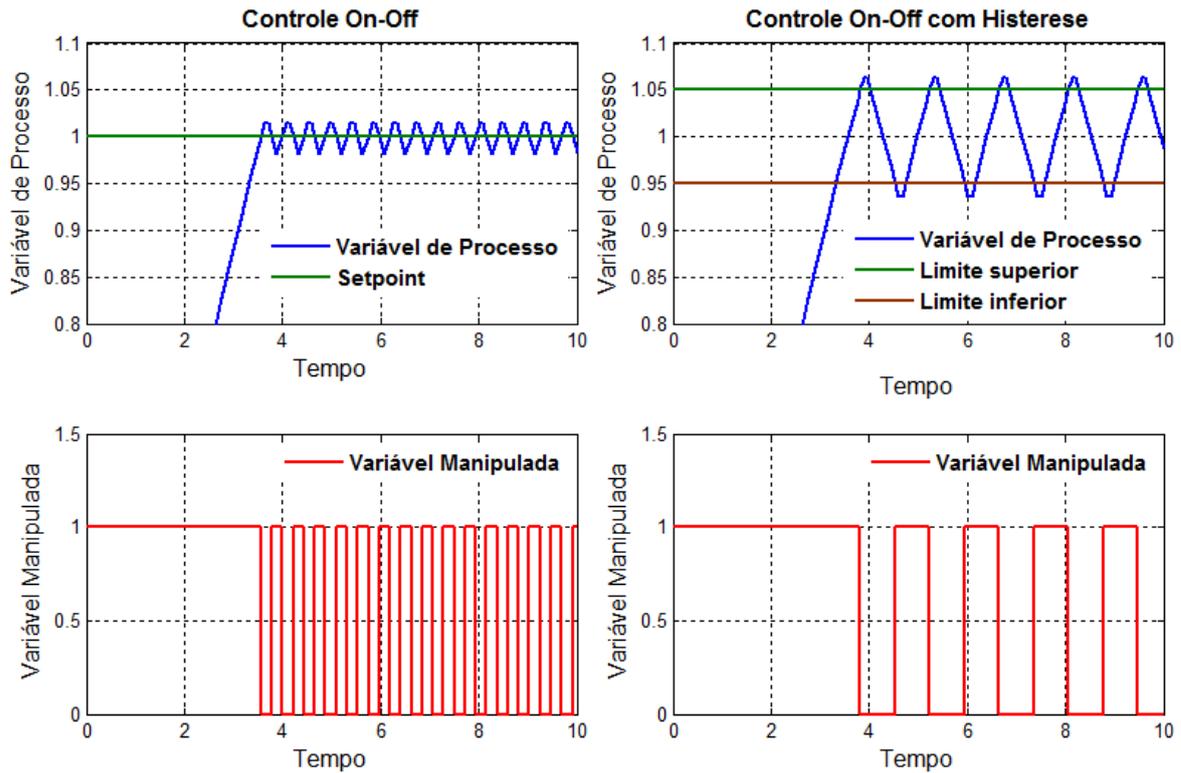
Uma adaptação ao controle *On-Off* é realizada por meio da adição de histerese ao processo. Neste caso o controlador compara a Variável de Processo com dois valores distintos, chamados de limite superior (E1) e inferior (E2) (figura 2.10), de forma que o sinal de saída só é alterado quando o sinal de entrada (erro) é inferior ao limite inferior ou superior ao limite superior. A diferença entre os limites superior e inferior é chamada de histerese, e quando o sinal de entrada (erro) encontra-se neste intervalo a saída não é alterada. A histerese minimiza a oscilação do acionamento do atuador, pois estende o período no qual ele permanece ligado e desligado, porém aumenta a amplitude da oscilação, gerando um erro maior.



**Figura 2.10– Histerese**

**Fonte: Silva, 2015**

Na figura 2.11 pode-se observar uma comparação entre o método com e sem histerese, em uma situação hipotética. Os gráficos mostram a resposta de um sistema de primeira ordem com atraso de 0,5s a um controle *On-Off* com e sem histerese. Comparando as imagens da figura, percebe-se uma diminuição significativa na frequência de acionamento da variável manipulada no controle com histerese, tendo como consequência um erro oscilando entre +5 e -5%. A frequência e amplitude da oscilação do sistema em regime permanente dependem do intervalo da histerese, e este intervalo deve ser determinado levando em conta a precisão, os níveis de ruído e vida útil desejados. (SILVA, 2015)



**Figura 2.11 - Controle On-Off com e sem histerese**

No sistema proposto neste trabalho o controle *On-Off* é aplicado utilizando como saídas possíveis para o controlador um valor máximo de corrente positiva (máximo aquecimento) e um valor máximo de corrente negativa (máximo resfriamento), no caso do controle bidirecional. Para o controle em uma direção (aquecimento ou resfriamento) as saídas possíveis são: valor máximo ou desligado.

## 2.4.2 Controle PID

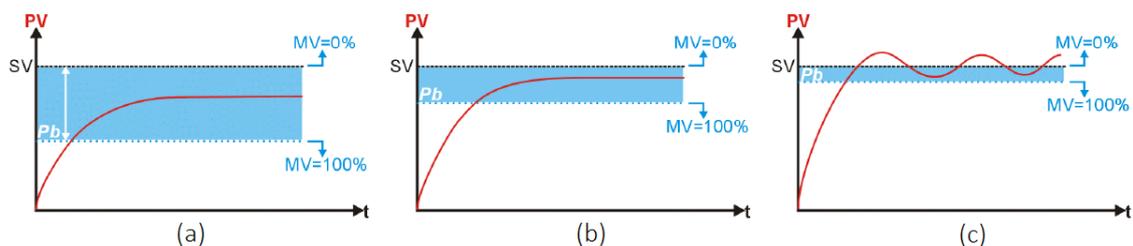
Para um controle mais preciso, se faz necessária a utilização de uma estratégia de controle diferente, que permita valores de saída contínuos, capaz de alcançar uma estabilização em regime permanente. Neste intuito será aplicado o controlador PID.

### 2.4.2.1 Controle Proporcional

Segundo Oliveira (1999) o controle *On-Off* realiza movimentos bruscos de liga e desliga provocando uma oscilação no resultado de controle. Para evitar tal tipo de movimento foi desenvolvido um tipo de ação no qual a ação corretiva produzida por este mecanismo é proporcional ao valor do desvio. Tal ação denominou-se ação proporcional.

No controle proporcional o valor da MV é proporcional ao desvio (erro), então, quando o erro é nulo a MV é igual a zero, e à medida que o erro aumenta, a o valor da MV aumenta até atingir 100%. O valor de erro que provoca ao controlador uma saída de 100% (MV = 100%) define a Banda Proporcional (Pb). Uma Pb alta implica na saída MV assumir valores altos apenas quando o erro também for alto, já uma Pb baixa significa uma saída MV alta mesmo para pequenos erros, ou seja, quanto menor for o valor de Pb mais forte é a ação proporcional de controle e mais sensível é o controlador (NOVUS PRODUTOS ELETRÔNICOS LTDA, 2003).

O efeito da redução de Pb na Variável de Processo pode ser notado na figura 2.12. Na figura 2.12.a o sistema atinge a estabilidade, porém possui um grande erro em regime permanente, mantendo a PV bem abaixo do *setpoint*. A redução da Pb (figura 2.9.b) faz com que a estabilização do sistema ocorra mais próxima do *setpoint*, resultando em um erro em regime permanente menor. No entanto, uma redução excessiva da Pb (figura 2.9.c) pode resultar em uma oscilação levando o sistema a instabilidade.



**Figura 2.12 - Efeito da redução de Pb no comportamento de PV**

**Fonte: NOVUS PRODUTOS ELETRÔNICOS LTDA, 2003**

É importante mencionar que os mecanismos da ação proporcional, eliminam as oscilações no processo provocadas pelo controle liga-desliga, porém o controle proporcional não consegue eliminar o erro em regime permanente, visto que um erro nulo resulta em uma saída proporcional nula fazendo com que um erro surja novamente.

A função de transferência do controlador  $G_c(s)$  é geralmente expressa por uma constante  $K_p$ .

$$\frac{U(s)}{E(s)} = G_c(s) = K_p \quad (2.4)$$

E o sinal de saída do controlador no domínio do tempo  $u(t)$ , neste caso, é:

$$u(t) = K_p e(t) \quad (2.5)$$

### 2.4.2.2 Controle Proporcional Integral

A ação Integral consiste na adição, na saída do controlador  $u(t)$ , de um valor que varia em uma razão proporcional ao sinal de erro  $e(t)$ . Conforme a equação 2.6.

$$\frac{du(t)}{dt} = K_i e(t) \quad (2.6)$$

Ou,

$$u(t) = K_i \int_0^t e(t) dt \quad (2.7)$$

Onde  $K_i$  é a constante que representa o ganho integral do controlador (OGATA, 1999).

A ação integral irá atuar no sistema enquanto houver diferença entre a Variável Controlada e o *setpoint*, integrando ao longo do tempo o sinal de correção. Desta forma, a ação integral atua de forma lenta até eliminar o erro em regime permanente (OLIVEIRA, 1999).

No entanto a ação integral pode piorar a estabilidade do sistema se aplicada de forma isolada, por esse motivo ela é quase sempre utilizada em conjunto com a ação proporcional, apresentando a equação de saída do controlador no domínio do tempo descrita na equação 2.8.

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2.8)$$

E a função de transferência representada pela equação 2.9.

$$\frac{U(s)}{E(s)} = G_C(s) = K_p \left( 1 + \frac{1}{T_i s} \right) \quad (2.9)$$

Onde  $K_p$  é o ganho proporcional e  $T_i$  é denominado tempo integral. Estes valores são ajustáveis de forma que  $T_i$  altera a ação de controle integral e  $K_p$  afeta as partes proporcional e integral da ação de controle (OGATA,1999).

### 2.4.2.3 Controle Proporcional Derivativo

A ação proporcional derivativa atua no controlador de acordo com a variação do sinal de Erro, através de um sinal proporcional à derivada do erro, de acordo com a equação 2.10.

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (2.10)$$

Com a função de transferência representada na equação 2.11.

$$\frac{U(s)}{E(s)} = G_C(s) = K_p (1 + T_d s) \quad (2.11)$$

Onde  $T_d$  é denominado tempo derivativo. A ação derivativa faz com que a magnitude do sinal de controle seja proporcional à velocidade de variação do sinal de erro, fazendo com que o controlador reaja de forma mais rápida. O tempo derivativo  $T_d$  é o tempo no qual a ação derivativa antecipa a ação proporcional, fazendo com que a ação derivativa tenha um caráter de previsão (OGATA, 1999).

A ação derivativa atua somente sobre períodos transitórios, por esse motivo nunca é utilizada individualmente. Além disso, por trabalhar em cima da variação do sinal de erro, a ação derivativa pode ser prejudicial ao sistema tornando-o mais sensível a sinais de alta frequência como ruídos, por exemplo, provocando um efeito de saturação no atuador.

#### 2.4.2.4 Controle Proporcional Integral e Derivativo

O controlador Proporcional Integral e Derivativo (PID), como o nome sugere, consiste na combinação das ações proporcional, integral e derivativa. Esta combinação possui as vantagens das três ações individuais, representada na equação 2.12.

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.12)$$

Com a função e transferência para o controlador apresentada na equação 2.13.

$$\frac{U(s)}{E(s)} = G_C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \quad (2.13)$$

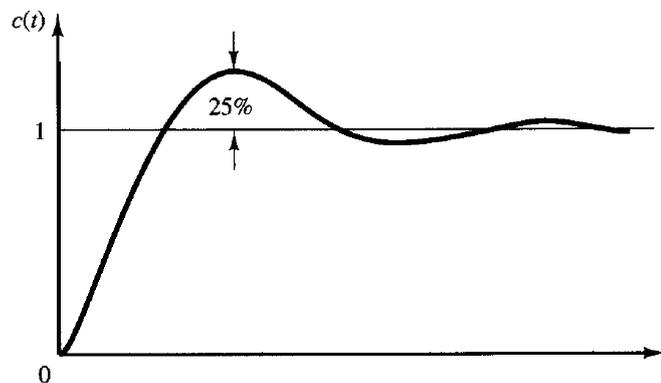
É uma das metodologias de controle mais populares na indústria e, segundo Dorf e Bishop (2001),

“[...] A popularidade dos controladores PID pode ser atribuída parcialmente ao seu desempenho robusto sobre uma grande faixa de condições operacionais e em parte à sua simplicidade funcional, que permite aos engenheiros operá-los de uma maneira simples e correta. Para implementar um controlador destes, três parâmetros devem ser determinados para um dado processo: ganho proporcional, ganho integral e ganho derivativo.” (DORF; BISHOP, 2001, p.289)

### 2.4.3 Método Ziegler-Nichols para sintonia de controladores PID

Na década de 1940, Ziegler e Nichols desenvolveram dois métodos empíricos para obter os ganhos de um controlador PID, o método da resposta em malha fechada e o método do degrau em malha aberta. Este último, que será utilizado neste projeto, é uma técnica muito popular para sintonia de controladores PID e necessita, basicamente, que os dados de resposta da planta a um degrau sejam gravados (BENNETT, et al. 2015).

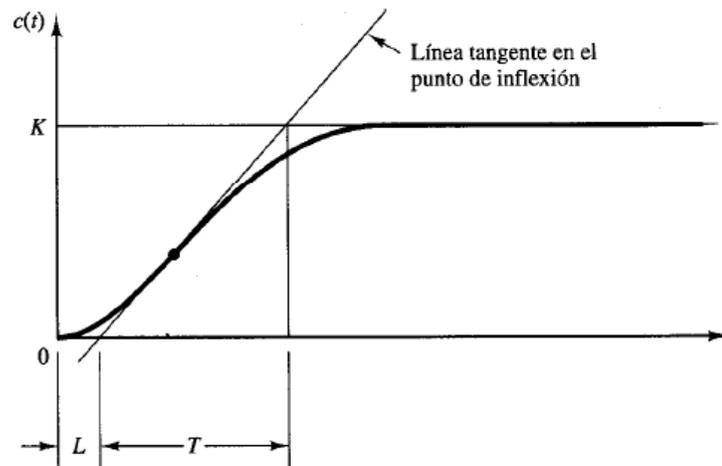
Ziegler e Nichols propuseram regras para determinar os valores do ganho proporcional  $K_p$ , do tempo integral  $T_i$  e do tempo derivativo  $T_d$ , com base na resposta transitória de uma planta. As regras propostas por eles visam obter um *overshoot* máximo de 25% quando o sistema é submetido a uma entrada do tipo degrau, como observado na figura 2.13.



**Figura 2.13 – Resposta ao degrau unitário com *overshoot* de 25%**

**Fonte: Ogata, 1999.**

Primeiramente, obtém-se a resposta da planta ao degrau em malha aberta, de maneira experimental. A planta deve apresentar uma resposta ao degrau semelhante à observada na figura 2.14, uma resposta em S e sem *overshoot*. Uma resposta em S implica em um sistema que não possui integradores e nem polos dominantes complexos conjugados. Este comportamento é fundamental para a aplicação do método. Caso a resposta da planta não possua este comportamento, o método não é adequado.



**Figura 2.14 – Curva de resposta ao degrau em MA**

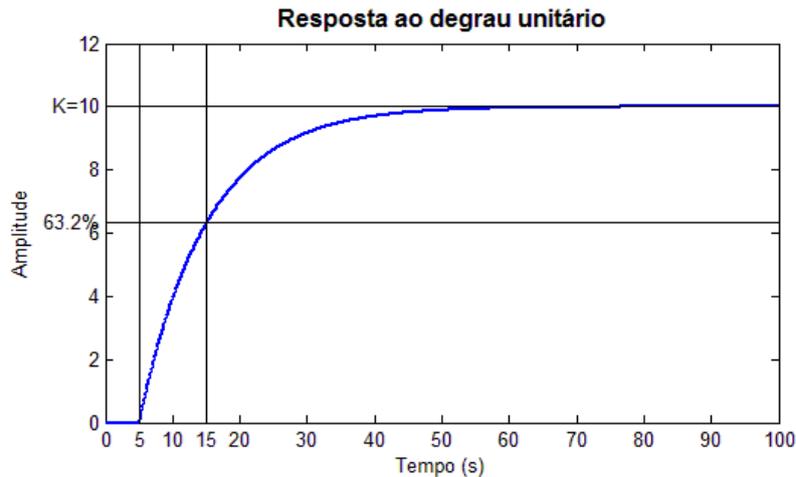
**Fonte: Ogata, 1999.**

A curva em S é essencial por poder ser aproximada a um sistema de primeira ordem com tempo morto. Tendo, portanto, como parâmetros principais o tempo de atraso  $L$ , a constante de tempo  $T$  e o ganho em malha aberta  $K$ , como descrito na equação 2.14.

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \quad (2.14)$$

Na figura 2.15 observa-se a resposta ao degrau unitário do sistema de primeira ordem descrito pela FT da equação 2.15, onde nota-se a semelhança com a curva em S esperada para o método Ziegler-Nichols.

$$\frac{C(s)}{U(s)} = \frac{10e^{-5s}}{10s+1} \quad (2.15)$$



**Figura 2.15 – Resposta ao degrau unitário de um sistema de primeira ordem com tempo morto**

Após a obtenção da resposta experimental do sistema ao degrau em malha aberta, deve-se estimar uma FT como a da equação 2.14, com ganho, tempo morto e constante de tempo. Para calcular os valores  $L$  e  $T$ , traça-se uma reta tangente ao ponto de inflexão da curva em  $S$  e observa-se o ponto de intersecção desta reta tangente com o eixo do tempo, figura 2.14. Pode-se também obter o tempo de atraso observando o gráfico de resposta ao degrau e pode-se assumir a constante de tempo como o instante em que a resposta atinge 63.2% de seu valor em regime permanente, como mostrado na figura 2.15.

Com a função de transferência estimada, os ganhos do controlador são calculados segundo a tabela 2.2.

**Tabela 2.2 – Método Ziegler-Nichols para sintonia de PID, baseado na resposta ao degrau unitário em MA**

Tipo de Controlador	$K_p$	$T_i$	$T_d$
P	$T/L$	$\infty$	0
PI	$0,9 * T/L$	$L/0,3$	0
PID	$1,2 * T/L$	$2L$	$0,5L$

**Fonte: OGATA, 1999.**

### 3 DESENVOLVIMENTO DO PROJETO

Neste capítulo apresentam-se as etapas para construção da planta utilizada nos experimentos, o detalhamento de partes do código desenvolvido em LabVIEW, as funcionalidades da GUI elaborada no mesmo e a estimação da função de transferência do sistema.

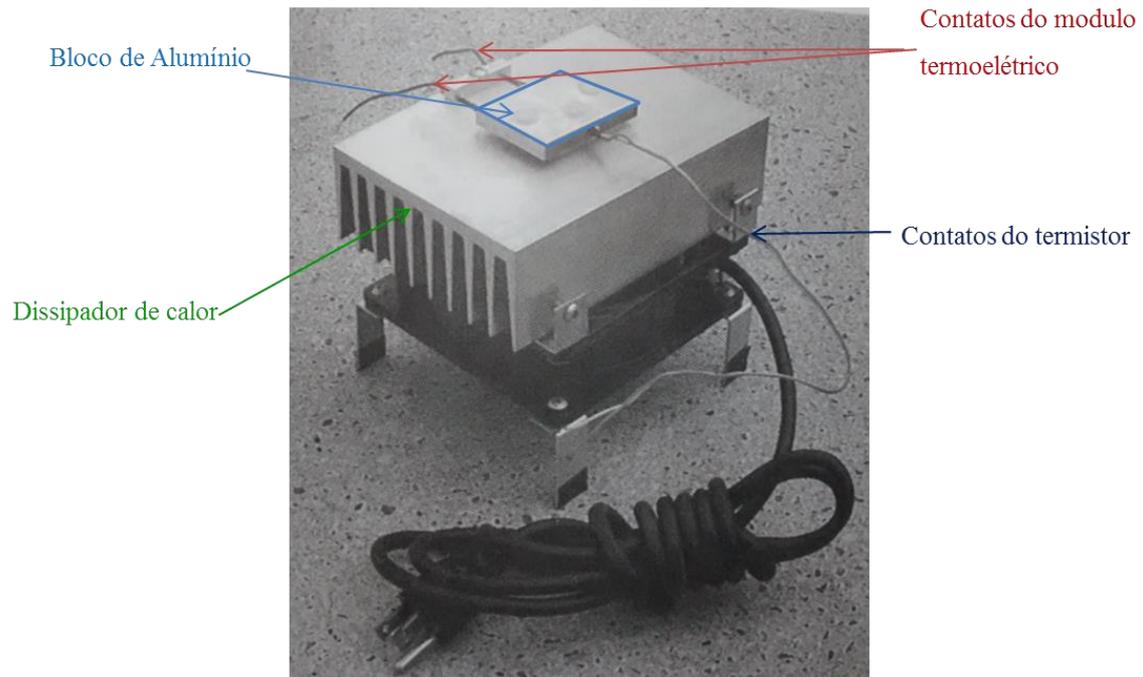
#### 3.1 Projeto da Planta

Primeiro é abordada a construção da planta, onde o controle de temperatura é aplicado, e o desenvolvimento do *driver* necessário para controlar a corrente aplicada ao módulo termoeletrico.

