



UNIVERSIDADE FEDERAL DE OURO PRETO - UFOP

**ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA
DE CONTROLE E AUTOMAÇÃO - CEC AU**



DANIEL AUGUSTO SALLES CORDEIRO

**USO DO COMPENSADOR PID NO CONTROLE DE TEMPERATURA DE UM
FORNO ELÉTRICO A RESISTÊNCIA PARA SOLDAGEM DE COMPONENTES
SMD**

**OURO PRETO - MG
2016**

DANIEL AUGUSTO SALLES CORDEIRO
danielsalles_em@outlook.com

**USO DO COMPENSADOR PID NO CONTROLE DE TEMPERATURA DE UM FORNO
ELÉTRICO A RESISTÊNCIA PARA SOLDAGEM DE COMPONENTES SMD**

Monografia apresentada ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do título de Engenheiro de Controle e Automação.

Professor orientador: Prof. MSc. Adrielle de Carvalho Santana
Co-orientador: Diógenes Viegas Mendes Ferreira

OURO PRETO – MG
2016

C794u Cordeiro, Daniel Augusto Salles.
Uso do compensador PID no controle de temperatura de um forno elétrico a resistência para soldagem de componentes SMD [manuscrito] / Daniel Augusto Salles Cordeiro. - 2016.

89f.: il.: color; grafs; tabs.

Orientador: Prof. MSc. Adrielle de C. Santana.
Coorientador: Eng. Diógenes V. Mendes Ferreira.

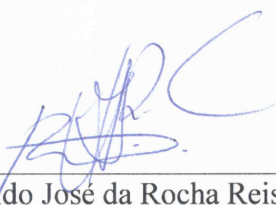
Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais.

1. Soldagem - Reflow. 2. Microcontroladores - PID. 3. Eletônica industrial - Forno - Resistivo. I. Santana, Adrielle de C.. II. Ferreira, Diógenes V. Mendes. III. Universidade Federal de Ouro Preto. IV. Título.

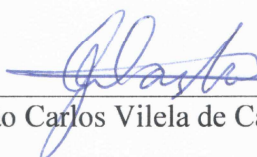
Monografia defendida e aprovada, em 05 de agosto de 2016, pela comissão avaliadora constituída pelos professores:



Prof. M. Sc. Adrielle de Carvalho Santana Silva - Orientadora



Prof. Dr. Agnaldo José da Rocha Reis – Professor Convidado



Prof. M. Sc. João Carlos Vilela de Castro – Professor Convidado

AGRADECIMENTO

Agradeço primeiramente a Deus, pela vida, saúde e sabedoria. Aos meus pais, Silvania e Sidney, por me ensinarem grandes valores e não medirem esforços para a realização deste sonho, amo vocês! Às minhas irmãs, Luísa e Duda, pelo amor e carinho. Aos meus avôs, pelas orações e exemplo de vida. As famílias Salles e Machado, em especial meus primos que torceram por mim. À Hortênsia, pelo amor e por estar sempre ao meu lado, obrigado! À UFOP/Escola de Minas/Professores, pelo ensino de qualidade em especial Adrielle, pela orientação, paciência e oportunidades. À Fundação Gorceix, pelo auxílio. A todos os amigos de Ouro Preto, em especial Diógenes Viegas e Felipe Tavares, pelo conhecimento compartilhado, auxílio em projetos e principalmente amizade. Obrigado a todos que de alguma maneira participaram de minha trajetória para torna este sonho possível.

“A melhor maneira de prever o futuro é cria-lo. ”

Peter Drucker

RESUMO

A necessidade de soldar componentes SMD é cada vez mais frequente. Muitos projetos eletrônicos necessitam de uma placa que utiliza componentes SMD, e sua solda tem que ter qualidade para o funcionamento correto do projeto. Além disso, é possível construir placas menores e mais baratas devido às dimensões e preços desses componentes. A soldagem de componentes convencionais é normalmente feita com um ferro de solda de baixa potência e uma solda muito fina. Porém, sua grande limitação é que os componentes a serem soldados têm de estar acessíveis na ponta do ferro de solda, sendo que com muitos componentes SMD isso é muito difícil de se conseguir ou até mesmo impossível para alguns deles devido às suas pequenas dimensões. Desta maneira surgiu a proposta da construção de um forno que consiga soldar de forma eficiente componentes SMD. O método utilizado para solda é conhecido como método de refluxo (ou em inglês, *reflow*), que consiste em aplicar uma fina camada de solda aos terminais da placa de circuito impresso, onde os pinos dos componentes SMD devem ser soldados. Após isso a placa é colocada no forno que irá seguir diferentes perfis de temperatura, resultando no final, em uma placa com solda de qualidade. O forno escolhido deve possuir uma câmara interna tão pequena quanto possível para caber as placas que se pretende soldar. Desse modo a temperatura irá ser elevada mais rápido, o que é extremamente importante na fase da soldagem. Na modelagem do forno, foram utilizadas diferentes tensões de alimentação e as curvas de temperatura foram obtidas com um termopar tipo K e o *software* LABVIEW. Assim foi possível, por meio do *software* MATLAB, modelar o forno e por meio desse modelo estimar os parâmetros do controlador PID, obtendo um controlador capaz de seguir o perfil de temperatura *reflow*. O forno foi desenvolvido no Laboratório de Protótipos e Desenvolvimento de Novas Tecnologias da Escola de Minas.

Palavras-chave: Microcontroladores; Soldagem Reflow; Forno Resistivo; Controle PID.

ABSTRACT

The need to weld SMD components is becoming increasingly often. Many electronic designs require a card that uses SMD components, and in order to the project to function properly, the quality of the component's weld must be seriously taken into account. Besides, it is possible to build smaller and cheaper plates due to the size and price of these components. The welding of conventional components is usually done with a low-power soldering iron and thin solder. However, it's major limitation is that the components to be welded must be accessible on the tip of the soldering iron, making, this very difficult to achieve or even impossible for some of the SMD components, because of their small size. Because of that, the construction of a furnace that can efficiently solder SMD components was proposed. The welding method used is known as reflow method, which consists of applying a thin layer of solder to the terminals of the printed circuit board, where the pins of the SMD components need to be welded. After this, the plate is placed in the furnace that will switch into several different temperature profiles, leading to a plate with good quality welding. The furnace was chosen due to being as small as possible to fit the plates intended to be welded, thus the temperature would rise faster, which is extremely important in the welding phase. To modeling of the oven, different supply voltages were used, the temperature curves were obtained with a type K thermocouple and LABVIEW software. Therefore, it was possible, with the MATLAB software, evaluate the model of the oven and by means of this model, estimate the parameters of the PID controller, obtaining a controller capable of following the reflow temperature profile. The oven was developed in the Prototypes and Development Laboratory of New Technologies in the School of Mines of Ouro Preto.

Keywords: Microcontroller; Reflow soldering; Resistive oven; PID control.

LISTA DE FIGURAS

Figura 2.1 – Comparação entre capacitor convencional e SMD (a); PIC18F4550 convencional e sua versão SMD (b); resistência convencional e SMD (c).	13
Figura 2.2 – Perfil de aquecimento Reflow.	15
Figura 2.3 – Arquitetura Von- Neumann.	17
Figura 2.4 – Arquitetura Harvard.	18
Figura 2.5 – Resposta ao degrau de um sistema de primeira ordem.	22
Figura 2.6 – Resposta ao degrau de um sistema de primeira ordem com atraso.	23
Figura 2.7 – Diagrama de blocos de um sistema de controle em malha fechada.	29
Figura 2.8 – Temperatura oscilando em torno do <i>set point</i>	30
Figura 2.9 – Características da ação de controle ON/OFF.	31
Figura 2.10 – Ação proporcional.	32
Figura 2.11 – Sistema de controle digital.	37
Figura 2.12 – Curva de resposta a um degrau.	40
Figura 2.13 – Resposta do sistema ganho crítico.	42
Figura 2.14 – Leitura de um termopar tipo j.	45
Figura 2.15 – Modelo simplificado de leitura de termopar.	45
Figura 3.1 – Bancada de trabalho.	47
Figura 3.2 – (a) Placa amplificador de tensão; (b) Projeto no AutoCAD.	48
Figura 3.3 – USB-6009.	50
Figura 3.4 – (a) Interface de programação; (b) Interface Visual.	51
Figura 3.5 – Cálculo do ponto de inflexão para tensão de 20.	53
Figura 3.6 – Resultado experimental vs modelo matemático para as tensões de 20 a 60 (a); 80 a 100 (b); 120 a 180 (c).	56
Figura 3.7 – Esquema elétrico microcontrolador.	57
Figura 3.8 – (a) Circuito de Controle; (b) Circuito de Potência.	58
Figura 3.9 – (a) Circuito elétrico do LCD.	59
Figura 3.10 – Projeto da PCI feito no Ares – Proteus.	61
Figura 3.11 – Placa controladora.	62
Figura 3.12 – Diagrama de blocos do controlador PI em malha fechada.	62
Figura 4.1 – Comparação entre o controle PI e PID.	64
Figura 4.2 – Perfil de temperatura obtido com o controlador PID.	65

LISTA DE TABELAS

Tabela 2.1 – Valores das temperaturas para o perfil Reflow.....	15
Tabela 2.2 – Tamanho do encapsulamento dos componentes.....	16
Tabela 2.3 – Comparação entre as Arquiteturas CISC e RISC.	18
Tabela 2.4 – Exemplo de microcontroladores com 12,14 e 16 Bits.	19
Tabela 2.5 – Efeitos no sistema das ações proporcional, integral e derivativo.	35
Tabela 2.6 – Calculo dos ganhos PID (primeiro método).	41
Tabela 2.7 – Tabela para cálculos dos ganhos PID (segundo método).	42
Tabela 2.8 – Termopares	44
Tabela 3.1 – Componentes placa amplificadora de tensão.....	49
Tabela 3.2 – Dados para modelagem do sistema.	53
Tabela 4.1 – Comparação entre perfil reflow recomendado e obtido.	66