O objeto utilizado para ter sua temperatura controlada, foi um bloco de alumínio. Esse bloco possui dimensões pequenas (3,8cm x 4,1cm x 1,5cm) que facilitam as trocas de calor. Junto ao bloco devem ser conectados o módulo termoeletrico, o sensor de temperatura, além do dissipador de calor.

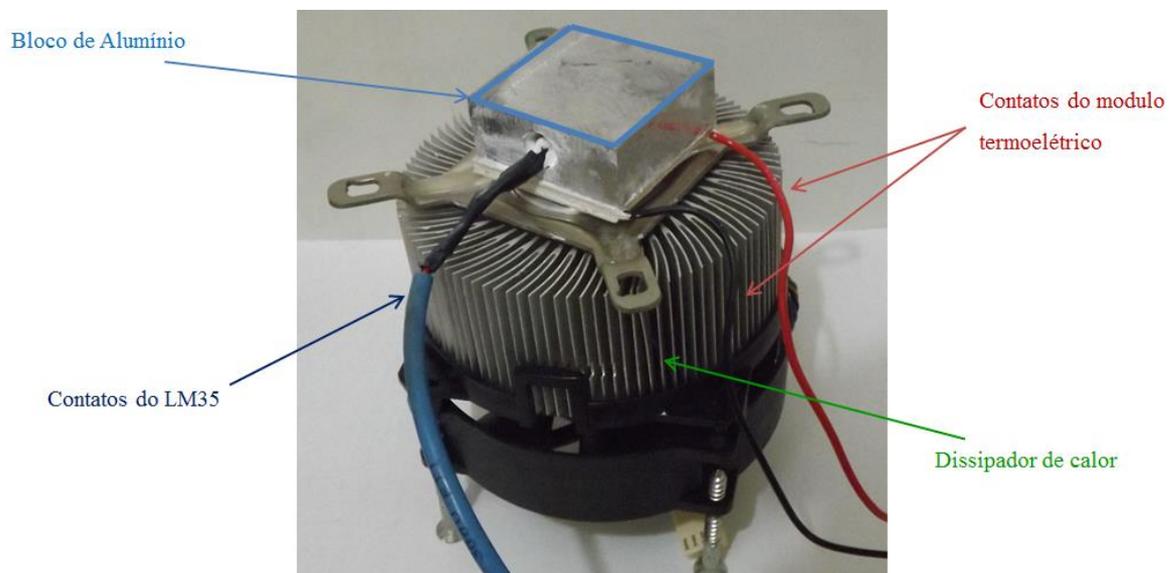
Na figura 3.1 pode-se observar um exemplo de montagem para o sistema de controle proposto. Uma estrutura semelhante foi construída utilizando materiais disponíveis nos laboratórios da Escola de Minas e/ou fáceis de serem adquiridos. O resultado pode ser verificado na figura 3.2.

Na montagem final constam todas as estruturas da planta que será controlada: o bloco de alumínio, o sensor de temperatura LM35 inserido em um orifício feito no bloco, o módulo termoeletrico, um dissipador de calor e uma ventoinha. Os contatos entre o sensor e o bloco, o bloco e o módulo termoeletrico e entre o módulo e o dissipador foram feitos utilizando pasta térmica visando uma melhor troca de calor entre estes equipamentos. O dissipador de calor, em conjunto com a ventoinha, atua como um reservatório de calor e facilita a troca de calor da superfície inferior do módulo termoeletrico com o meio.



**Figura 3.1 – Exemplo de montagem**

**Fonte: Essick, 2009**



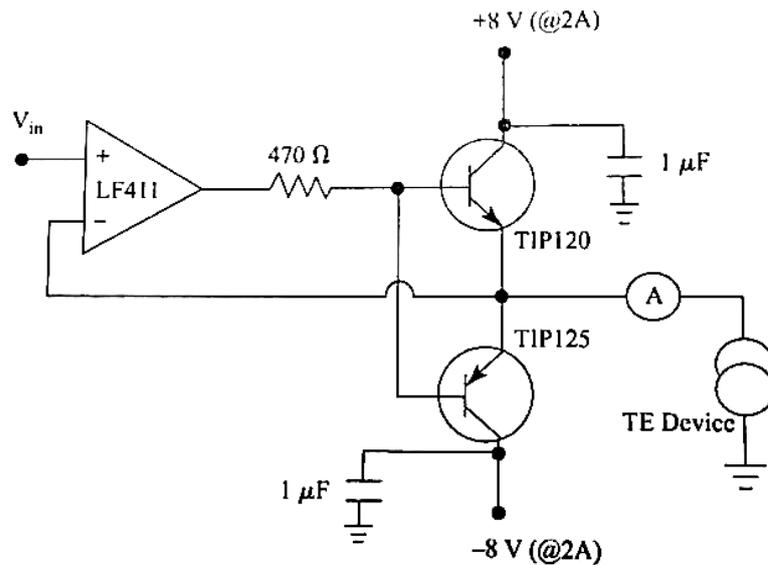
**Figura 3.2 – Estrutura desenvolvida para o projeto**

### **3.1.1 Driver para o controle da corrente**

Foi desenvolvido um *driver* para aplicar a corrente nos dois sentidos e na amplitude necessária ao funcionamento do módulo Peltier. O circuito consiste basicamente de dois

transistores de potência e um amplificador operacional e fornece ao módulo termoeétrico uma corrente proporcional ao sinal de entrada. A direção e módulo da corrente serão controlados pela entrada  $V_{in}$  do amplificador operacional, cuja saída é conectada as bases dos transistores, ou seja, tensões de entrada positiva levam a uma corrente de saída positiva e vice versa.

A figura 3.3 ilustra o circuito básico de um *driver* bidirecional (Essick, 2009), capaz de fornecer uma corrente de saída em dois sentidos, sendo essa variação de sentido crucial para a execução do controle de temperatura. Porém, a interface entre o sistema de controle no computador e o *driver* de corrente é feita pela placa micro controladora Arduino cujo sinal de saída é limitado a tensões positivas de 0 a 5V, então algumas alterações no circuito são necessárias para o funcionamento adequado do mesmo. Essas alterações são apresentadas a seguir.



**Figura 3.3 – Driver de corrente bidirecional controlado por tensão**

**Fonte: Essick, 2009**

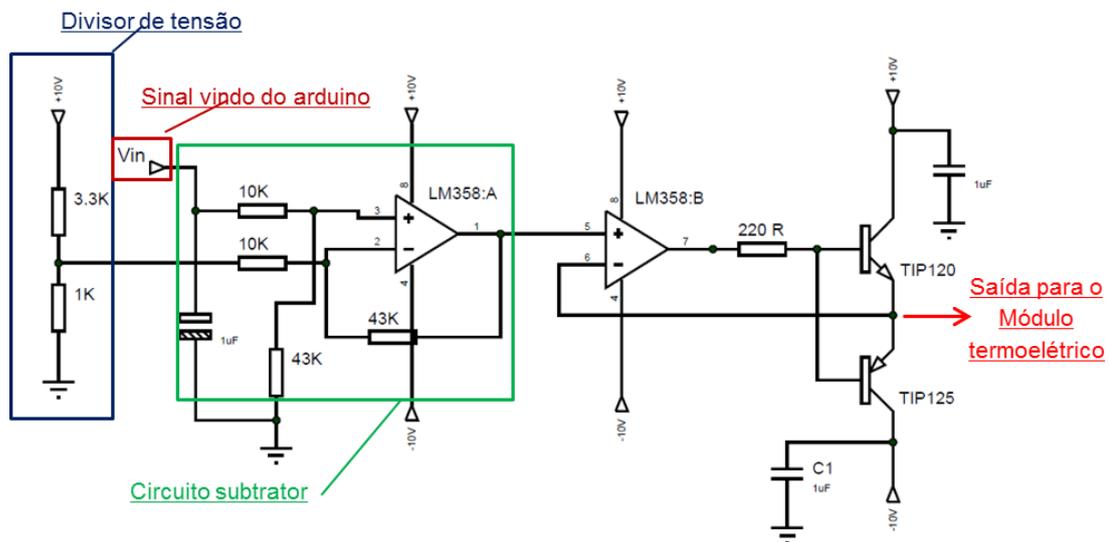
O circuito apresentado na figura 3.3 utiliza os transistores de potência TIP120 e TIP125 e o amplificador operacional LF411, este último foi substituído pelo LM358 sem que a funcionalidade original do sistema fosse alterada. O LM358 consiste de dois amplificadores independentes de alto ganho em um mesmo encapsulamento, que operam com uma única fonte de alimentação (ST MICROELECTRONICS, 2014). Para fornecer a alimentação do

sistema foi utilizado inicialmente uma fonte simétrica capaz de fornecer uma tensão de  $\pm 15V$  com uma corrente de 5A.

### 3.1.1.1 Primeira versão do *Driver*

Para atender a necessidade de um sinal de entrada simétrico (tensões negativas e positivas) utilizando a placa Arduino, que possui tensão de saída de 0 a 5V, a primeira solução encontrada foi subtrair 2,5V do sinal de saída utilizando a fonte de alimentação externa. Desta forma o sinal de saída aplicado ao *driver* varia de -2,5 a +2,5V. Em seguida o sinal foi amplificado para atender a tensão de trabalho do sistema, de modo que uma saída de 0V no Arduino representasse a máxima tensão negativa no sistema e uma saída de 5V representasse a máxima tensão positiva.

Para aplicar a estratégia descrita foram feitas as seguintes modificações explorando as diversas funcionalidades dos amplificadores operacionais segundo Wendling (2014). A alimentação do circuito foi alterada para -10V e +10V e um amplificador operacional (LM358), em conjunto com um divisor de tensão, foi utilizado para subtrair 2,5V do sinal de entrada e amplificá-lo em aproximadamente 4,3 vezes (razão entre os resistores de 43k $\Omega$  e 10k $\Omega$ ). O circuito resultante pode ser observado na figura 3.4.



**Figura 3.4 - *Driver* bidirecional modificado**

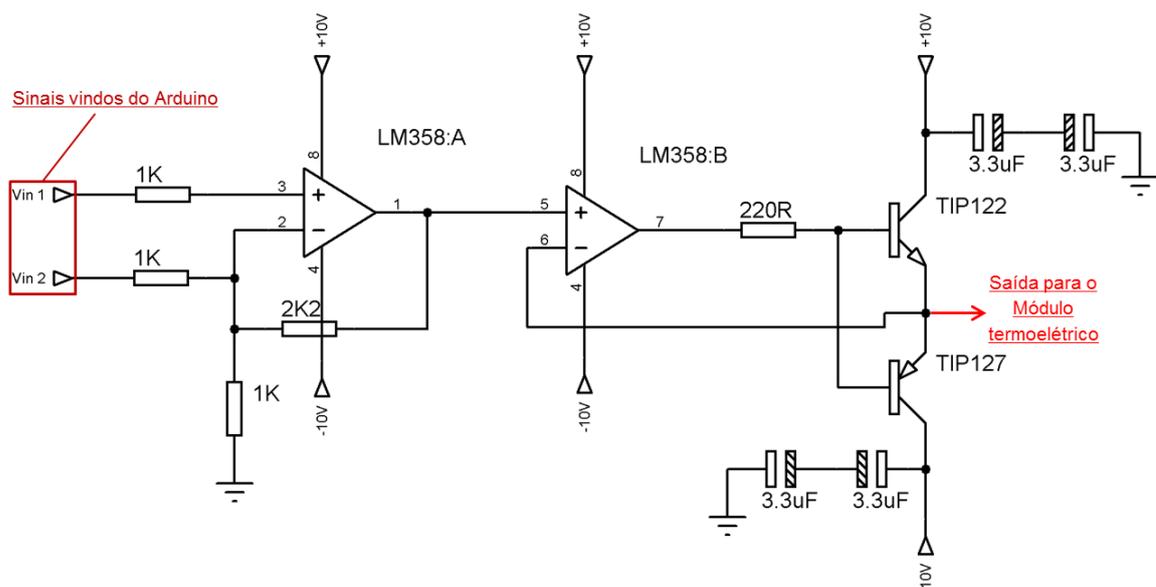
O divisor de tensão fornece uma tensão de 2,5V para ser subtraída do sinal de entrada vindo do Arduino. O circuito subtrator faz a subtração e amplifica o sinal em 4,3 vezes. Feito isso, o sinal de entrada efetivo do circuito amplificador do *driver* será um sinal entre -10,75V e

10,75V e não mais o sinal de 0 a 5V do Arduino. Esta alteração muda a referência do sistema fazendo com que sinais acima de 2,5V do Arduino forneçam corrente positiva ao módulo Peltier e sinais abaixo de 2,5 corrente negativa, possibilitando o controle bidirecional (resfriamento e aquecimento do módulo).

Esta metodologia possibilita o controle bidirecional da corrente, necessário para controlar a troca de calor do módulo Peltier nos dois sentidos. Porém, esta solução não se mostrou a ideal, uma vez que reduzia a precisão do sinal de entrada utilizando uma variação de sinal menor (2,5 a 5V ao invés de 0 a 5V) e pela dificuldade de fornecer um sinal de entrada nulo. O sistema proposto, ao utilizar um subtrator de 2,5V, faz com que o sinal de entrada varie de -2,5V a +2,5V (antes de amplificá-lo). Desta forma, um sinal de entrada nulo no sistema (saída do subtrator) equivale a um sinal de 2,5V do Arduino, pois a subtração resulta em zero. Porém, devido a imperfeições dos componentes e pequenos desbalanceamentos na fonte de alimentação é provável que esta subtração não seja exata levando a pequenas oscilações em torno do zero, de modo que uma corrente indesejada, mesmo que pequena, seja aplicada ao módulo quando o controlador está solicitando uma saída nula.

### 3.1.1.2 Segunda versão do *Driver*

Para solucionar os problemas encontrados com *driver* desenvolvido, melhorias foram feitas resultando em uma nova versão de *driver* para ser utilizado. O circuito resultante pode ser observado na figura 3.5.



**Figura 3.5 - *Driver* bidirecional modificado (segunda versão)**

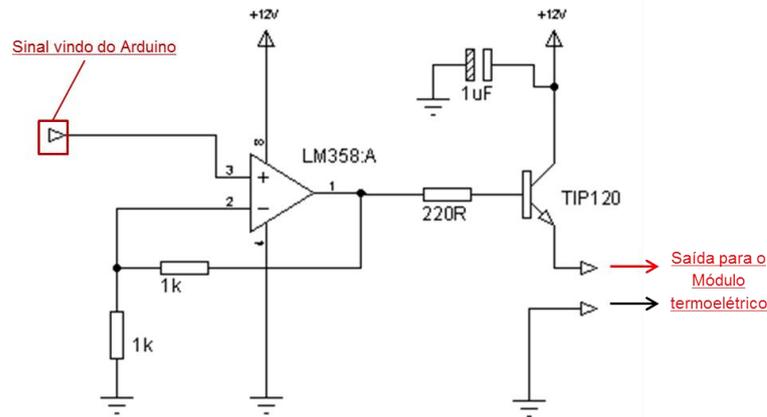
Este circuito possui como principal modificação a utilização de dois sinais de entrada vindos do Arduino, um sinal para controlar a corrente aplicada ao módulo em um sentido e outro sinal para controlá-la no sentido oposto. Ambos os sinais variam de 0 a 5V (sinal de saída do Arduino), porém o sinal Vin 2 será invertido pelo amplificador operacional LM358:A, de forma que a entrada Vin 1 forneça uma saída positiva e a entrada Vin 2 uma saída negativa. Assim, não é necessário reduzir a variação do sinal de entrada para 2,5V e o sistema possui um sinal de entrada efetivamente nulo (quando ambas as entradas assumem 0V). O LM358:A (figura 3.5) também amplifica o sinal de entrada, de forma a fornecer um sinal na tensão adequada ao LM358:B (figura 3.5). Neste sistema é necessário que o software de controle garanta que as duas entradas não serão ativadas simultaneamente, visando evitar uma subtração de sinais.

### 3.1.1.3 Terceira versão do *Driver*

Na primeira etapa da realização deste projeto, quando os *drivers* de potência estavam sendo idealizados e testados, foi utilizada uma fonte de corrente contínua robusta, capaz de fornecer uma tensão simétrica com corrente suficiente para atender ao funcionamento do sistema. Durante este período foi possível realizar alguns experimentos preliminares utilizando a segunda versão do *driver*, com um controle bidirecional de temperatura.

Porém, em uma etapa seguinte, com a interface em LabVIEW finalizada, os experimentos finais foram realizados utilizando uma fonte mais simples capaz de fornecer uma corrente satisfatória, contudo em um único sentido. Optou-se por uma fonte de alimentação para computador TDA, modelo ATX230WP4B, de 230W de potência real, com corrente de pico de 14A na saída +12V.

Deste modo, se fez necessário alterar a abordagem do projeto partindo para um controle unidirecional de temperatura, e não mais o controle bidirecional proposto inicialmente. Para isso uma nova versão mais simples do *driver* de potência foi utilizada, o circuito correspondente a esta nova versão pode ser observado na figura 3.6.

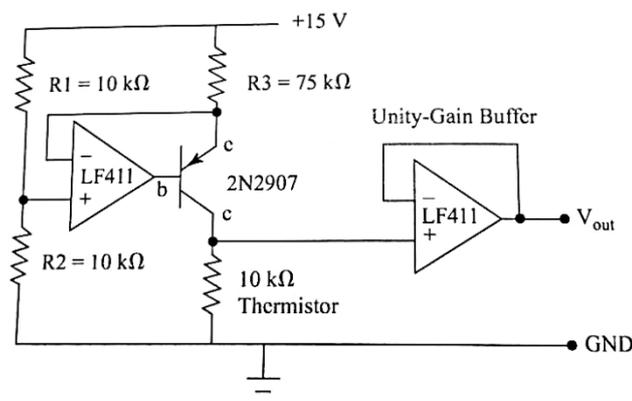


**Figura 3.6 – Driver unidirecional (terceira versão)**

Esta versão fornece ao módulo corrente em apenas um sentido gerando resfriamento ou aquecimento dependendo da polaridade da conexão do módulo com o *driver*. Neste circuito, a tensão fornecida pelo Arduino (0 a 5V) é amplificada em uma razão de 2 para 1 e entregue a base do transistor TIP120, que entrega esta tensão amplificada ao módulo, com uma corrente suficiente para realizar as trocas de calor. No Apêndice 1 deste trabalho encontra-se uma sugestão de montagem de uma placa de circuito impresso para este circuito.

### 3.1.2 Medição de temperatura

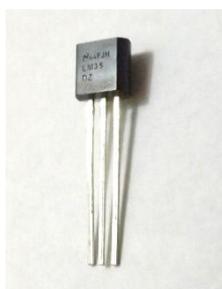
Para a obtenção dos dados de temperatura, um sistema para a medição de temperatura é necessário, uma vez que estes dados são interpretados por um software de controle. Essick (2009) sugere a utilização de um sistema com base em um termistor e dois amplificadores operacionais, observado na figura 3.7.



**Figura 3.7 - Hardware para medição de temperatura**

Fonte: Essick, 2009

No entanto, para simplificar a realização do projeto, optou-se pela utilização do sensor de temperatura de precisão LM35. O LM35 é um circuito integrado para medição de temperatura de precisão cuja tensão de saída é linearmente proporcional à temperatura em graus Celsius, sendo esta uma de suas vantagens, por não ser necessário a conversão de temperatura. Além disso, o LM35 não requer calibração e fornece uma precisão de  $\pm 1/4^{\circ}\text{C}$  em temperatura ambiente e  $\pm 3/4^{\circ}\text{C}$  na sua amplitude máxima, de  $-55$  a  $+150^{\circ}\text{C}$  (TEXAS INSTRUMENTS, 2015).



**Figura 3.8 – Foto do sensor LM35**

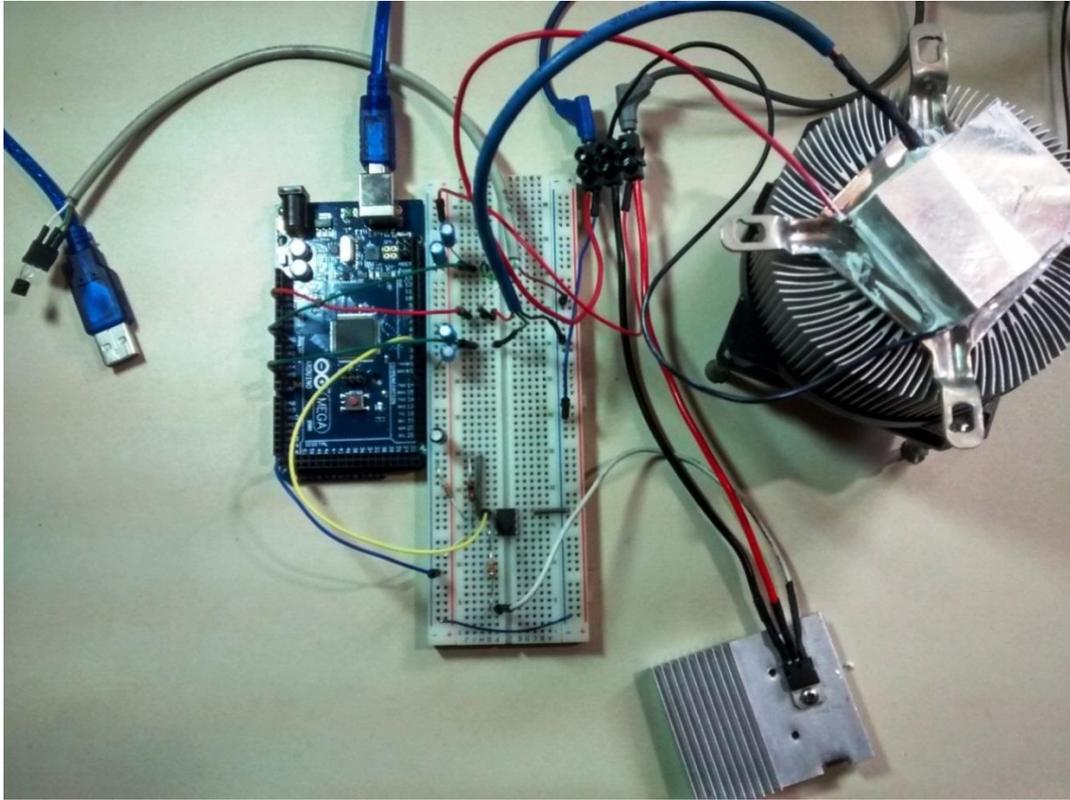
O sensor é alimentado pelo Arduino com uma tensão de  $+5\text{V}$  e a leitura de temperatura é feita por um pino de leitura analógica do Arduino que converte esta leitura em valores digitais para serem interpretados pelo sistema de controle desenvolvido em LabVIEW. Utilizando o sensor na configuração básica, ele possui uma resposta de  $10\text{mV}/^{\circ}\text{C}$  (TEXAS INSTRUMENTS, 2015).

Durante a realização dos experimentos com a fonte de alimentação de computador, observou-se um ruído considerável no sinal de leitura dos sensores LM35 utilizados. Visando minimizar o ruído presente nas leituras, utilizou-se quatro capacitores eletrolíticos de  $3,3\mu\text{F}$ , dois em cada entrada analógica do Arduino, conectados em paralelo entre as entradas analógicas e a referência (GND). A utilização dos capacitores foi positiva, melhorando a qualidade do sinal.

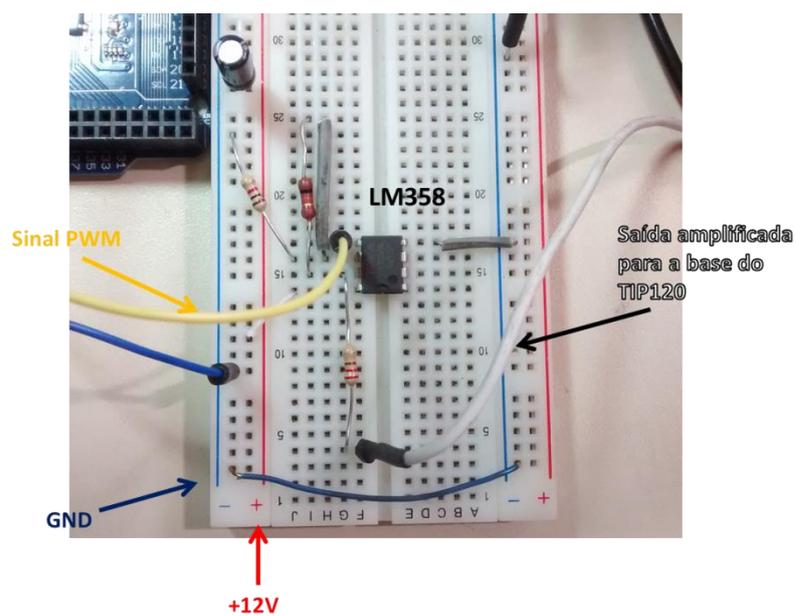
### **3.1.3 Planta Final Utilizada**

Na figura 3.9 a visão geral da planta utilizada para o controle unidirecional da temperatura pode ser observada. Nela nota-se o bloco de alumínio com o sensor de temperatura interno e o módulo Peltier, a placa Arduino, o sensor externo para a medição da temperatura ambiente, o circuito para amplificar o sinal do Arduino e o TIP 120 utilizado para entregar a corrente ao módulo. O circuito de amplificação utilizado pode ser observado de forma mais detalhada na

figura 3.10, onde se encontram indicados o pino correspondente ao sinal PWM vindo do Arduino e o pino com a saída para a base do transistor TIP120.



**Figura 3.9 – Visão geral da Planta**



**Figura 3.10 – Circuito do Driver de Potência Unidirecional**

## 3.2 Desenvolvimento do Sistema em LabVIEW

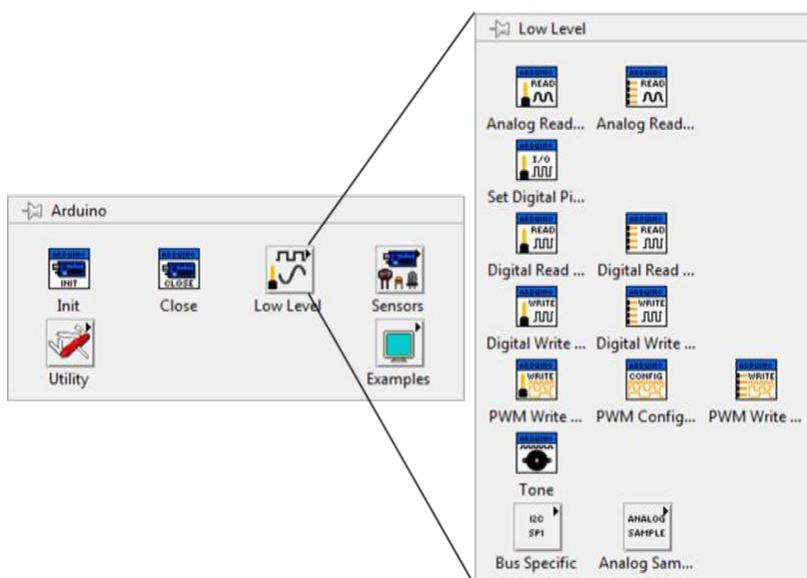
Este capítulo tem como objetivo apresentar a interface virtual desenvolvida em LabVIEW, evidenciando os seus elementos principais e explicando o seu funcionamento.

### 3.2.1 Conexão com o Arduino

A primeira etapa do programa desenvolvido no LabVIEW é estabelecer a conexão com o Arduino de forma que a interface entre o ambiente virtual e a planta possa ocorrer adequadamente. A comunicação com o Arduino foi programada no LabVIEW utilizando um *toolkit* próprio para este fim, chamado *LabVIEW Interface for Arduino*. Os *Toolkits* do LabVIEW são pacotes de expansão, como explicado pelo fabricante:

“Os *toolkits* adicionam bibliotecas de funções, VIs, assistentes interativos de configuração, exemplos, utilitários e documentação à sua instalação do LabVIEW, reduzindo o tempo necessário para a conclusão de suas tarefas. Eles podem ser acessados através de novas paletas ou opções de menu em seu ambiente de desenvolvimento. Os *toolkits* do LabVIEW atendem diferentes necessidades, de ferramentas para aplicações ou indústrias específicas a funções de uso geral.” (NATIONAL INSTRUMENTS, 2015)

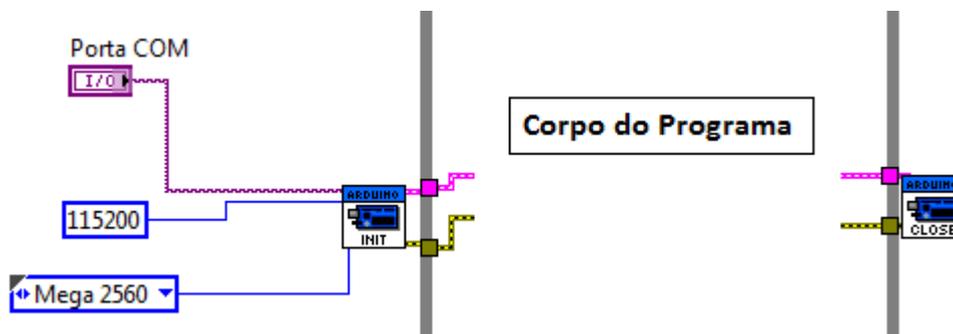
Este *toolkit* adiciona uma nova paleta com diversos blocos exclusivos para a interface com o Arduino, como observado na figura 3.11.



**Figura 3.11 - Blocos de interface com Arduino**

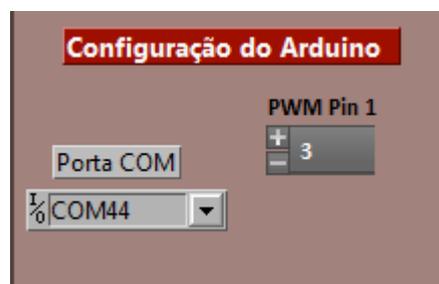
Para que a comunicação do LabVIEW com o Arduino possa acontecer através do *toolkit LabVIEW Interface for Arduino*, é necessário que o Arduino esteja executando um *firmware* específico, este *firmware* é fornecido junto ao *toolkit* e deve ser carregado no Arduino através do seu ambiente de programação.

A conexão é estabelecida através de comunicação serial, onde a velocidade de conexão depende do modelo de Arduino adotado. Os blocos de conexão com o Arduino (figura 3.12) definem a velocidade de comunicação (*baud rate*), que é configurada para 115200 bits/s, e o modelo do Arduino utilizado, neste caso, Mega 2560. A porta de comunicação utilizada também deve ser fornecida ao programa, e para isso um bloco de comando permite ao usuário escolher a porta serial na qual o Arduino está conectado.



**Figura 3.12 - Conexão com o Arduino**

Na figura 3.13 notam-se os blocos de comando posicionados no painel frontal do LabVIEW (no GUI desenvolvido), que permitem alterar a porta de comunicação serial e o pino de saída PWM que será utilizado ao longo da execução do programa.



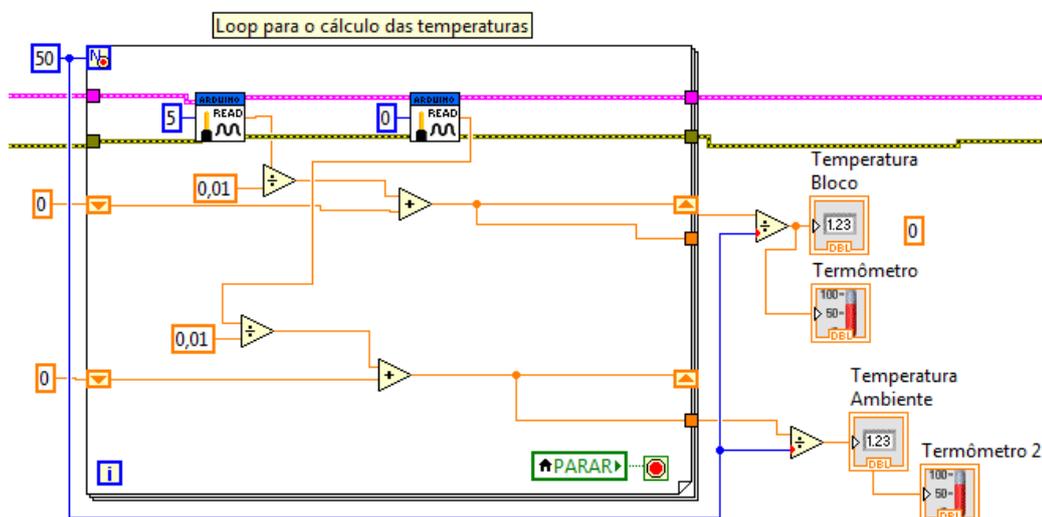
**Figura 3.13 - Configurações do Arduino na interface virtual**

### 3.2.2 Leitura da temperatura

Uma das partes fundamentais do sistema proposto é a leitura dos valores de temperatura fornecidos pelo sensor LM35. Esta leitura é feita utilizando dois pinos de leitura analógica do

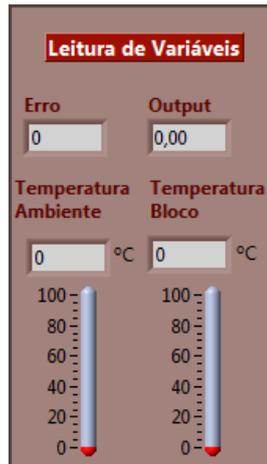
Arduino, cada pino conectado a um sensor de temperatura independente. Dois sensores são utilizados de forma que a temperatura ambiente possa ser medida junto com a temperatura do bloco de alumínio, que é a Variável de Processo.

Devido à grande precisão do sensor utilizado e a velocidade com a qual os dados são coletados, a temperatura instantânea medida pelo sensor sofre pequenas variações entre cada iteração do algoritmo do LabVIEW, criando uma pequena oscilação no sinal medido. Para garantir uma medição mais uniforme, optou-se pela utilização do valor médio de 50 medições, obtendo um sinal mais estável. Na figura 3.14 a parte do algoritmo responsável por ler o sensor de temperatura e calcular a média aritmética de 50 medições pode ser observada.



**Figura 3.14 - Cálculo do valor médio de temperatura**

A leitura é feita através dos pinos de entrada analógica de números 0 e 5 do Arduino. No canto superior esquerdo da figura 3.14 nota-se a quantidade de iterações que o *loop* irá realizar. A cada iteração os pinos são lidos, fornecendo um valor em Volts, este valor é dividido por 0,01 para ser convertido em graus Celsius, segundo o *datasheet* do LM35 (TEXAS INSTRUMENTS, 2015). Então, os valores são somados com os valores da iteração anterior (zero no caso da primeira iteração) e depois enviados para a próxima iteração. Após 50 leituras os valores somados são divididos por 50 e mostrados no *display* 'Leitura de Variáveis' como 'Temperatura Bloco' e 'Temperatura Ambiente' (figura 3.15). O valor de temperatura do bloco é realimentado ao sistema para ser comparado ao *setpoint*.



**Figura 3.15 - Display de leitura das variáveis**

O tempo gasto para o cálculo desta média aritmética é desprezível quando comparado ao tempo de resposta do sistema, que possui uma dinâmica naturalmente lenta, deste modo, o tempo gasto não tem influência significativa no sistema.

### 3.2.3 Métodos de Controle

O sistema desenvolvido permite alternar entre três métodos de controle, de modo que diferentes comportamentos da planta possam ser observados. Estes métodos são o Manual, onde o usuário determina livremente o valor de saída controlador, o controle On-Off, que permite ao usuário determinar o *setpoint* e o valor de histerese, e o controle PID, onde as constantes do controlador podem ser modificadas.

A escolha do método utilizado é feita pelo usuário através de abas no *display* de controles da interface desenvolvida (figura 3.16). Ao clicar na aba o programa faz a mudança para o método selecionado através de uma *Case Structure* (equivalente do LabVIEW a um *switch case* nas linguagens de programação mais comuns).

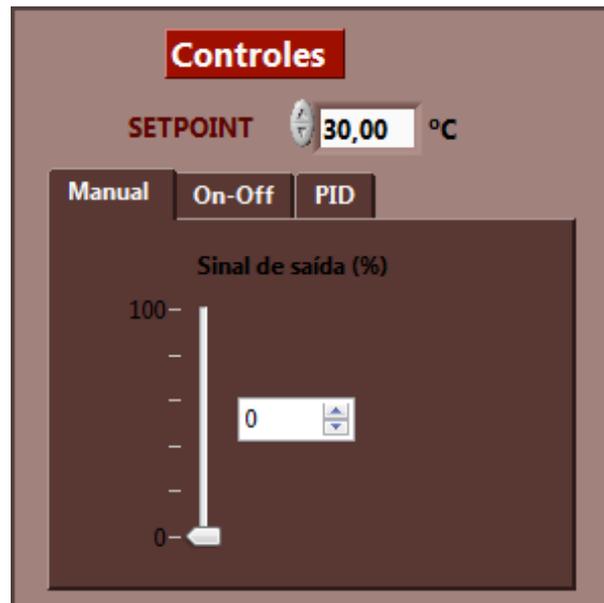


Figura 3.16 - Menu de Controles

### 3.2.3.1 Manual

Ao clicar na aba ‘Manual’ o usuário assume o controle do sinal de saída aplicado ao sistema. Este método permite que o usuário, através de um ponteiro deslizante, altere o valor do sinal de 0 a 100%, alterando o valor do *duty cycle* do PWM aplicado pelo Arduino ao driver de potência, sendo uma saída de 100% correspondente a um sinal de 5V no pino do Arduino. Na figura 3.17 pode-se observar o *Case Structure* no modo Manual, nele o programa lê o valor selecionado pelo usuário através do ponteiro deslizante, converte este valor para valores compatíveis com o PWM de 8bits do Arduino e envia para o *Output* do sistema.