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivo Geral.....	11
1.2	Objetivos Específicos	11
1.3	Justificativa do Trabalho.....	11
1.4	Metodologia	12
1.5	Estrutura do Trabalho	12
2	REVISÃO DA LITERATURA	13
2.1	Tecnologia SMD	13
2.1.1	<i>Perfil de soldagem reflow</i>	13
2.2	Microcontroladores	16
2.2.1	<i>Arquitetura dos microcontroladores</i>	16
2.2.2	<i>Microcontroladores PIC</i>	19
2.2.3	<i>Conversor Analógico Digital</i>	20
2.2.4	<i>Portas de entrada e saída (I/O)</i>	20
2.2.5	<i>Interrupções</i>	21
2.3	Controle de tensão por corte do ângulo de fase	21
2.4	Modelagem de sistemas de primeira ordem	22
2.4.1	<i>Identificação de um sistema de primeira ordem com atraso</i>	22
2.4.2	<i>Aproximação de Padé</i>	24
2.4.3	<i>Método dos mínimos quadrados</i>	26
2.5	Controladores PID	29
2.5.1	<i>Controle de temperatura</i>	29
2.5.2	<i>Ação de controle liga / desliga (ON/OFF)</i>	30
2.5.3	<i>Ação proporcional</i>	31
2.5.4	<i>Ação proporcional-integral</i>	32
2.5.5	<i>Ação proporcional derivativa</i>	33
2.5.6	<i>Ação proporcional integral derivativa</i>	34
2.5.7	<i>Controle Digital</i>	35
2.5.8	<i>Implementação Digital do Controlador PID</i>	36
2.6	Métodos práticos de sintonia de controle PID	39

2.6.1	<i>Método da resposta ao degrau</i>	40
2.6.2	<i>Método do ganho crítico</i>	41
2.7	Medição de temperaturas	42
2.7.1	<i>Termopares</i>	42
2.7.2	<i>Compensação da junta de referência</i>	44
2.7.3	<i>Linearização de dados</i>	45
3	METODOLOGIA	47
3.1	Projeto do sistema de aquisição de temperatura	48
3.2	Aquisição das temperaturas	51
3.3	Identificação dos parâmetros do forno	52
3.4	Validação do modelo	54
3.5	Projeto do controlador	56
3.5.1	<i>Firmware do controlador</i>	59
3.5.2	<i>Placa de Circuito impresso</i>	60
3.5.3	<i>Parâmetros PID do controlador</i>	62
4	RESULTADOS	64
4.1	Teste dos controladores	64
4.2	Perfil reflow projetado	65
5	CONCLUSÕES E RECOMENDAÇÕES	67
5.1	Conclusões	67
5.2	Recomendações para trabalhos futuros	67
	REFERÊNCIAS BIBLIOGRÁFICAS	68

1 INTRODUÇÃO

A necessidade de medir a temperatura é ponto de interesse da ciência há muitos anos, e foi objeto de estudo de grandes cientistas como Claudius Galenus, Galileu Galilei, Fernando II – Grão Duque da Toscana, entre outros, que se preocuparam em construir equipamentos que medissem com precisão a mesma. Com o avanço da tecnologia, o interesse em controlar a temperatura se tornou cada vez maior, visto que ela é uma das variáveis mais usadas na indústria de controle de processos e em seus diversos segmentos. A temperatura é uma grandeza básica para medição e controle de vazão, densidade, entre outros parâmetros.

Na área da eletrônica, umas das aplicações que utilizam o controle preciso da temperatura é a confecção de placas que possuem microcomponentes, também conhecidos como componentes SMD (*Surface-Mount Device*). A etapa que necessita de maior atenção em relação aos parâmetros de temperatura é durante a solda destes componentes que é realizada normalmente em um forno elétrico.

Basicamente, o controle de temperatura de um forno elétrico é dividido nas seguintes etapas: (i) detecção da temperatura do forno por um sensor de temperatura; (ii) comparação da temperatura medida com o valor desejado (valor de referência); e (iii) a partir desta comparação, utilizando um controlador, gera-se um sinal de controle para um dispositivo que irá controlar a tensão que o forno irá receber. Outro fator importante é o perfil de temperatura, que tem que seguir um determinado modelo composto de diversas etapas. Um modelo de referência é a soldagem reflow, que possui 4 etapas, sendo elas: pré-aquecimento, aquecimento, derretimento dos grãos de solda e arrefecimento do conjunto.

Neste trabalho, foi utilizado um forno elétrico resistivo com o intuito de soldagem de componentes SMD. Por não precisar atender uma demanda em escala industrial, o forno utilizado é constituído basicamente de uma câmara de aquecimento, um conjunto de resistências elétricas, uma carcaça metálica e tem uma capacidade de 8l. A câmara de aquecimento tem um papel importante, uma vez que ela determina as perdas de calor, enquanto que as resistências elétricas determinam a temperatura máxima de operação do forno elétrico.