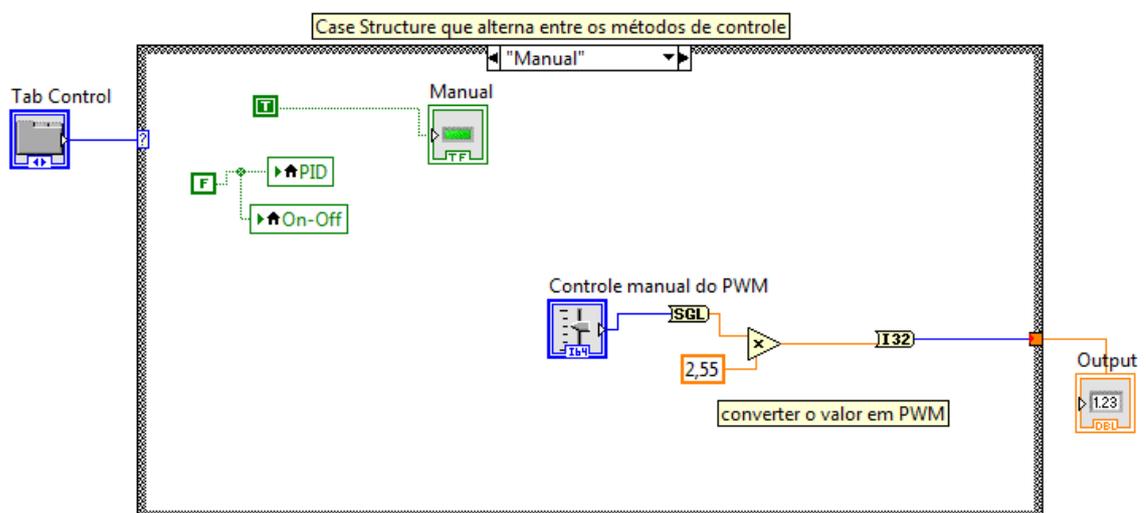
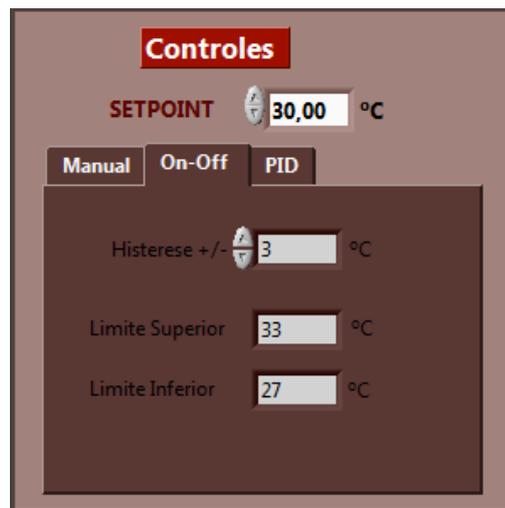


Figura 3.17 - Case Structure do modo Manual

### 3.2.3.2 On-Off

Ao alternar para o controle *On-Off*, o sistema automaticamente compara a temperatura do bloco com o valor de *setpoint* determinada no menu de controles (figura 3.18) e, dependendo do sentido do sinal de erro (positivo ou negativo), envia um valor máximo ou mínimo (zero) de saída para a planta.

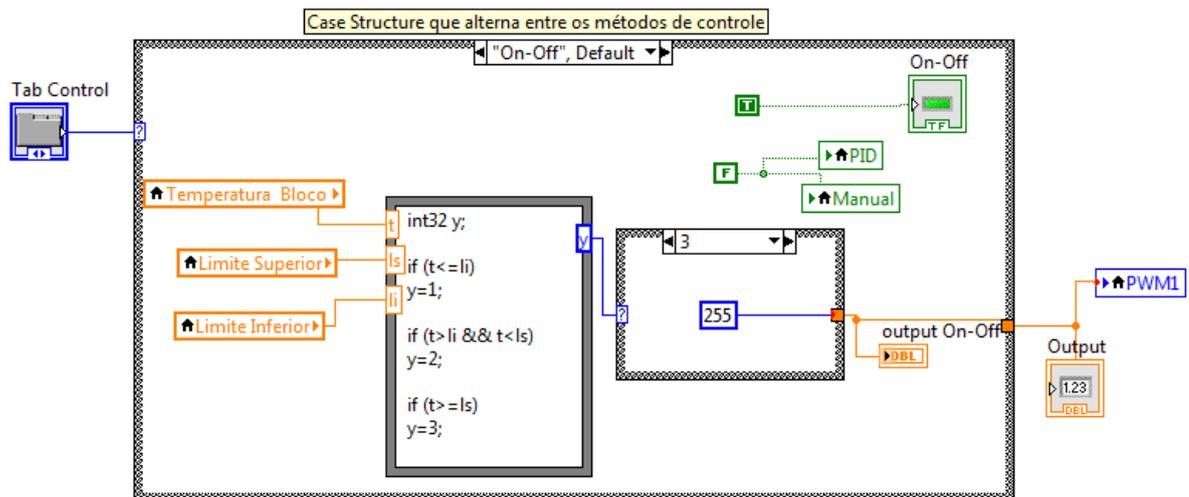


**Figura 3.18 - Menu de controle On-Off**

O controle *On-Off* permite ao usuário determinar o valor da histerese que será utilizada no controle, este valor é somado ao *setpoint* para estabelecer o Limite Superior do controle e subtraído do *setpoint* determinando o Limite Inferior. Estes valores são enviados para o *Case Structure* e são utilizados para determinar o valor de saída do controlador (figura 3.19).

No *Case Structure* do modo *On-Off*, o valor de temperatura do bloco é comparado com os limites superior e inferior da histerese, e dependendo do resultado desta comparação uma segunda *Case Structure* seleciona a saída para o sistema.

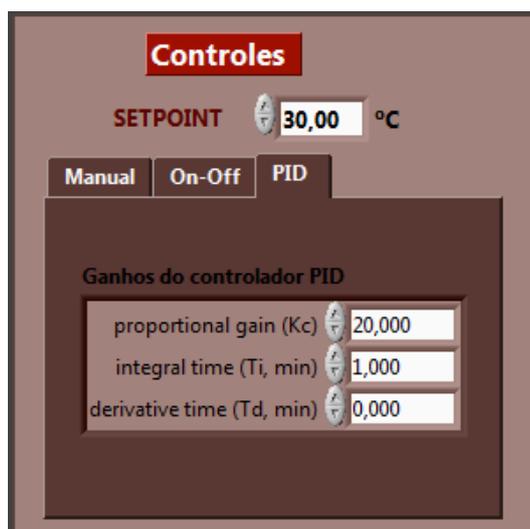
Com o sistema funcionando em apenas um sentido e sendo utilizado para resfriar o bloco, caso a temperatura medida se encontre abaixo do limite inferior, a saída é zero, caso a temperatura esteja acima do limite superior, a saída é 255. No entanto, se a temperatura estiver no intervalo entre os limites é mantido o último valor de saída aplicado. Para utilizar o sistema no sentido de aquecimento, deve-se inverter os valores de saída do *Case Structure* do modo *On-Off*.



**Figura 3.19 - Case Structure do modo On-Off**

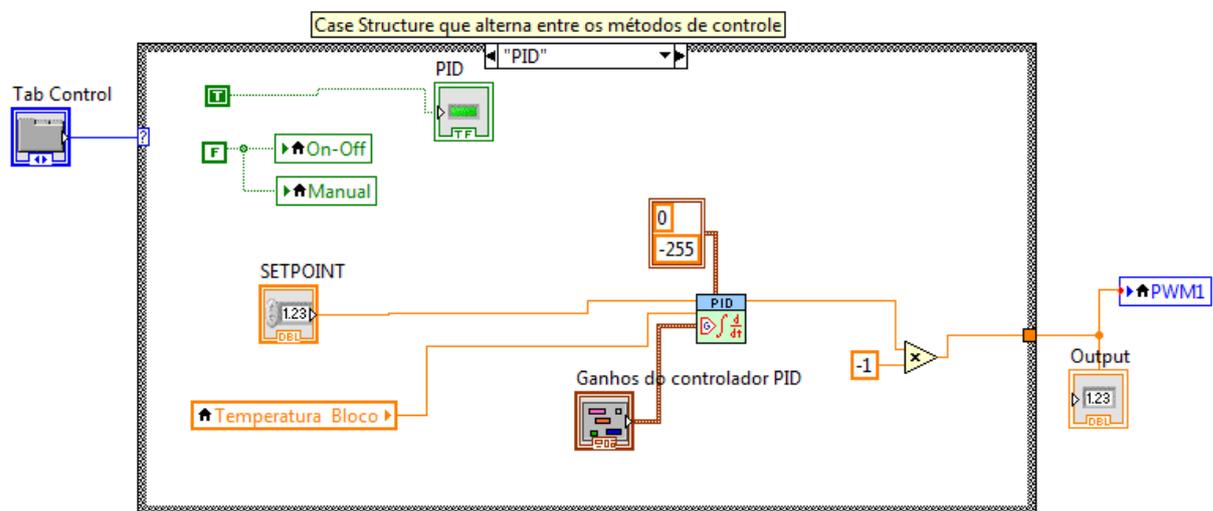
### 3.2.3.3 PID

O terceiro método de controle possível, nesta interface, é o controle PID. Nesta configuração, o programa permite ao usuário determinar as três constantes do controlador, o ganho proporcional, o tempo integral e o tempo derivativo (figura 3.20). Ao ser acionado, o controlador PID automaticamente começa a corrigir o sinal de saída de acordo com o erro medido. Neste caso o valor de saída irá depender da amplitude e variação do erro e das constantes de controle utilizadas, de acordo com a teoria apresentada na seção 2.4.2 deste trabalho.



**Figura 3.20 - Menu de controle PID**

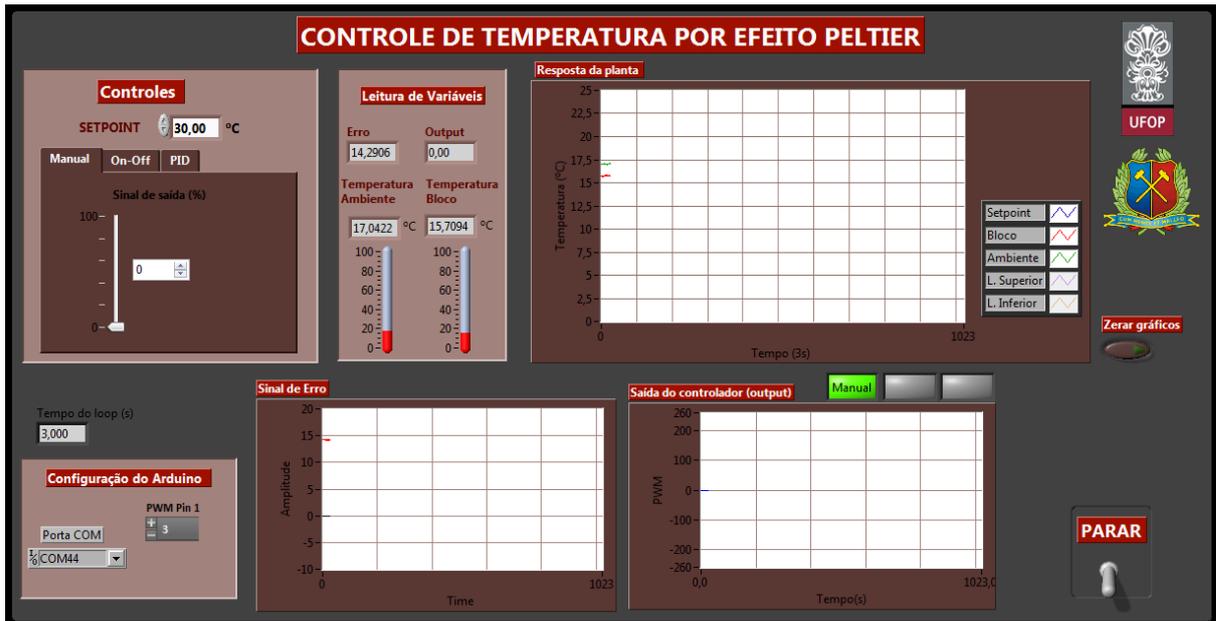
A utilização da estratégia de controle PID é facilitada pelo software LabVIEW através do módulo *Control Design & Simulation*, que permite a simulação de sistemas dinâmicos, o projeto e a implementação de controladores em tempo real, possuindo um bloco exclusivo para o controle PID. Como observado na figura 3.21, o programa fornece ao bloco de ação PID o valores de *setpoint*, o valor da variável de processo (temperatura do bloco), o ganhos do controlador e o intervalo de saída do controlador (0 a -255, no resfriamento). Com estes valores o LabVIEW calcula automaticamente o sinal de erro e a resposta do controlador. Para utilizar o sistema visando o controle de aquecimento do bloco, os valores limites do controlador devem ser alterados para +255 e 0.



**Figura 3.21 - Case Structure do modo PID**

### 3.2.4 Interface de Usuário

O GUI (*Graphical User Interface*) desenvolvido neste trabalho pode ser observado na figura 3.22. Nele constam o painel de controles, onde o usuário determina o *setpoint* e a ação de controle que será utilizada, o painel de configuração dos pinos do Arduino, o painel de leitura de variáveis, o gráfico que mostra as temperaturas envolvidas (resposta da planta), o gráfico do sinal de saída do controlador além do gráfico da variação do erro ao longo do tempo. O gráfico Resposta da planta exibe, de acordo com a escolha do usuário, a temperatura do bloco, a temperatura ambiente, a temperatura desejada e limite superior e inferior, no caso do controle *On-Off*.



**Figura 3.22 - Interface de usuário desenvolvida**

O diagrama de blocos completo, referente a esta interface desenvolvida, é apresentado no Apêndice 2 deste trabalho.

### 3.2.5 Tempo de Amostragem e Limitações da Interface

A programação em LabVIEW se dá de uma forma bastante diferente do usual aprendido ao longo dos cursos de graduação. Geralmente são estudadas programações em linhas de código, enquanto o LabVIEW trabalha com uma programação por blocos que é muitas vezes mais intuitiva. Porém, a utilização desta programação gráfica pode se tornar uma desvantagem dependendo da lógica que se deseja programar. Às vezes por limitação da linguagem ou, então, pela falta de familiaridade com a mesma, o programa final acaba por não possuir uma estrutura otimizada, utilizando mais processamento do que seria necessário.

Na plataforma desenvolvida este excesso de processamento foi notado na velocidade de resposta do sistema. Nas primeiras versões a interface respondia de forma rápida, sendo necessária a utilização de atrasos de tempo para atingir uma frequência de amostragem adequada. Já nas versões finais, após a unificação das aplicações em um só programa, a resposta se mostrou mais lenta, fornecendo um tempo de execução superior a dois segundos.

No projeto proposto, devido à dinâmica lenta do sistema, o tempo de execução não se mostrou problemático e, para utilizar um valor inteiro, o intervalo de amostragem foi fixado em três

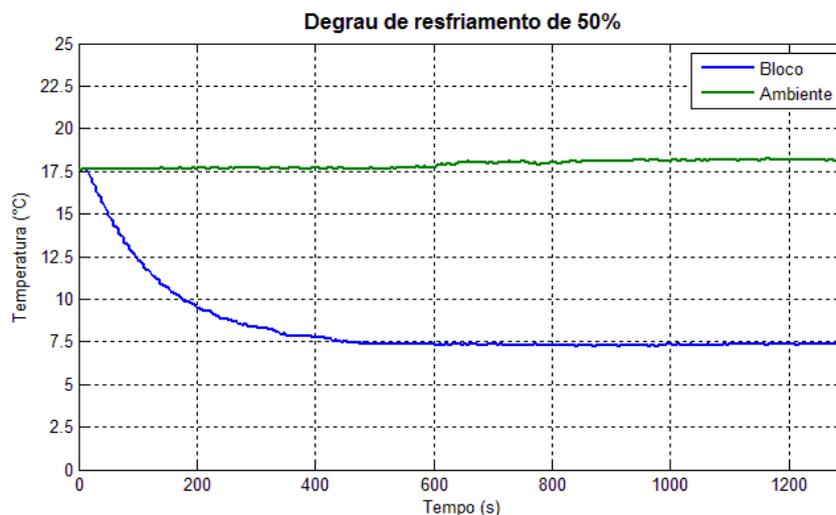
segundos. Este tempo de amostragem é configurado através do tempo execução do *loop while* no qual o programa é inserido.

Outra limitação verificada durante a execução, foi o tamanho máximo do vetor de dados dos sinais plotados nos gráficos. O LabVIEW possui um tamanho máximo de 1023 pontos para estes sinais, no entanto, como o tempo de amostragem foi de três segundos, os 1023 pontos foram suficientes para o armazenamento dos dados relativos aos experimentos realizados.

### 3.3 Estimação da Função de Transferência e Ganhos do Controlador

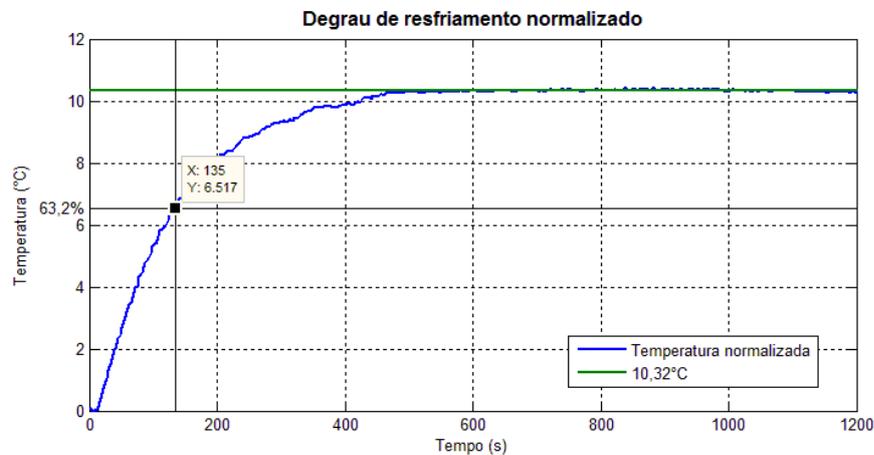
Neste capítulo será apresentada a metodologia para estimar a função de transferência do sistema. A estimação desta função de transferência se faz necessária uma vez que esta será utilizada no cálculo dos ganhos do controlador. É importante mencionar que o objetivo do trabalho é aplicar de forma simples e didática o controle de temperatura, e para isso é utilizada uma metodologia simples e funcional para a estimativa da função de transferência, que atenda às demandas do projeto.

Primeiramente é observada a resposta do sistema em malha aberta a um degrau. No caso opta-se por utilizar um degrau correspondente a 50% do valor máximo (saturação), ou seja, aplica-se um *duty cycle* de 50% na saída analógica do Arduino (128 de 255 bits), correspondendo a uma tensão de aproximadamente 5V entregue ao módulo termoeletrônico. Os dados obtidos com este experimento são salvos em uma planilha para posteriormente serem lidos e trabalhados com o *software* Matlab. A resposta do sistema a este estímulo, bem como a temperatura ambiente ao longo do experimento são apresentadas na figura 3.23.



**Figura 3.23 - Resposta em Malha Aberta ao degrau de resfriamento de 50%**

A partir destes dados o sistema é analisado e observa-se um comportamento que se aproxima ao de um sistema de primeira ordem com um pequeno atraso de tempo, uma vez que responde exponencialmente não apresentando *overshoot*. Sendo assim, visando à simplicidade de execução do projeto, optou-se por estimar uma FT de primeira ordem. Observando o tempo decorrido até que o sistema apresente uma mudança de temperatura a partir da aplicação do degrau, estimou-se um atraso de 10 segundos. Em seguida a resposta do sistema foi normalizada subtraindo-se a temperatura inicial e tomando os valores absolutos desta subtração, obtendo-se um sinal iniciando em zero e atingindo aproximadamente 10,32°C em regime permanente, como observado na figura 3.24.



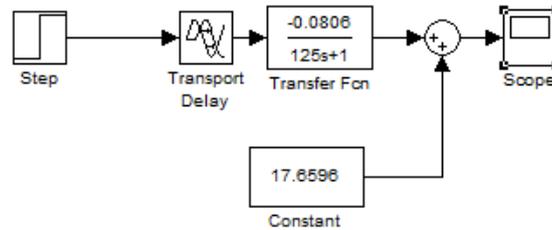
**Figura 3.24 - Degrau de resfriamento de 50% normalizado**

O valor de 10,32°C é a média das leituras a partir dos 700 segundos, ponto onde o sistema encontra-se claramente em regime permanente. O tempo gasto para atingir 63,2% do valor em regime permanente pode ser adotado como a constante de tempo  $T$  do sistema de primeira ordem. No caso do experimento este valor é alcançado após 135 segundos, e, descontando o atraso de tempo considerado de 10 segundos, obtém-se uma constante de tempo de 125 segundos. O ganho  $K$  do sistema é a razão entre o valor em regime permanente e a entrada, no caso a razão entre 10,32°C (temperatura final normalizada) e -128 (valor aplicado ao Arduino correspondente a um *duty cycle* de 50%), que resulta em -0,0806. O ganho possui um valor negativo, uma vez que atua no resfriamento do sistema.

Com os valores calculados, obtém-se a FT descrita na equação 3.1:

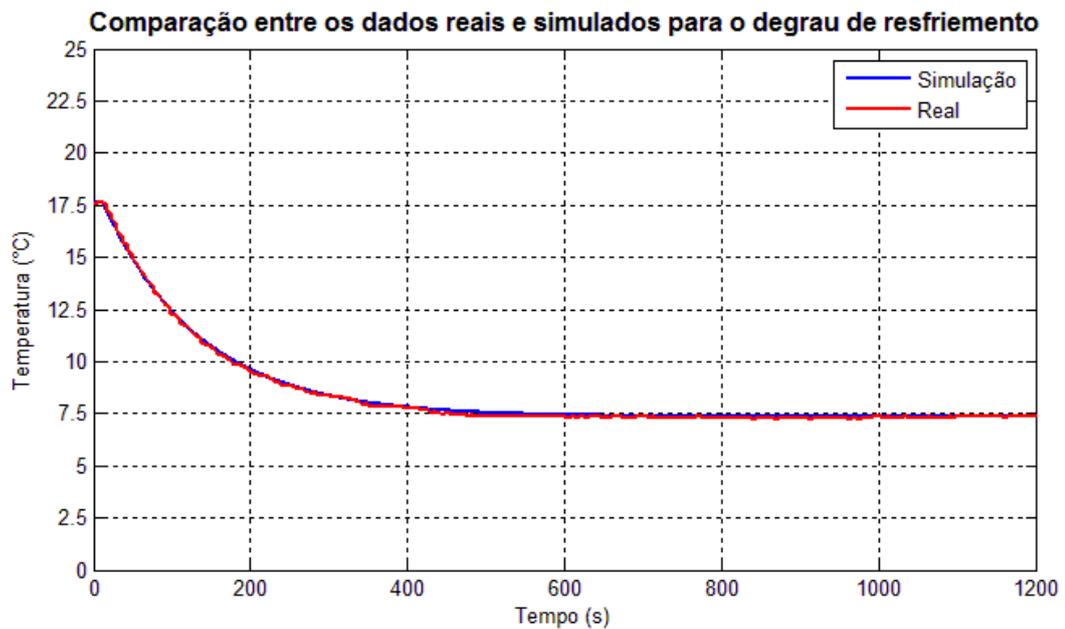
$$G(s) = e^{-10s} \frac{-0,0806}{125s + 1} \quad (3.1)$$

Para verificar a semelhança entre a FT obtida e os valores coletados com o experimento, foi realizada a uma simulação com a ferramenta Simulink do Matlab. O diagrama de blocos da simulação pode ser observado na figura 3.25.



**Figura 3.25 - Simulação com a FT obtida**

No diagrama de blocos da simulação (figura 3.25) aplicando um degrau de -128 em Malha Aberta à FT estimada para o sistema, considerando um atraso de tempo (*Transport Delay*) de 10 segundos e adicionando o valor de 17,6596°C à resposta do sistema obtém-se um gráfico normalizado com as condições iniciais do experimento. Os valores provenientes desta simulação, junto aos dados do experimento, são apresentados na figura 3.26.



**Figura 3.26 - Comparação entre o degrau de resfriamento real e simulado**

Observando o gráfico da figura 3.26 é possível concluir que a FT estimada apresenta uma resposta bastante semelhante aos dados reais e, portanto, representa de forma satisfatória o sistema proposto.

A partir da FT estimada os ganhos do controlador PID foram calculados pelo método Ziegler-Nichols mostrado na seção 2.4.3. Uma vez que o sistema apresenta um comportamento semelhante a um sistema de primeira ordem com atraso, este método é adequado para a sintonia do controlador.

Os ganhos obtidos para o controlador são apresentados na tabela 3.1.

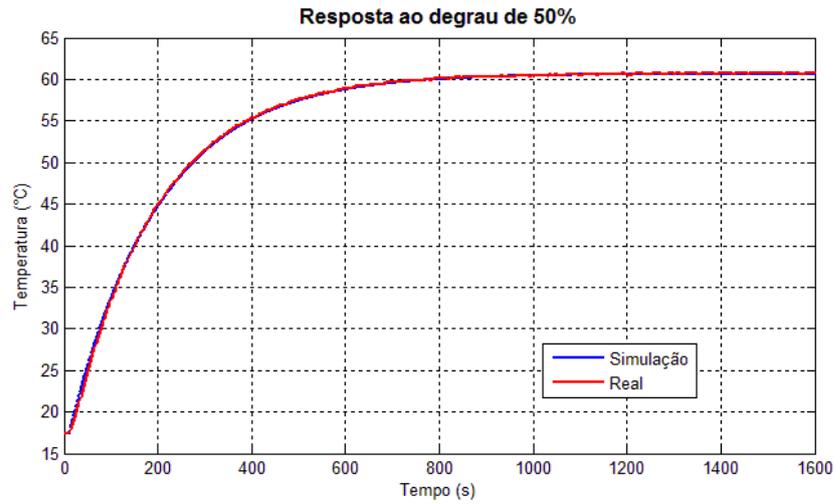
**Tabela 3.1 - Ganhos do Controlador para o resfriamento**

Tipo de Controlador	Kp	Ti	Td
P	12,5	-	-
PI	11,25	33,33333333	-
PID	15	20	5

De forma análoga é realizado um experimento no sentido de aquecimento do bloco, invertendo-se a polaridade da conexão entre o módulo termoeletrico e o driver. É aplicado um degrau de 50% do valor de saída, que equivale a um valor de 128 aplicado ao PWM do Arduino. A função de transferência obtida é representada na equação 3.2.

$$G(s) = e^{-10s} \frac{0,3385}{188s + 1} \quad (3.2)$$

O comportamento do sistema ao degrau e o resultado da simulação com o FT obtida, podem ser vistos na figura 3.27.



**Figura 3.27 – Resposta ao degrau de 50% no aquecimento**

Com a FT obtida para o aquecimento pelo método Ziegler-Nichols, são calculados os ganhos do controlador PID, exibidos na tabela 3.2.

**Tabela 3.2 - Ganhos do Controlador para o aquecimento**

Tipo de Controlador	$K_p$	$T_i$	$T_d$
P	18,8	-	-
PI	16,92	33,33	-
PID	22,56	20	5

## 4 RESPOSTA DO SISTEMA E ANÁLISE DOS RESULTADOS

Este capítulo é dedicado à apresentação dos experimentos de controle da temperatura do bloco através de resfriamento para um mesmo *setpoint*, utilizando o controle *On-Off* e o PID, com diferentes configurações de controladores. Também é apresentada a resposta do sistema ao aquecimento para o controle PID.

Para a realização dos experimentos é utilizada uma fonte de alimentação de 12V, de 230W. Como a fonte não é simétrica, o controle bidirecional de temperatura não é executado, sendo realizado o controle em um único sentido (resfriamento ou aquecimento) com o driver específico para este fim. No entanto, um experimento realizado previamente, utilizando uma fonte simétrica com o driver bidirecional, é apresentado no subcapítulo 4.4.

### 4.1 Resposta ao controle *On-Off*

A primeira resposta analisada é a resposta ao método de controle mais simples, o controle *On-Off* no sentido de resfriamento. No experimento é adotado um valor desejado (*setpoint*) de 8°C e, num primeiro momento, é realizado o controle sem histerese. O resultado deste experimento pode ser observado nas figuras 4.1 e 4.2.

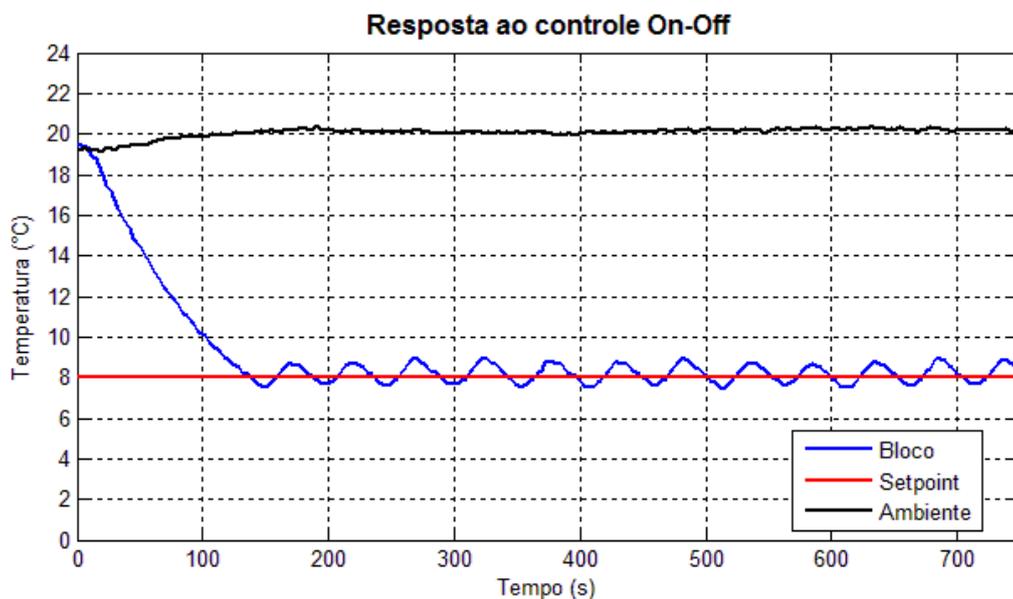
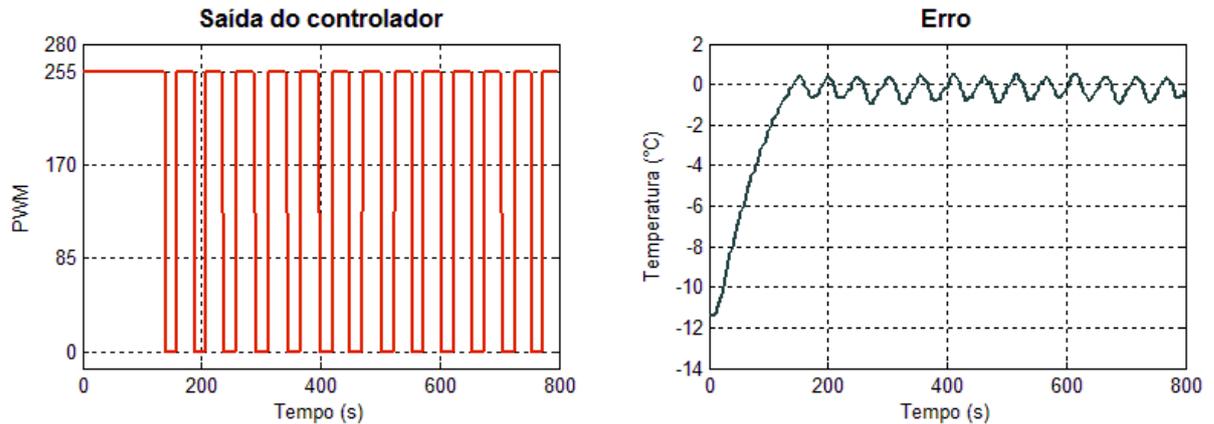


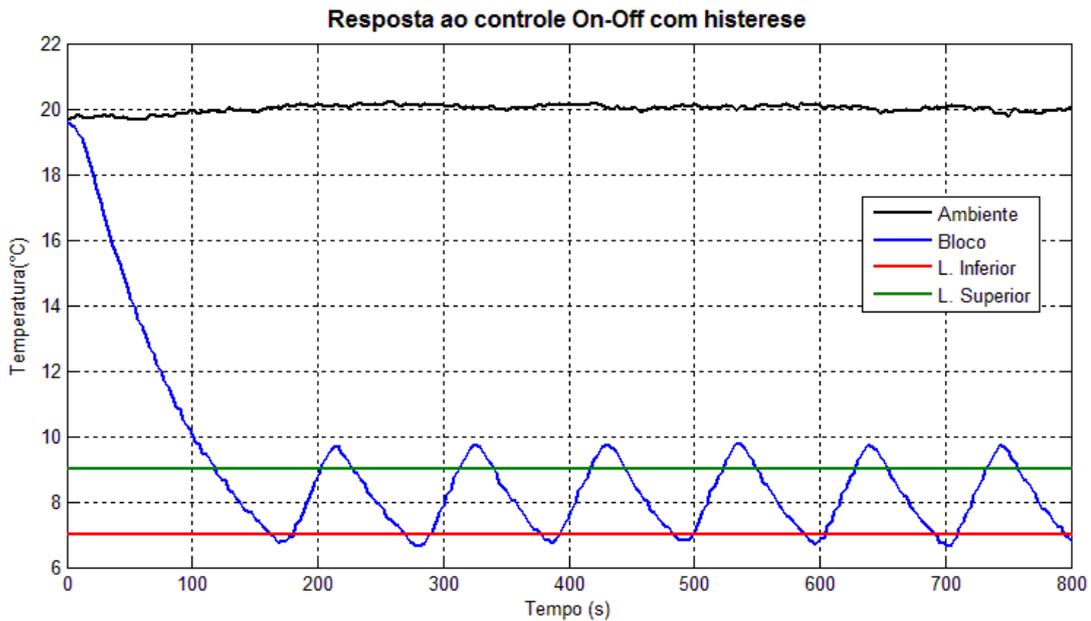
Figura 4.1 - Resposta ao controle *On-Off* no resfriamento



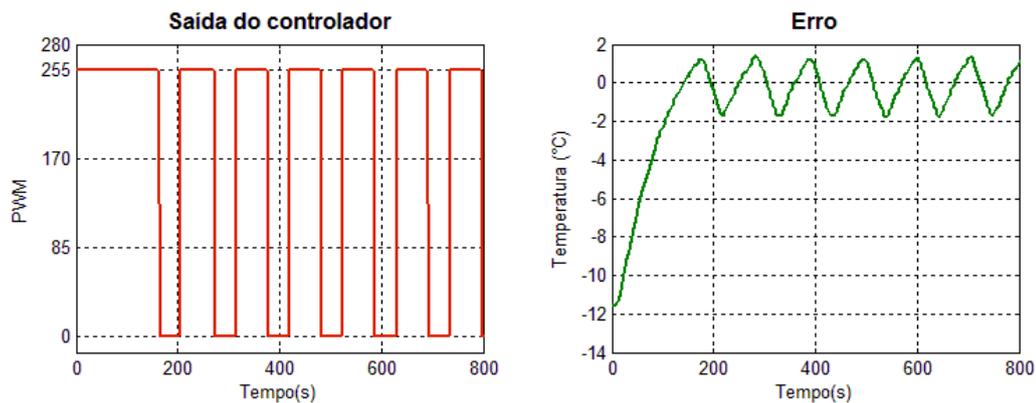
**Figura 4.2 - Saída do controlador e erro na resposta ao controlador *On-Off* no resfriamento**

O sistema se comporta conforme esperado, atingindo o valor de temperatura desejado com uma oscilação constante em torno do mesmo. O erro do sistema, em regime permanente, oscila em torno do zero com valores entre aproximadamente  $+0,4^{\circ}\text{C}$  e  $-0,7^{\circ}\text{C}$ . Como é utilizado o controle sem histerese, a saída do controlador possui uma oscilação com uma frequência significativa. Esta frequência de acionamento do módulo pode ser prejudicial caso seja utilizado um acionamento por relés ao invés de um dispositivo semiconductor. Como mencionado no capítulo 2.4.1, uma alternativa para minimizar a frequência de acionamento é a utilização de uma histerese. Porém, esta histerese tem como consequência imediata o aumento da amplitude da oscilação, uma vez que cria uma faixa de valores desejados ao invés de um único valor de *setpoint*.

Nas figuras 4.3 e 4.4 a resposta ao controle *On-off* com histerese de  $\pm 1^{\circ}\text{C}$ , para um *setpoint* de  $8^{\circ}\text{C}$ , pode ser observada. No gráfico da resposta do sistema verifica-se o limite superior de  $9^{\circ}\text{C}$  e o limite inferior de  $7^{\circ}\text{C}$ . Quando a temperatura do bloco encontra-se dentro desta faixa de operação, o sinal do controlador não se altera. Desta forma a frequência de acionamento do módulo é menor quando comparada ao controle sem histerese, no entanto, o erro em regime permanente possui uma oscilação maior em torno do zero, variando entre aproximadamente  $+1,2^{\circ}\text{C}$  e  $-1,7^{\circ}\text{C}$ .



**Figura 4.3- Resposta ao controle *On-Off* com histerese no resfriamento**



**Figura 4.4 - Saída do controlador e Erro na resposta ao controle *On-Off* com histerese no resfriamento**

## 4.2 Resposta ao controlador P, PI e PID no resfriamento

Utilizando os valores de ganhos do controlador obtidos pelo Método Ziegler-Nichols, são aplicados degraus na referência do sistema em malha fechada para observar a resposta do sistema aos três diferentes controladores P, PI e PID.

Primeiramente é observada a resposta do sistema a um degrau com o controlador P. O ganho adotado para o controlador foi  $K_c = 12,5$ , sendo aplicado um *setpoint* de  $8^\circ\text{C}$ , configurados conforme a figura 4.5. A resposta do sistema pode ser vista nas figura 4.6 e 4.7.

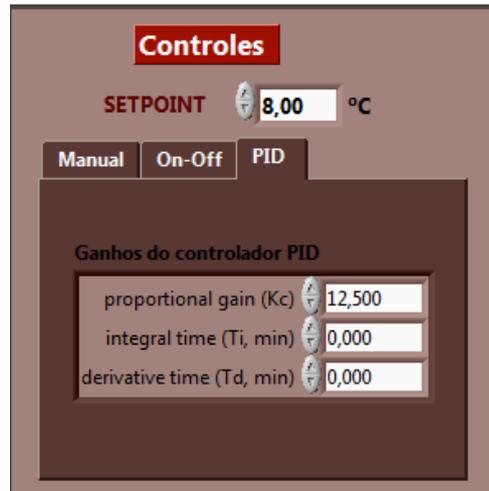


Figura 4.5 - Configuração do controle P no LabVIEW

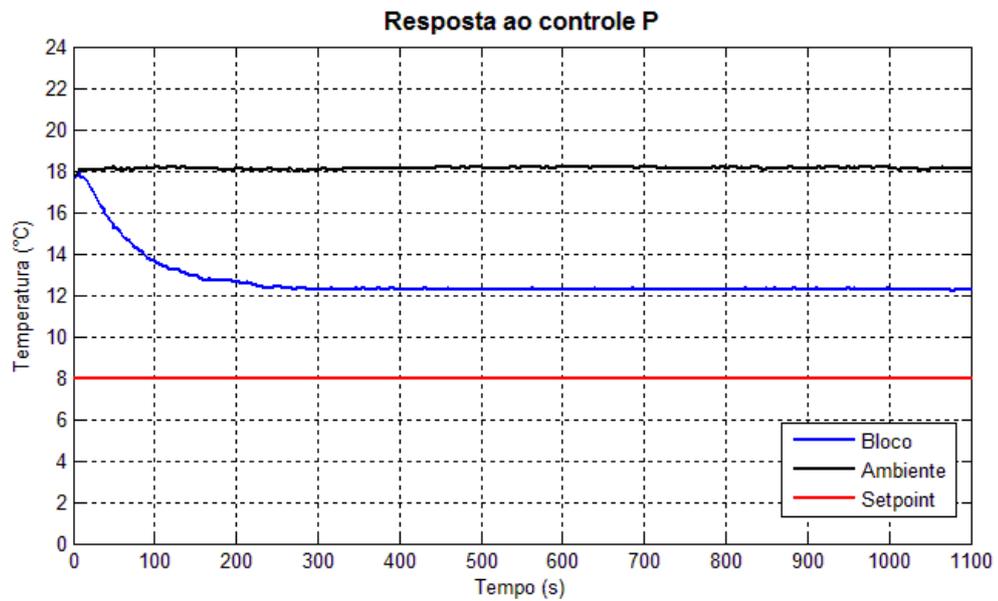
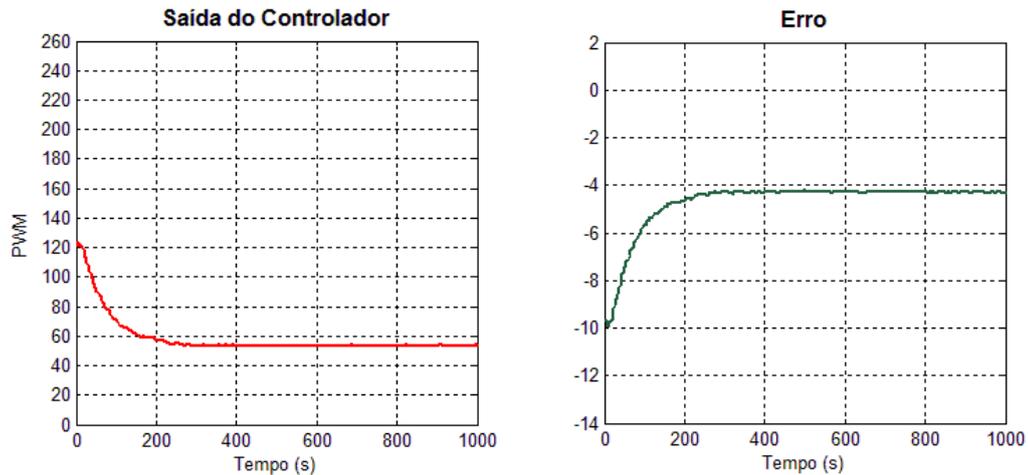
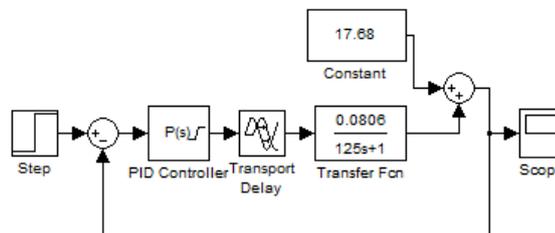