Uma das dificuldades encontradas quando se trabalha com fornos elétricos é encontrar um modelo matemático que tenha um comportamento suficientemente próximo do sistema real. No caso de

um forno resistivo, sabe-se que ele pode ser modelado como um sistema de primeira ordem com atraso. No entanto, sabe-se também que seus parâmetros variam de acordo com a tensão aplicada no forno.

Desta forma, utilizando um sensor de temperatura, diversos parâmetros são adquiridos dos diferentes perfis de temperatura que o forno adquire. E com o auxílio de ferramentas matemáticas, como o Matlab, é obtido um modelo para o forno. Para concluir o projeto, foi projetado um controlado do tipo PID para seguir o modelo de *soldagem reflow*.

1.1 Objetivo Geral

O principal objetivo deste projeto é adaptar um forno resistivo para atender ao perfil de soldagem *reflow*, de modo que seja possível realizar a soldagem de componentes SMD com qualidade.

1.2 Objetivos Específicos

- Desenvolver um sistema de solda SMD de baixo custo;
- Projetar um sistema de controle de temperatura para a implementação do perfil *reflow* no forno resistivo.

1.3 Justificativa do Trabalho

Os laboratórios de Automação da Escola de Minas, possuem como equipamentos para soldagem de componentes SMD a estação de solda convencional e a vapor. Porém, estes dois equipamentos apresentam algumas limitações como a ponta do ferro inacessível em alguns componentes SMD, no caso de utilizar a estação de solda convencional e a restrição de soldar pino por pino, quando se utiliza a estação de solda a vapor. Além disso, o valor de compra de um forno SMD é muito elevado quando comparado com a construção de um. Assim, com o desenvolvimento de um forno espera-se incentivar a utilização de componentes SMD para construção de placas de circuito impresso (PCI), nos diversos projetos de pesquisa na área de inovação e tecnologia da Universidade Federal de Ouro Preto (UFOP). Pelo baixo custo de construção e operação do forno, juntamente com a viabilidade de investimento em componentes SMD, poderão ser fabricadas PCI's menores e com maior qualidade.

1.4 Metodologia

- Configurar uma plataforma para a aquisição de dados dos sensores de temperatura utilizados;
- Realizar a modelagem do forno, identificando a função de transferência que representa o sistema em malha aberta;
- Projetar e sintonizar um controlador PID, com ênfase em métodos empíricos.
- Verificar a viabilidade da utilização do microcontrolador PIC18F4550 e implementar o controlador PID projetado em seu *firmware*.
- Desenvolver uma placa de circuito impresso contendo o sistema para controlar a temperatura do forno.

1.5 Estrutura do Trabalho

Este trabalho está dividido em cinco capítulos. O capítulo um contextualiza o trabalho, apresentando seus principais objetivos e justificativa. No capítulo dois apresenta-se o referencial teórico da proposta, explicando o perfil de soldagem desejado, as vantagens da utilização dos componentes SMD, o microcontrolador utilizado e seus principais módulos, alguns conceitos de modelagem e teoria de controle que embasam o desenvolvimento desse projeto e alguns tipos de controladores e suas particularidades. O capítulo três apresenta a metodologia utilizada no desenvolvimento do trabalho em questão, com foco no projeto e montagem do sistema de controle desde a aquisição das temperaturas para a modelagem do sistema até o projeto do controlador, abordando seu funcionamento. No capítulo quatro são apresentados os resultados experimentais, e por fim, no último capítulo são tratadas as conclusões e considerações finais.

2 REVISÃO DA LITERATURA

2.1 Tecnologia SMD

SMD são as iniciais em inglês das palavras “*Surface-Mount Device*”. Pode-se encontrar siglas como SMT (*Surface-Mount Technology*) ou SMC (*Surface-Mount Component*). Todos estes termos têm significado semelhante e são relativos ao processo de montagem de um componente eletrônico.

O surgimento da tecnologia SMD foi um significativo evento na eletrônica desde o advento do circuito integrado. O SMD é definido como o posicionamento e fixação (soldagem) de componentes eletricamente passivos ou ativos na superfície de uma Placa de Circuito Impresso.

A utilização desses componentes apresenta muitas vantagens sobre os componentes convencionais, entre eles: miniaturização da placa, velocidade de resposta ampliada do circuito, desenvolvimento da automação industrial e redução de custos. Nas figuras 2.1 (a), (b) e (c) pode-se ver algumas comparações entre componentes convencionais e componentes SMD (indicados pelas setas).