Figura 4.6 - Resposta ao controle Proporcional no resfriamento



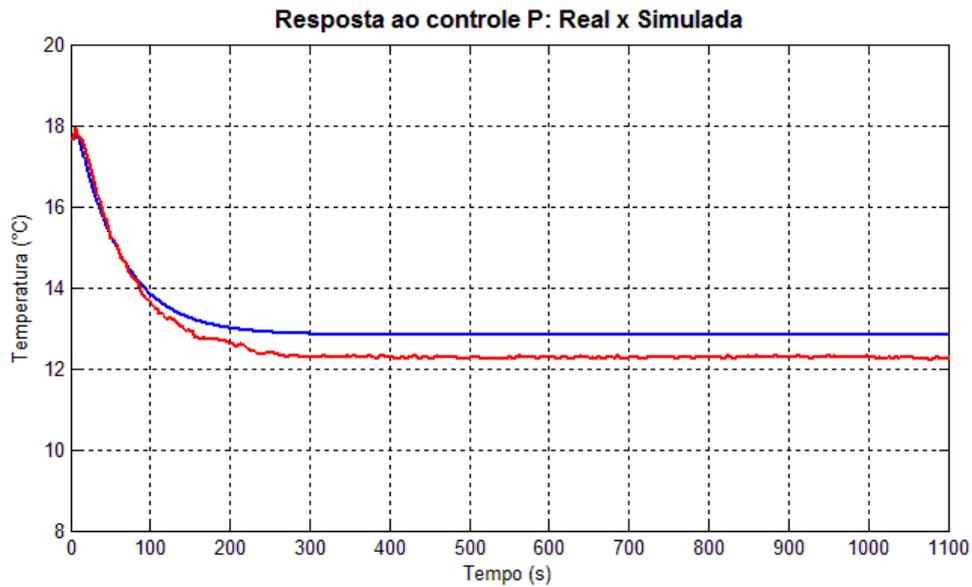
**Figura 4.7 - Saída do controlador e Erro na resposta ao controlador P no resfriamento**

Como esperado para um controle proporcional de um sistema de primeira ordem, a resposta apresenta um erro em regime permanente, neste caso um erro considerável de aproximadamente  $4,3^{\circ}\text{C}$ . Os valores obtidos neste experimento são confrontados com valores simulados através da ferramenta Simulink do Matlab, utilizando o diagrama de blocos observado na figura 4.8.



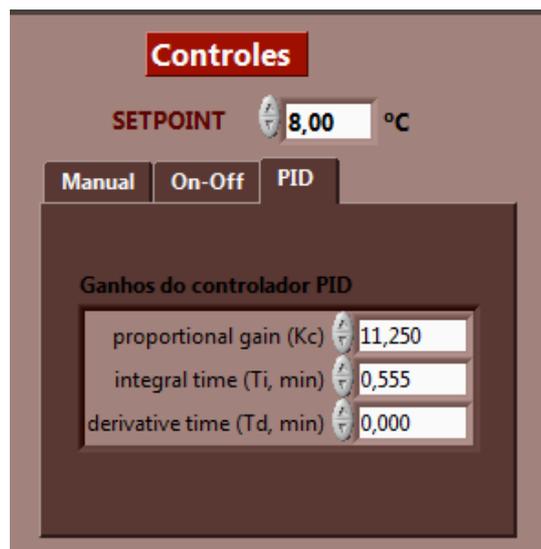
**Figura 4.8 - Diagrama de blocos do controle P**

A simulação realizada com diagrama de blocos no Simulink assume condições iniciais iguais a zero, ou seja o sistema estaria em equilíbrio a uma temperatura de  $0^{\circ}\text{C}$  antes da aplicação do degrau. Para poder realizar uma comparação adequada entre os resultados, uma constante com a temperatura inicial do experimento ( $17,68^{\circ}\text{C}$ ) deve ser adicionada. O gráfico contendo as duas respostas (real e simulada) é apresentado na figura 4.9.



**Figura 4.9 - Comparação entre as respostas ao controle P, real e simulada**

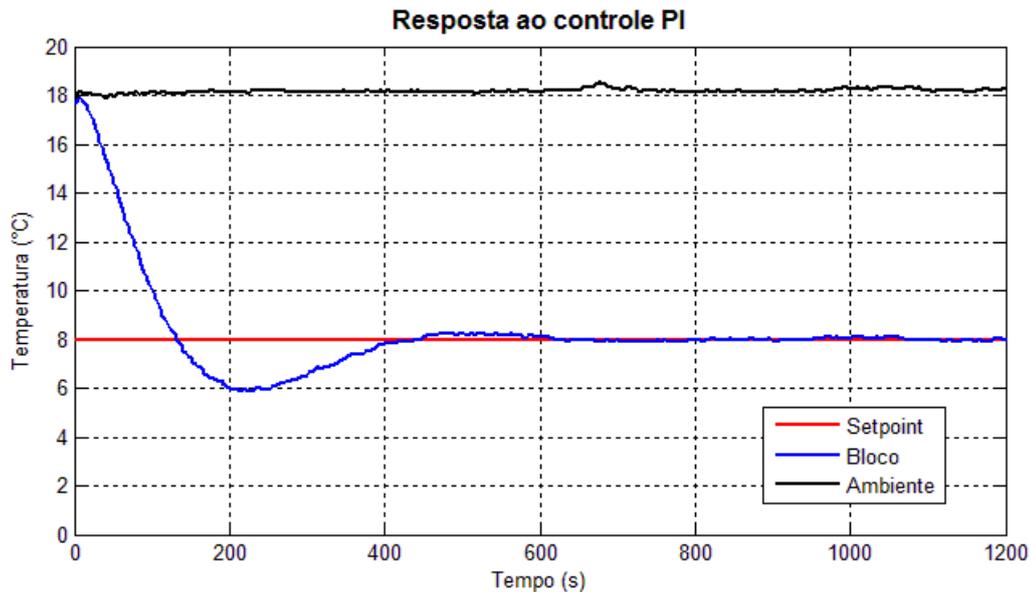
A resposta do experimento se mostrou semelhante à resposta obtida com a simulação, porém a resposta simulada apresentou um maior erro em regime permanente. Para eliminar o erro, se faz necessário adicionar uma ação integral ao controlador. Esta modificação pode ser vista na figura 4.10. Nela observam-se os valores utilizados para o ganho proporcional e o tempo integral. O LabVIEW trabalha com constantes de tempo para o controlador PID em minutos, então é necessário converter o valor de  $T_i$  calculado anteriormente em segundos.



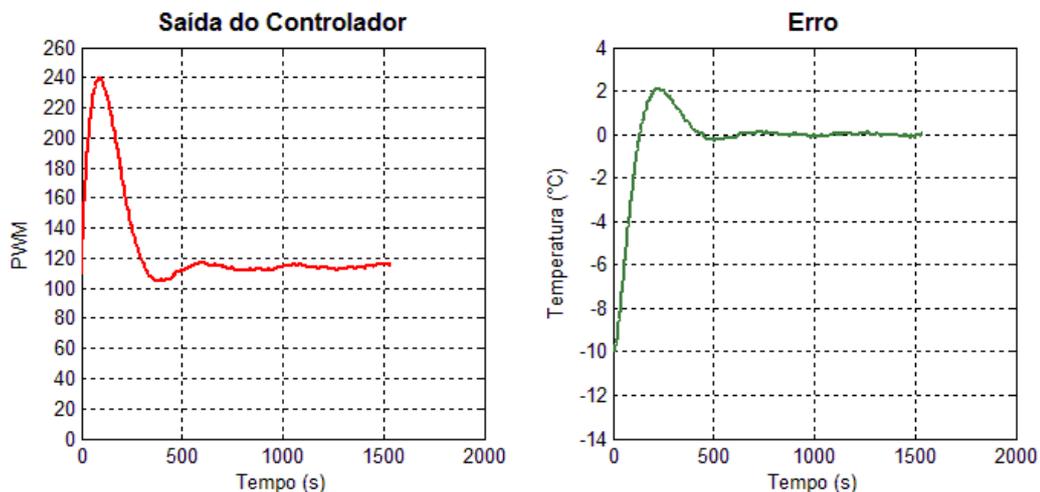
**Figura 4.10 - Configuração do controle PI no LabVIEW**

A resposta obtida com o controle PI pode ser observada nos gráficos das figuras 4.11 e 4.12. Neles, nota-se uma significativa diferença na resposta do sistema, atingindo um pico de

aproximadamente 6°C, que representa um *overshoot* de 25% do *setpoint* definido, e, como esperado, o erro do sistema tende a zero, levando a resposta ao valor desejado de 8°C em regime permanente, com uma oscilação bem pequena.

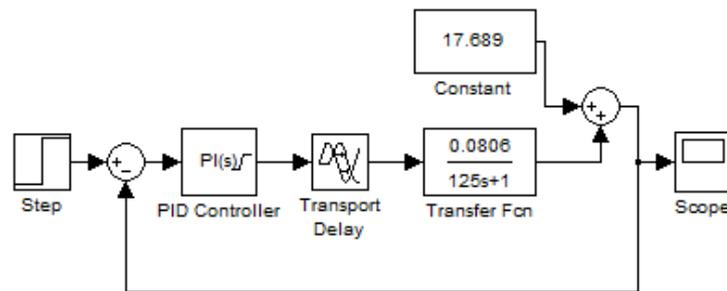


**Figura 4.11 - Resposta ao controle PI no resfriamento**



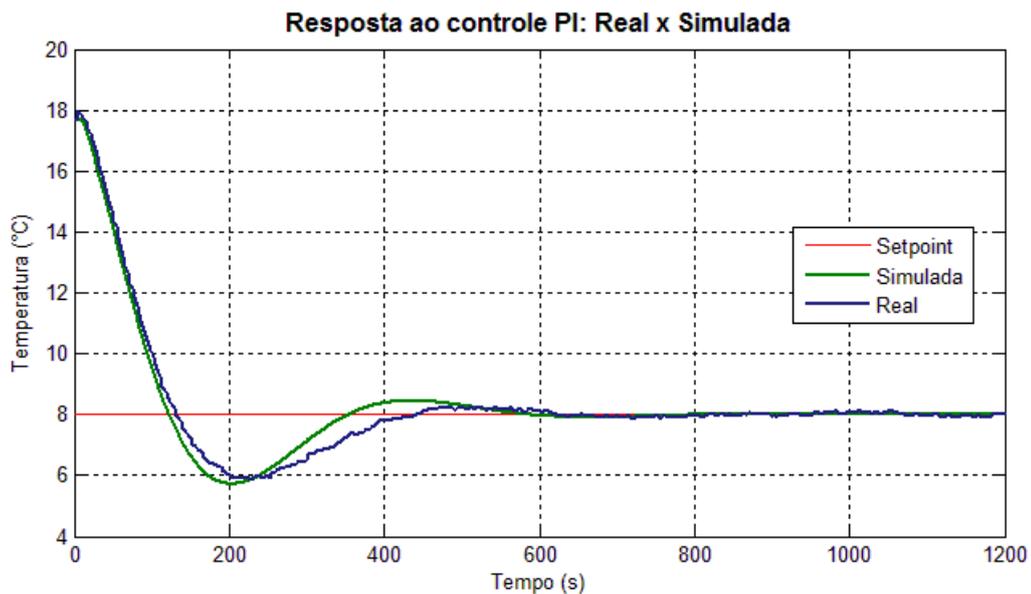
**Figura 4.12 - Saída do controlador e Erro na resposta ao controlador PI no resfriamento**

A resposta do sistema ao controle PI é simulada através do diagrama de blocos da figura 4.13. No controlador são aplicados os mesmos valores de ganhos utilizados no LabVIEW para o experimento, com uma pequena correção. O controlador PID do LabVIEW recebe os valores das constantes de tempo integral e derivativa em minutos, e no Simulink o PID utiliza o valor de ganho integrativo e derivativo, que correspondem ao ganho proporcional dividido pelas constantes de tempo em segundos. Ou seja,  $I = K_c/T_i$  e  $D = K_c/T_d$ .



**Figura 4.13 - Diagrama de blocos do controle PI**

Para poder ser comparado ao valor simulado, uma constante representando a temperatura inicial do sistema foi adicionada ao diagrama de simulação. A comparação entre a resposta simulada e a real pode ser observada na figura 4.14.



**Figura 4.14 - Comparação entre as respostas ao controle PI real e simulada**

As duas respostas (real e simulada) ao controle PI tiveram comportamentos razoavelmente semelhantes, apresentado o *overshoot* de aproximadamente 25% e atingindo o tempo de acomodação em instantes próximos.

Apesar da dinâmica do sistema ser bastante lenta e uma ação derivativa não ser essencial para o sistema, pois a resposta ao controle PI já se mostrou eficiente, o controle PID é testado com as configurações da figura 4.15, e o resultado do experimento pode ser visto nas figuras 4.16 e 4.17.

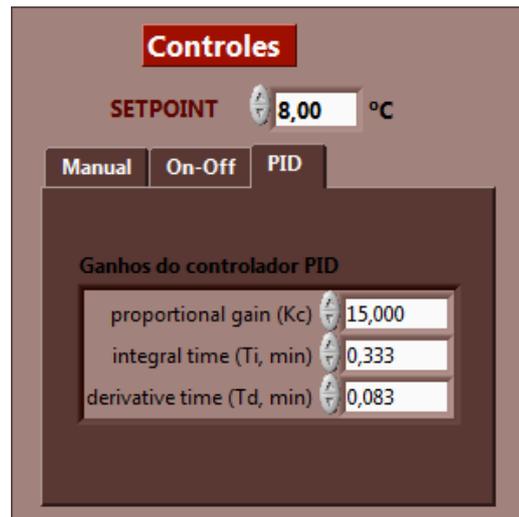


Figura 4.15 - Configuração do controle PID no LabVIEW

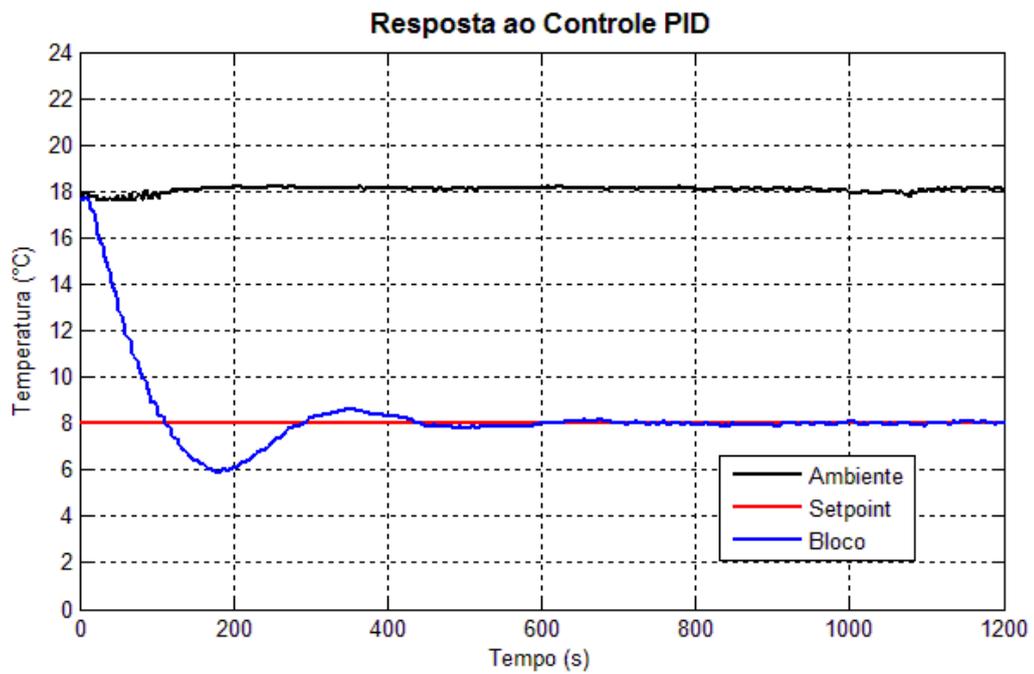
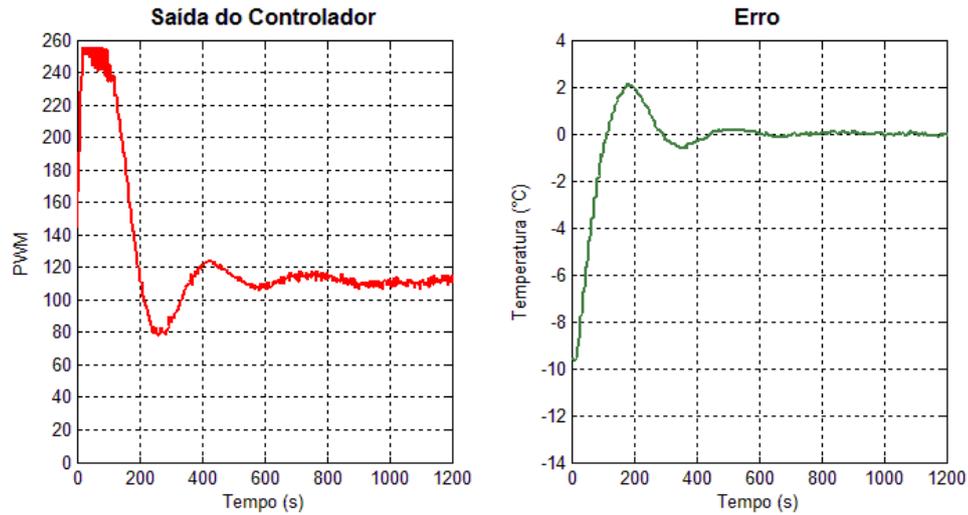
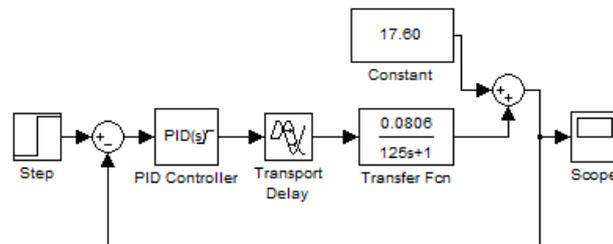


Figura 4.16 - Resposta ao controle PID no resfriamento



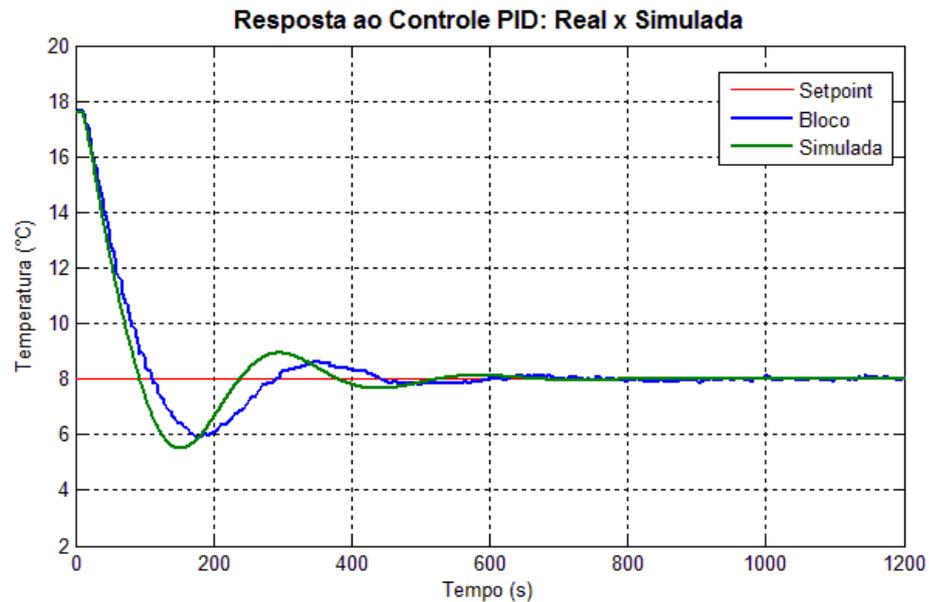
**Figura 4.17 - Saída do controlador e Erro na resposta ao PID no resfriamento**

O resultado do experimento com o controlador PID também é comparado com uma resposta simulada, para isso utiliza-se o mesmo diagrama de blocos do controle PI, porém com um controlador PID (figura 4.18).



**Figura 4.18 - Diagrama de blocos do controle PID**

Para esta comparação, assim como nas anteriores, é necessário normalizar os dados com a adição de uma constante equivalente a temperatura inicial do bloco no experimento. A comparação entre as respostas real e simulada pode ser observada na figura 4.19. Nota-se um comportamento semelhante entre as respostas, porém, a resposta simulada apresentou um tempo de resposta menor e um *overshoot* ligeiramente superior, comparado aos dados do experimento real.

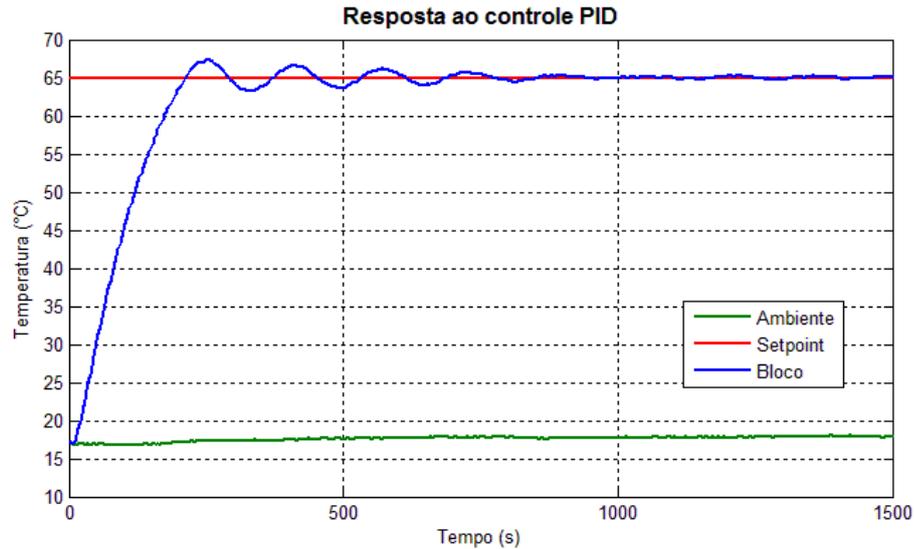


**Figura 4.19 - Comparação entre as respostas ao controle PID real e simulada**

### 4.3 Resposta ao controle PID no aquecimento

No intuito de demonstrar a versatilidade do funcionamento do módulo termoeétrico Peltier, foi realizado um experimento de aquecimento do bloco. Para isso foi necessário apenas inverter a polaridade da conexão do módulo com o driver de potência e alterar alguns sinais na logica do programa em LabVIEW. No caso do PID alteraram-se os valores de saturação de -255 a 0 no resfriamento para de 0 a +255, pois no aquecimento o controlador atua quando o erro é positivo.

Utilizando os ganhos para o controlador no aquecimento calculados anteriormente, ao se aplicar um *setpoint* de 65°C ao sistema obtém-se e a resposta apresentada na figura 4.20.

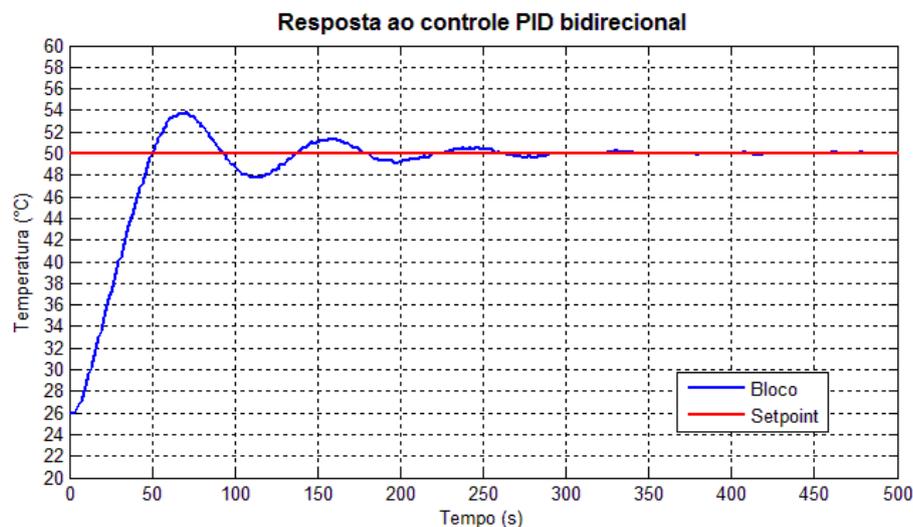


**Figura 4.20 - Resposta ao controle PID no aquecimento**

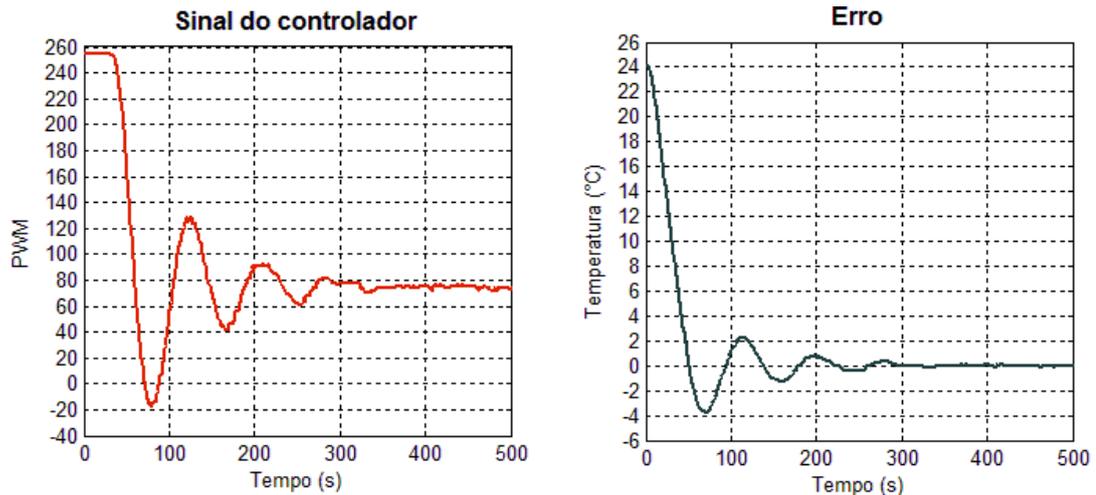
#### 4.4 Resultados prévios com o driver bidirecional

O objetivo inicial deste trabalho era realizar um controle bidirecional de temperatura, utilizando para isto um módulo Peltier e um *driver* bidirecional de corrente. Durante o desenvolvimento da segunda versão do *driver* bidirecional (seção 3.1.1.2) alguns testes foram realizados utilizando uma fonte simétrica. O resultado de um destes testes será apresentado neste capítulo no intuito de demonstrar o funcionamento de um controle bidirecional.

Para o teste é configurado um *setpoint* de 50°C e um controlador com ganhos  $K_c$  de 22,36 e  $T_i$  de 0,353. O resultado do experimento pode ser observado nas figuras 4.21 e 4.22.



**Figura 4.21 – Resposta ao controle PID bidirecional no aquecimento**



**Figura 4.22 - Saída do controlador e Erro na resposta ao PID bidirecional no aquecimento**

A diferença principal desta metodologia de controle quando comparada aos resultados obtidos na seção anterior está no fato de o sinal de saída do controlador poder assumir valores positivos e negativos, de forma que o sistema alcance o regime permanente em menos tempo. Este experimento teve como objetivo testar a funcionalidade do sistema no controle bidirecional. Para isso, os ganhos do controlador foram escolhidos empiricamente baseados em alguns testes. Neste experimento pode ser observado um comportamento satisfatório, com um tempo de acomodação menor. No entanto para uma comparação mais precisa entre os dois controles (bidirecional e unidirecional) seria interessante utilizar em ambos os experimentos um controlador calculado com o método Ziegler-Nichols e um mesmo *setpoint*.

## 5 CONCLUSÕES E CONSIDERAÇÕES FINAIS

O estudo prático das técnicas de controle permite aprofundar os conhecimentos sobre os processos estudados proporcionando um melhor entendimento do comportamento dos mesmos. Este estudo se justifica na ampla aplicabilidade das técnicas de controle de processos, que são utilizadas em equipamentos simples do dia a dia, equipamentos de precisão de laboratórios de pesquisa e em diversos processos industriais.

O controle de temperatura por efeito Peltier realizado neste trabalho faz uso de uma estrutura de baixo custo e fácil construção que permite a aplicação de técnicas de controle, principalmente o PID, de forma bastante intuitiva e eficaz. A utilização do sistema desenvolvido proporciona um grande aprendizado no que diz respeito à resposta de sistemas de primeira ordem aos controladores do tipo P, PI e PID, sendo possível observar a influência de cada ganho do controlador na planta.

A intenção original do projeto era explorar uma das principais diferenças na utilização de um módulo termoelétrico como atuador em um sistema de controle de temperatura, que seria a sua capacidade de trabalhar nas duas direções de troca de calor, aquecendo e resfriando o objeto proposto de acordo com o sentido da corrente fornecida ao módulo. Esta funcionalidade pôde ser testada durante o desenvolvimento do projeto e foi verificada uma resposta satisfatória, onde sinais de controle positivos e negativos foram aplicados dependendo da temperatura do bloco e do *setpoint* escolhido.

Em linhas gerais, os resultados obtidos foram muito satisfatórios, uma vez que foi possível aplicar as técnicas de controle propostas. O sistema funcionou conforme o previsto, fornecendo em tempo real a resposta da planta, o sinal de erro e a saída do controlador em gráficos que possibilitam acompanhar o comportamento do sistema ao longo do regime transiente.

A planta apresentou o comportamento desejado, fornecendo respostas condizentes com o esperado para um sistema com estas características e semelhantes às simulações realizadas.

Para aplicações futuras, visando tirar maior proveito da plataforma desenvolvida, deve-se buscar uma fonte de tensão simétrica adequada ao sistema, para que o controle bidirecional possa ser mais profundamente estudado. Outras modificações podem ser feitas no sentido de atingir respostas mais significativas do sistema, como a utilização de uma pastilha Peltier de potência mais elevada e a aplicação de mais de uma pastilha atuando em paralelo para obter

uma maior troca de calor. Para se estimar a FT do sistema de forma mais exata, outros métodos podem ser utilizados, além de observar o comportamento do sistema sujeito a diferentes degraus de temperatura. No intuito de aprimorar o sistema, vencendo as limitações encontradas no desenvolvimento do programa em LabVIEW, como o tamanho máximo do vetor de dados e o tempo de amostragem, seria interessante explorar outros ambientes de programação. Uma alternativa para o desenvolvimento de Interfaces Gráficas de Usuário é o próprio Matlab, que possui as ferramentas necessárias para a comunicação com o Arduino e possui funcionalidade semelhante à interface do LabVIEW. Por fim, uma modificação que ampliaria a área de estudo e os conhecimentos adquiridos com o sistema seria a inclusão de outros sensores para aferição da temperatura, de forma a proporcionar um comparativo entre os seus comportamentos e influências no sistema.

## REFERÊNCIAS

ARDUINO. **What is Arduino?** Disponível em: <<http://arduino.cc/en/Guide/Introduction>>.

Acesso em: 20 de Julho de 2014. a.

ARDUINO. **Arduino Mega 2560**. Disponível em:

<<http://arduino.cc/en/Main/ArduinoBoardMega2560>>. Acesso em: 20 Jul. 2014. b.

BAYER, F. M. & ARAÚJO, O. C. B. de, **Controle Automático de Processos**. Santa Maria-RS: e-Tec Brasil, 2012. 92 p.

BEGA, E. A. et al. **Instrumentação Industrial**, 2a edição. Rio de Janeiro: Editora Interciência, 2006.

BENNETT, J., BHASIN, A., GRANT, J., LIM, W. C. **PIDTuningClassical**. Disponível em: <[https://controls.engin.umich.edu/wiki/index.php/PIDTuningClassical#Ziegler-Nichols\\_Method](https://controls.engin.umich.edu/wiki/index.php/PIDTuningClassical#Ziegler-Nichols_Method)>. Acessado em: 15 Jul. 2015.

CUI INC. **Peltier Application Note**. Disponível em: <<http://www.cui.com/catalog/resource/peltier-app-note.pdf>>. Acesso em: 5 Jun. 2015.

DORF, R. C. & BISHOP, R. H. **Sistemas de controle modernos**. 8a Edição. Tradução: Bernardo Severo da Silva Filho, Rio de Janeiro: Livros Técnicos e Científicos Editora, 2001.

ESSICK, J. **Hands-on introduction to LabVIEW for Scientists and Engineers**. Oxford University Press, 2009.

FRADEN, J. **Handbook of modern sensors: physics, designs, and applications**, 3a edição. Nova Iorque: Springer, 2004.

HEBEI I.T., LTD. **Thermoelectric Cooler: TEC1-12706**. Rev 2.03. Disponível em: <<http://peltiermodules.com/peltier.datasheet/TEC1-12706.pdf>>. Acessado em: 04 Jun. 2015.

MATIAS, J. Teoria de Controle PID. **Revista Mecatrônica Atual**, Automação, Nº 3, p. 17-25, Abril, 2002.

MCKINNON, C. Commercial Bismuth Telluride-based Peltier Plates for Use as Heat Flux Transducers (A Concept). **Ecolibrium**, Austrália, Mai. 2010. Disponível em: <[http://www.airah.org.au/imis15\\_prod/Content\\_Files/EcoLibrium/2010/May/2010-05-01.pdf](http://www.airah.org.au/imis15_prod/Content_Files/EcoLibrium/2010/May/2010-05-01.pdf)>. Acessado em: 05 Jul. 2015.

NATIONAL INSTRUMENTS. **O que é Aquisição de Dados?** Disponível em: <<http://www.ni.com/data-acquisition/what-is/pt/>>. Acesso em: 20 Jul. 2014. a.

NATIONAL INSTRUMENTS. **Ambiente gráfico de desenvolvimento de sistemas LabVIEW** Disponível em: <<http://www.ni.com/labview/pt/>>. Acesso em: 20 Jul. 2014. b.

NATIONAL INSTRUMENTS. **Toolkits do NI LabVIEW.** Disponível em: <<http://sine.ni.com/nips/cds/view/p/lang/pt/nid/10447>>. Acesso em: 07 Jun. 2015.

NOVUS PRODUTOS ELETRÔNICOS LTDA, **Introdução ao controle PID.** Disponível em: <[http://www.das.ufsc.br/~aarc/ensino/posgraduacao/DAS6613/PID\\_Novus.pdf](http://www.das.ufsc.br/~aarc/ensino/posgraduacao/DAS6613/PID_Novus.pdf)>. Acesso em: 5 Jun. 2015.

OGATA, K. **Engenharia de Controle Moderno.** Tradução André Fábio Kohn, José Carlos Teixeira de Barros Moraes; Rio de Janeiro: Prentice Hall, 1985.

OGATA, K. **Ingeniería de Control Moderna.** 3ª Edição. Tradução Carlos Talancón Espinosa; Mexico: Pearson Educación, 1999.

OLIVEIRA, A. L. de L., et al. **Instrumentação: Fundamentos de Controle de Processo.** Espírito Santo: SENAI, 1999. 72 p.

PEREIRA, L. S. **Dopagem eletônica.** Disponível em: <<http://www.infoescola.com/quimica/dopagem-eletronica/>> Acesso em: 05 Jul. 2015.

SILVA, J. M. G. **Ação Liga-Desliga (On-off).** Disponível em: <<http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node20.html>>. Acesso em: 05 Jun. 2015.

ST MICROELECTRONICS. **LM158,A-LM258,A LM358,A: Low Power Dual Operational Amplifiers.** Jul. 2003. 12 p. Disponível em: <<http://pdf1.alldatasheet.com/datasheet-pdf/view/22762/STMICROELECTRONICS/LM358.html>>. Acesso em: Jun. 2014.

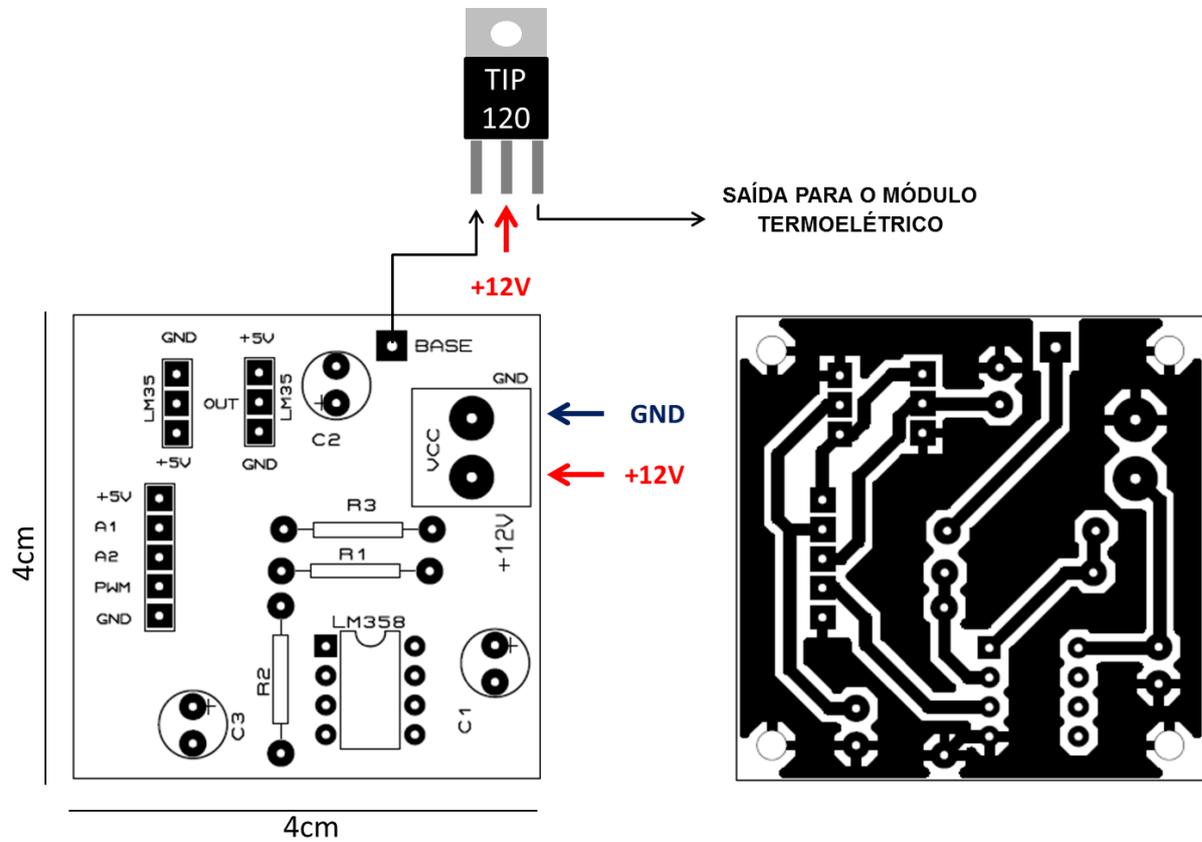
TEXAS INSTRUMENTS. **LM35 Precision Centigrade Temperature Sensors.** Ver. Jan. 2015. Disponível em:< <http://www.ti.com/lit/ds/symlink/lm35.pdf>>. Acesso em: 10 Jul. 2015.