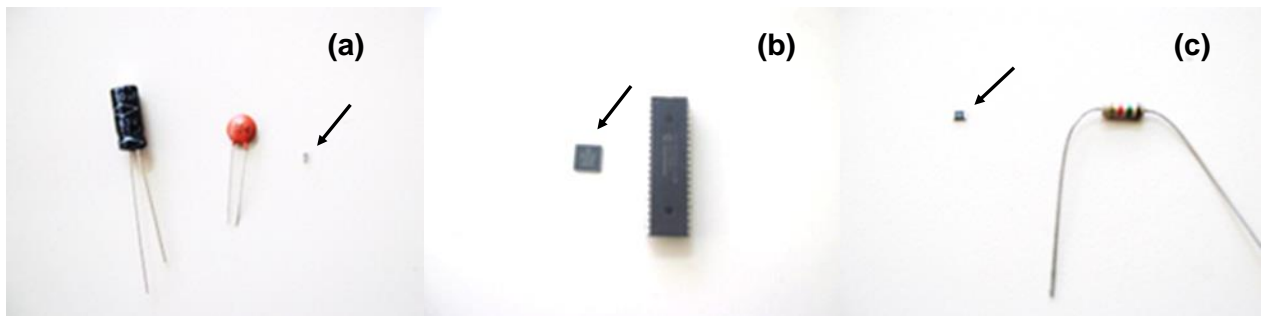


Figura 2.1 –Comparação entre capacitor convencional e SMD (a); PIC18F4550 convencional e sua versão SMD (b); resistência convencional e SMD (c).

Fonte: Pesquisa direta (2016)

2.1.1 Perfil de soldagem reflow

O perfil de soldagem *reflow* se difere do processo convencional, que apresenta componentes com terminais e que são inseridos através de furos na PCI, principalmente porque não há interferência mecânica em níveis similares ao sistema de corte e climpagem dos terminais. Neste processo a placa inteira é aquecida, fazendo a solda passar por cinco etapas de temperatura.

O processo começa com a fase de pré-aquecimento, onde a temperatura no forno é incrementada até 125 °C. A esta temperatura o fluxo presente na solda, torna-se líquido. O excesso de fluxo escorre e afasta-se dos terminais de soldadura, deixando os grãos de solda no local devido. A

temperatura sobe então de forma relativamente lenta até aos 175 °C, temperatura próxima do ponto de fusão dos grãos de solda. A razão para o incremento lento é que a placa e os componentes precisam de tempo para ficarem todos na mesma temperatura. Esta fase é designada de liquefação. Uma vez alcançada essa temperatura, o forno deve aquecer a placa e os componentes até a temperatura máxima (normalmente 220-240 °C). Durante esta fase os grãos de solda derretem e ligam-se ao metal circundante (GOOSENS, 2006).

Após a temperatura máxima ser alcançada, é preciso arrefecer todo o conjunto. Esta fase tem o nome de arrefecimento. Contudo, este arrefecimento não deve ser muito rápido, para evitar diferenças de temperatura muito grandes entre os componentes e a placa, que podem deformar os componentes ou mesmo quebrá-los. Por outro lado, este arrefecimento não deve ser muito lento, especialmente no início, pois alguns componentes podem apenas permanecer acima de uma dada temperatura crítica por um intervalo de tempo (GOOSENS, 2006).

A figura 2.2 mostra o perfil de temperatura que deve ser seguido, respeitando as restrições da tabela 2.1 e tabela 2.2.

As pastas de solda podem ser caracterizadas em dois grupos, sendo elas, as soldas baseadas em chumbo e as que não contém chumbo. Na maioria das soldas baseadas em chumbo a liga é composta por estanho e chumbo (SnPb). Esta liga tem o seu ponto de fusão nos 183°C. O segundo grupo, o das pastas sem chumbo, consiste normalmente numa liga de estanho, prata e cobre (SnAgCu). Por não ter chumbo, esta liga normalmente só começa a derreter a partir dos 217°C (GOOSENS, 2006). Na tabela 2.2 é possível ver a diferença entre os perfis de temperatura para estes dois tipos de solda. É importante estar atento ao tempo máximo entre a temperatura ambiente e a temperatura de pico que é de 6 minutos para a solda utilizada (Última linha da tabela 2.1).

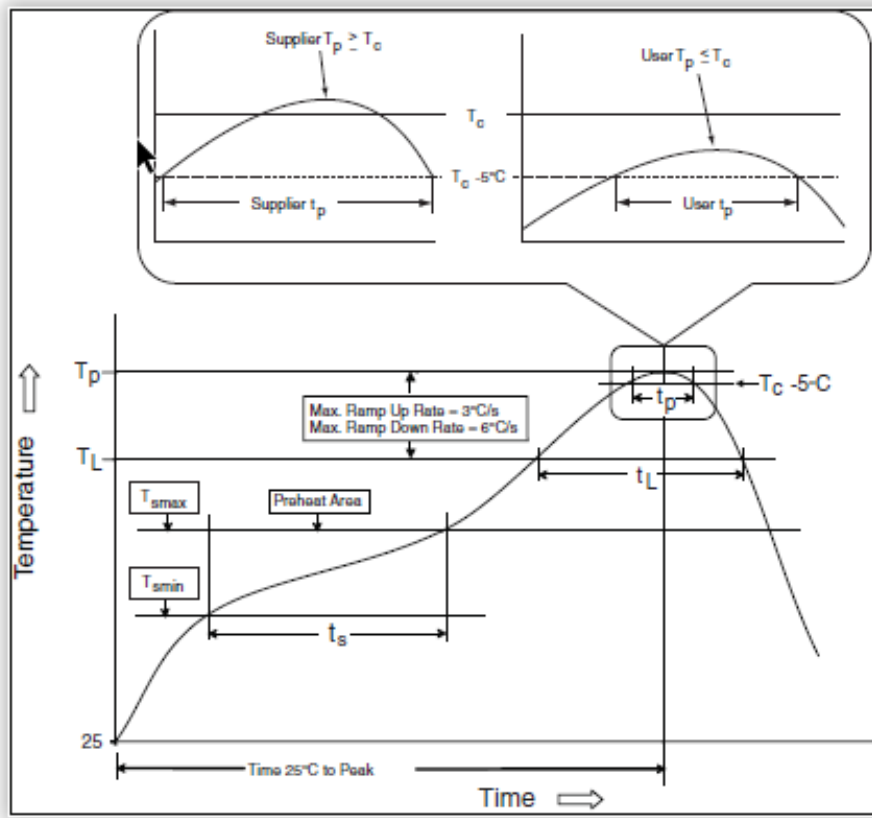


Figura 2.2 – Perfil de aquecimento Reflow.

Fonte: Altera (2011).

Tabela 2.1 – Valores das temperaturas para o perfil Reflow.

Profile Feature	Sn-Pb Eutectic Assembly	Pb-Free Assembly
Average ramp-up rate (T_{sMAX} to T_p)	3°C/ second maximum	3°C/ second maximum
Minimum preheat temperature (T_{sMIN})	100°C	150°C
Maximum preheat temperature (T_{sMAX})	150°C	200°C
Preheat time (T_{sMIN} to T_{sMAX}) (t_s)	60–120 seconds	60–120 seconds
Temperature (T_L)	183°C	217°C
Time above liquidous temperature T_L (t_L)	60–150 seconds	60–150 seconds
Peak temperature (T_p)	Refer to Table 2.2	Refer to Table 2.2
Average ramp-down rate (T_p to T_{sMAX})	6°C/ second maximum	6°C/ second maximum
Time 25°C to T_p	6-minutes maximum	8-minutes maximum

Fonte: Altera (2011) (Adaptado).

Tabela 2.2 – Tamanho do encapsulamento dos componentes.

Package Thickness	Volume mm ³ < 350	Volume mm ³ ≥ 350
< 2.5 mm	235°C	220°C
≥ 2.5 mm	220°C	220°C

Fonte: Altera (2011).

2.2 Microcontroladores

Os microcontroladores são pequenos dispositivos eletrônicos dotados de uma “inteligência” programável, utilizados no controle de processos lógicos. Basicamente são constituídos de uma CPU, memória e periféricos. Suas dimensões reduzidas são resultantes da alta capacidade de integração em que milhões de componentes são inseridos em uma única pastilha de silício pela técnica de circuitos integrados (MIYADAIRA,2009).

Segundo Souza *et al.* (2005), o microcontrolador é composto por periféricos, tais como: LED's, botões, display's de segmentos, display's de cristal líquido (LCD), resistências, relés, sensores diversos (pressão, temperatura, etc.) e muitos outros. São chamados de controles lógicos, pois a operação do sistema baseia-se nas ações lógicas que devem ser executadas, dependendo do estado dos periféricos de entrada e/ou saída. A “inteligência” do componente se deve à Unidade Lógica Aritmética (ULA), que realiza todas as operações matemáticas e lógicas quando executadas.

O microcontrolador é programável, ou seja, toda a lógica de operação é estruturada em um tipo de linguagem de programação e gravada dentro do componente. Desta maneira, toda vez que o microcontrolador for alimentado o programa interno será executado.

Por terem tamanho reduzido, baixo custo, e baixo consumo de energia aliados a facilidade de desenvolvimento de aplicações, os microcontroladores são amplamente utilizados na automação industrial e na construção de aparelhos eletrônicos em geral, tais como, celulares, MP3 players, impressoras, aplicações de robótica, instrumentação entre outros.