WESTE, N. H. E & HARRIS, D. M. **CMOS VLSI Design: A Circuits and Systems Perspective.** 4ª edição. Boston, Massachusetts: Pearson Education, 2011. 840 p.

WENDLING, M. **Amplificadores Operacionais**. Versão 2.0. Disponível em: <<http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/3---amplificadores-operacionais-v2.0.pdf>>. Acesso em: Jun. 2014.

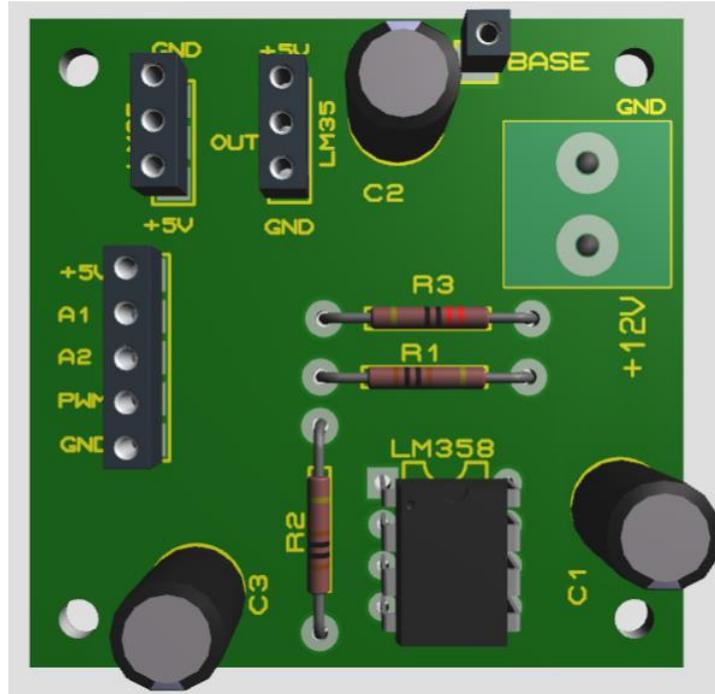
## APÊNDICE 1

Sugestão de montagem da placa de circuito impresso para o driver unidirecional.



**Figura A1.1 - Circuito do driver unidirecional**

Onde A1 e A2 são as conexões para os pinos de leitura analógica do Arduino e PWM para o pino de saída PWM do Arduino.



**Figura A1.2 - Placa de circuito impresso do driver unidirecional**

## APÊNDICE 2

### Diagrama de blocos do programa em LabVIEW

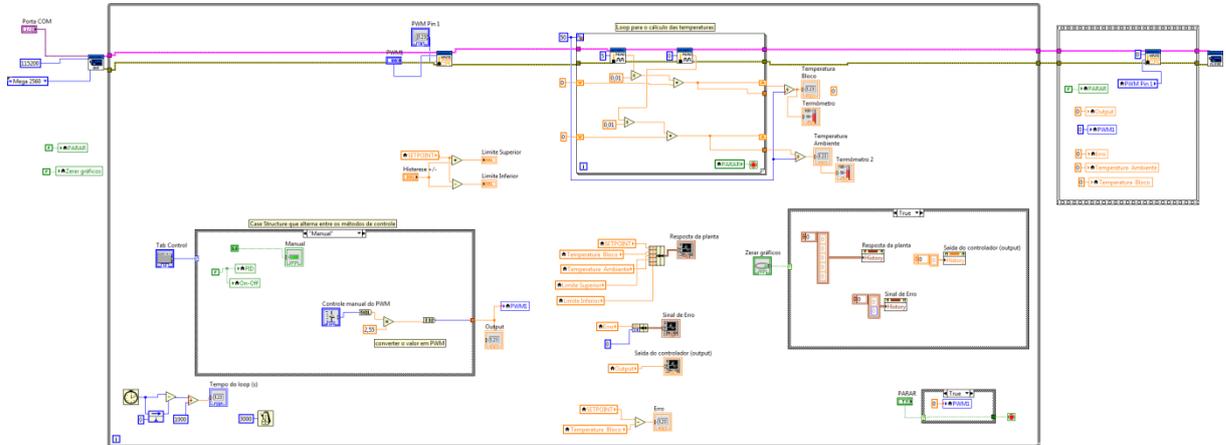


Figura A2.1 – Visão geral do diagrama de blocos

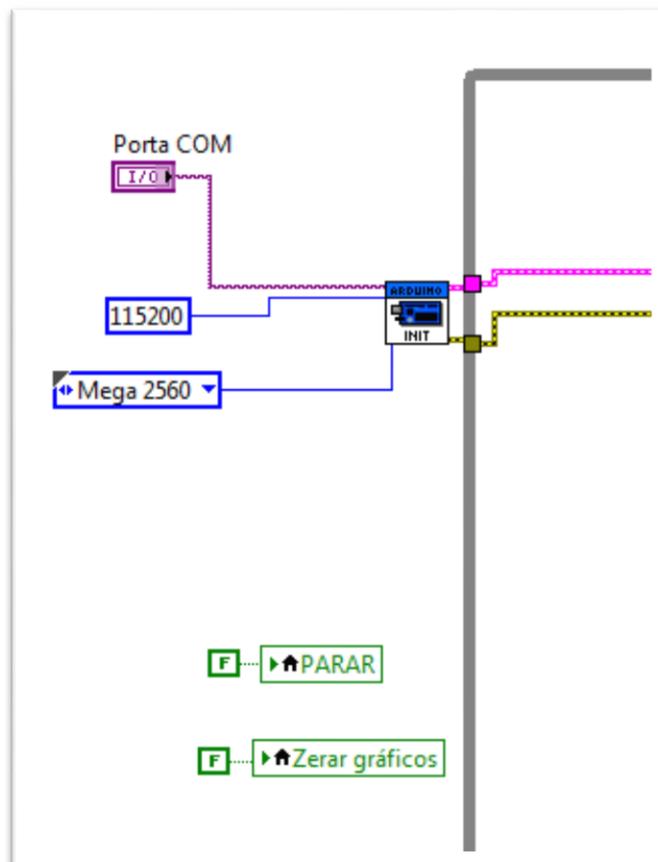


Figura A2.2 – Abertura da conexão com o Arduino

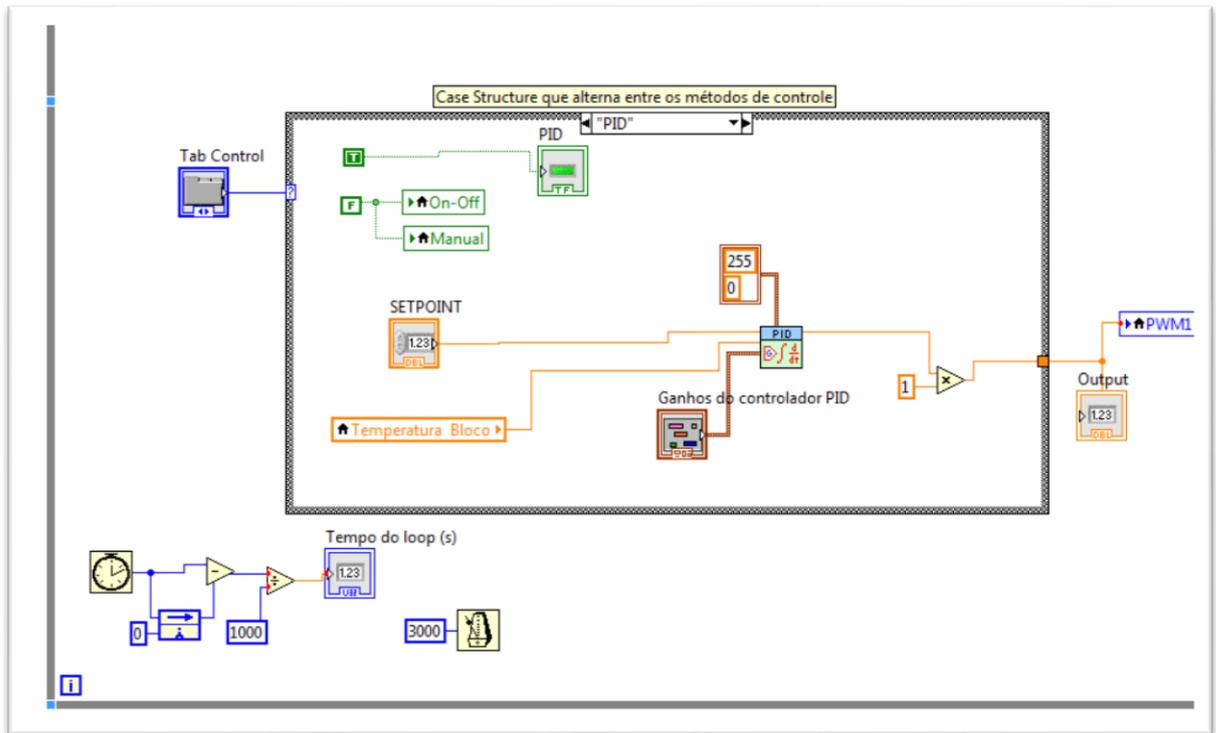


Figura A2.3 – *Case Structure* dos Métodos de Controle e Tempo de Execução do *Loop*

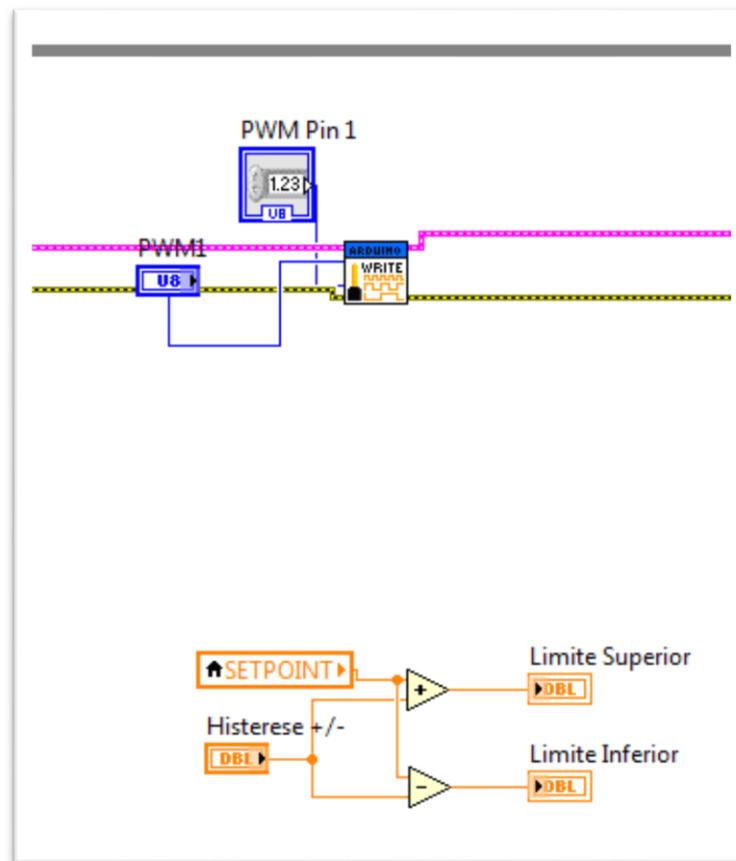


Figura A2.4 – Pino de saída PWM e Seleção dos LS e LI da Histerese

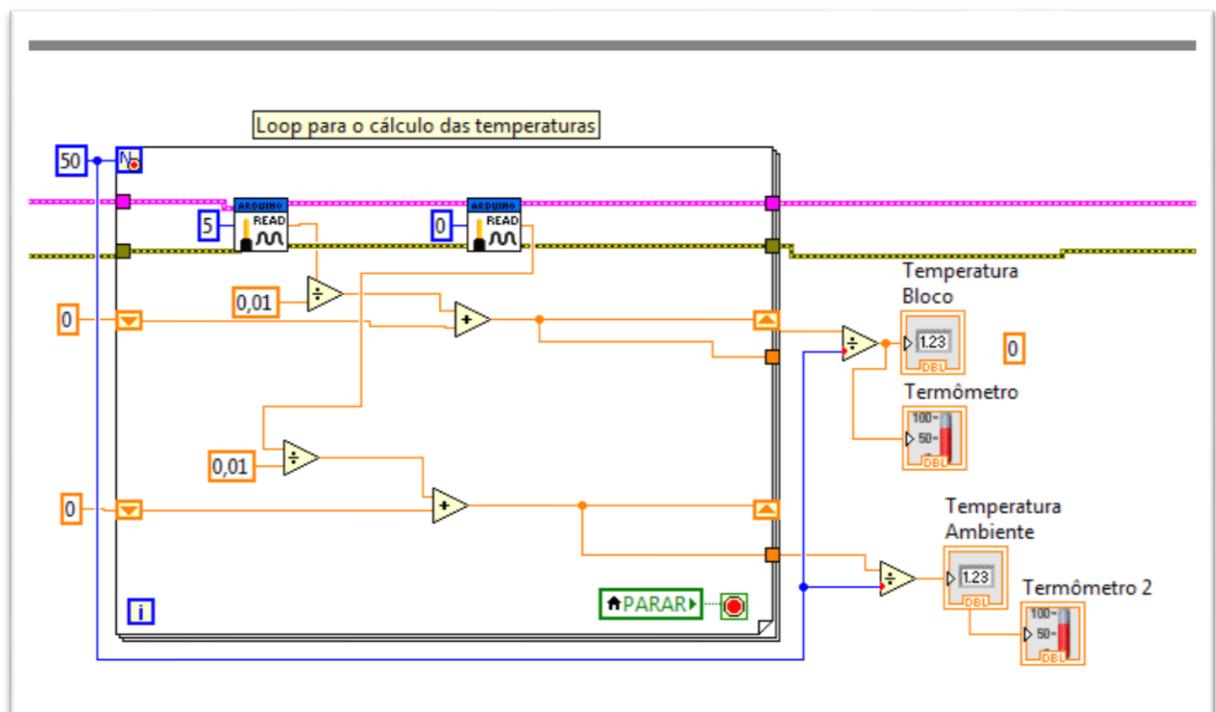


Figura A2.5 – Loop de aferição de temperatura

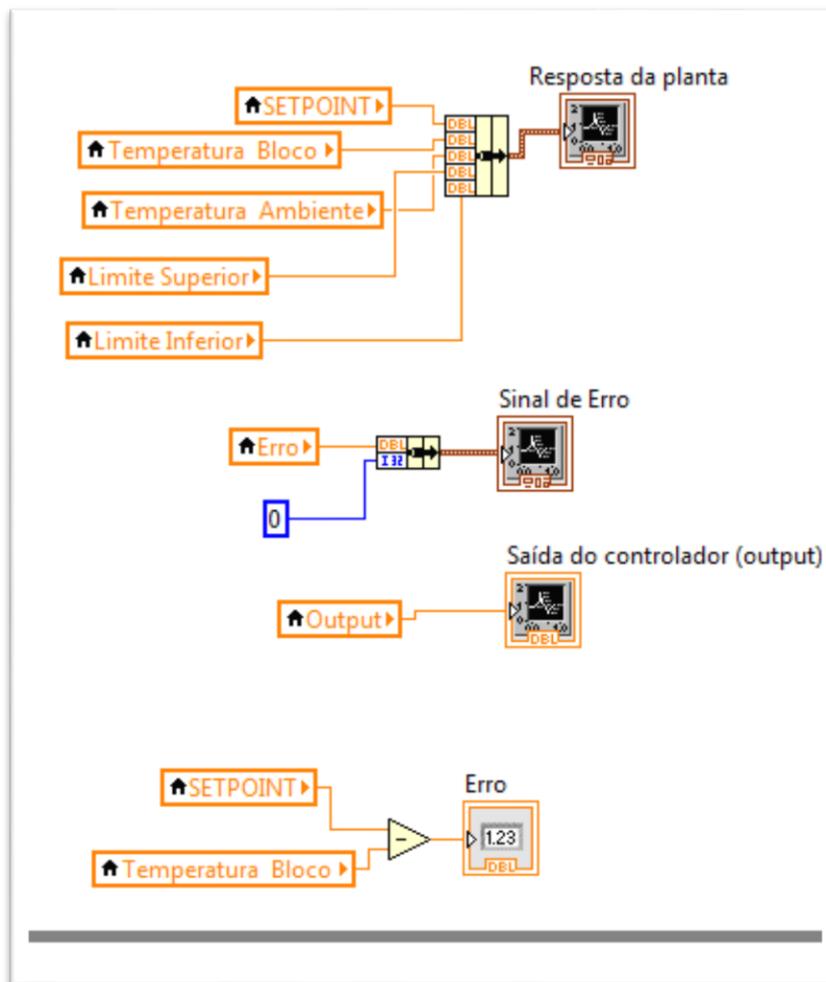


Figura A2.6 – Sinais mostrados nos gráficos



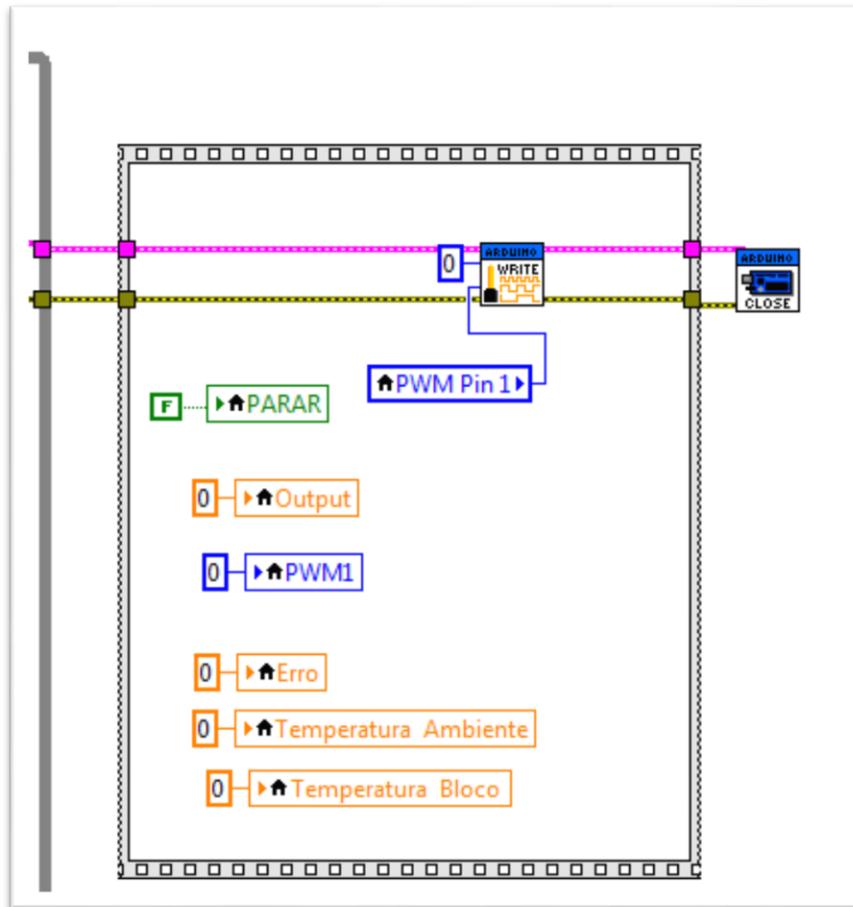


Figura A2.8 – Reinicialização das variáveis e encerramento da conexão com o Arduino