2.2.1 Arquitetura dos microcontroladores

Todos os microcontroladores modernos usam uma de duas arquiteturas: Harvard ou von-Neumann. Em suma, são duas formas distintas de trocar informações entre a CPU (Unidade Central de Processamento) e a memória. Grande parte dos microcontroladores tradicionais apresenta a

arquitetura do tipo von-Neumann que utilizam uma zona de memória e um barramento de 8 bits. Como todos os dados são trocados, usando estas 8 linhas, este barramento é sobrecarregado e a comunicação fica extremamente lenta e ineficiente. O CPU pode ler uma instrução, ou ler/escrever dados de/para a memória. Ambas as situações ao mesmo tempo são impossíveis, pois as instruções partilham o barramento com os dados (Sena, s.d). Esta configuração é ilustrada na figura 2.3:

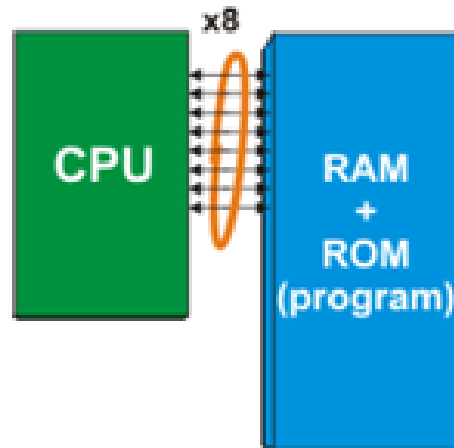


Figura 2.3 – Arquitetura Von- Neumann.
Fonte: Sena (s.d).

Sena (s.d) também descreve que os microcontroladores PIC apresentam uma estrutura de máquina do tipo Harvard que dispõe de dois barramentos de dados distintos. Um é de 8 bits e liga o CPU à RAM (Memória de acesso remoto) e outro tem várias linhas (12, 14 ou 16) e liga o CPU à ROM (Memória somente para leitura). Assim, a CPU pode ler uma instrução, ao mesmo tempo que faz um acesso à memória de dados. Como todos os registros RAM são de 8 bits, todos os dados trocados dentro do microcontrolador estão no mesmo formato. Adicionalmente, durante a escrita do software, apenas 8 bits de dados são considerados. Por outras palavras, só se pode trabalhar com 8 bits de dados, no máximo. Um programa escrito para estes microcontroladores, depois de compilado, vai ser guardado na ROM interna. No entanto, estas posições de memórias não têm apenas 8 bits, mas sim 12,14 ou 16. Os restantes 4, 6 ou 8 bits representam a própria instrução, especificando ao CPU o que fazer com os dados de 8 bits. Esta configuração é ilustrada na figura 2.4:

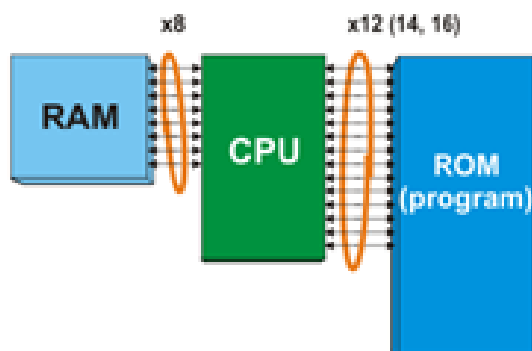


Figura 2.4 – Arquitetura Harvard.
Fonte: Sena (s.d).

As arquiteturas von-Neumann e Harvard também são conhecidas como CISC (*complex instruction set computer* – computador com conjunto de instruções complexas) e RISC (*reduced instruction set computer* – computador com conjunto de instruções reduzido), respectivamente.

Na tabela 2.3, é feita uma comparação mais detalhada sobre a diferença dessas duas arquiteturas.

Tabela 2.3 – Comparação entre as Arquiteturas CISC e RISC.

CARACTERÍSTICAS	CISC	RISC
INSTRUÇÕES POR CICLOS	<i>Instruções complexas executadas em vários ciclos</i>	<i>Instruções simples executadas em um ciclo</i>
ACESSO À MEMÓRIA	<i>Qualquer instrução pode referenciar a memória</i>	<i>Apenas operações LOAD/STORE em memória</i>
PIPELINE	<i>Pouco ou nenhum pipeline</i>	<i>Uso Intenso de pipeline</i>
EXECUÇÃO DAS INSTRUÇÕES	<i>Instruções executadas pelo hardware</i>	<i>Instruções interpretadas pelo próprio programa</i>
FORMATO DAS INSTRUÇÕES	<i>Instruções com formato variável</i>	<i>Instruções com formato fixo</i>
QUANTIDADE DE INSTRUÇÕES E MODOS DE ENDEREÇAMENTO	<i>Várias instruções e modos de endereçamento</i>	<i>Poucas instruções e modos de endereçamento</i>
COMPLEXIDADES DO SISTEMA	<i>Complexidade está no microprograma</i>	<i>Complexidade está no compilador</i>
REGISTRADORES	<i>Conjunto de registradores único</i>	<i>Múltiplos conjuntos de registradores</i>

Fonte: Monteiro (1996).

2.2.2 Microcontroladores PIC

Os microcontroladores PIC são uma família de dispositivos fabricados pela Microchip. Utilizando uma arquitetura RISC, com frequência de clock de até 40Mhz, até 2048k palavras de memória de programa e até 3968 bytes de memória RAM. Além disso podem ser encontrados com periféricos internos, como até quatro temporizadores/contadores, memória EEPROM interna, gerador/comparador/amostrador PWM, conversores A/D de até 12 bits, interface de barramento CAN, I2C, SPI, entre outros. Existem basicamente três famílias de PICS diferenciadas pelo tamanho da palavra da memória de programa: 12, 14 e 16 Bits. Todos estes dispositivos possuem um barramento interno de dados de oito bits (PEREIRA, 2002). Na tabela 2.4 são especificados alguns microcontroladores PIC que possuem 12,14 e 16bits

Tabela 2.4 – Exemplo de microcontroladores com 12,14 e 16 Bits.

12 Bits	14 Bits	16 Bits
12C508	12C671	17C4x
12C509	12C672	17C75x
12CE518	12CE673	17C76x
12C3519	12CE674	18C2xx
16C54	14000	18C4xx
16C55	16C55x	18C858
16C56	16F62x	18F242
16C57	16C7x	18F252
16C58	16F7x	18F258
16C505	16F8x	18F442
16HV540	16F87x	18F458

Fonte: Pereira (2002).

Neste trabalho foi utilizado o microcontrolador PIC18F4550 fabricado pela Microchip. Este modelo possui 40 pinos, dos quais 35 podem ser configurados como I/O, e diversos periféricos, tais como memória EEPROM de 256 bytes, um módulo CCP e ECCP, um módulo SPI e I²C, 13 conversores A/D de 10bits de resolução com tempo de aquisição programável, dois comparadores analógicos, uma comunicação EUSART, um TIMER de 8bits (TIMER2) e três de 16bits (TIMER0, TIMER1 e TIMER3), um módulo de detecção de alta/baixa voltagem (HLVD), além de ter um módulo USB 2.0 capaz de operar no modo *low-speed* (1.5Mbps) ou *full-speed* (12Mbps) (MIYADAIRA, 2009).

2.2.3 Conversor Analógico Digital

Um conversor analógico-digital (A / D) é usado para converter um sinal analógico, como a tensão, para a forma digital, de modo que um microcontrolador pode ler e processar. Alguns microcontroladores têm embutido esses conversores. Um conversor A / D externo também pode ser ligado a qualquer tipo de microcontrolador. Os conversores A / D podem ter diversas resoluções, sendo que os mais comuns são os de 8, 10, 12 e 16 bits, e os níveis de quantização são dados por 2^n , sendo n a resolução do conversor. A maioria dos microcontroladores PIC, com características A/D têm conversores A/D multiplexados que fornecem mais de um canal de entrada analógica, compartilhados com o mesmo conversor. (IBRAHIM, 2008).

Normalmente, utilizam-se sensores especiais para converter a grandeza desejada em um nível de tensão proporcional. A leitura é realizada por meio da conversão analógica em um número binário, proporcional à tensão analógica. Nos conversores A/D, utiliza-se uma tensão de referência (V_{ref}), que serve de fundo de escala para o sinal de saída do conversor, ou seja, as saídas do conversor estarão todas em nível “1” quando a tensão de entrada do conversor for igual ou maior que a tensão V_{ref} . Como a saída do conversor é um número binário proporcional à tensão medida e não o próprio valor binário da tensão, é necessário utilizar técnicas conhecidas como escalonamento, de forma a calcular o valor equivalente à saída do conversor (PEREIRA, 2002).

2.2.4 Portas de entrada e saída (I/O)

I/Os de um microcontrolador são agrupados por PORT's. Este agrupamento se dá geralmente por características elétricas particulares e para facilitar o gerenciamento da máquina. Outra característica importante é que estas portas dizem respeito à sua operação elétrica. Internamente esses pinos são ligados de formas diferentes, principalmente pela sobrecarga de recursos aplicados a cada um deles (SOUZA,2003).

Cada porta de I/O possui dois registradores que controlam suas funções: um registrador PORT e um registrador TRIS. O registrador PORT é utilizado para acesso aos pinos do MCU: uma escrita no registrador PORT escreve nos pinos que estiverem configurados como saída, enquanto uma leitura irá ler o nível lógico presente nos pinos I/O. O registrador TRIS é utilizado para configurar cada pino da respectiva porta como: entrada ou saída. Assim, cada bit do registrador corresponde

a um pino da porta. Se o bit estiver em ‘1’, configura o pino como entrada e se estiver em ‘0’ como saída (PEREIRA, 2002).

2.2.5 Interrupções

Interrupção é um conceito importante em microcontroladores. Ela faz o microcontrolador responder a um evento externo ou interno (por exemplo, um temporizador) muito rapidamente. Quando uma interrupção ocorre, o microcontrolador deixa o fluxo normal do programa em execução e salta para uma parte especial do programa conhecido como rotina do serviço de interrupção (ISR). O código do programa dentro do ISR é executado, e após o retorno do ISR o programa retoma o seu fluxo normal de execução. (IBRAHIM, 2001).

O compilador CCS (utilizado para programação do código no PIC), prevê duas formas básicas para o tratamento de eventos de interrupção: uma modalidade na qual o compilador gera praticamente todo o código necessário ao tratamento de interrupções, cabendo ao programador somente a inclusão de funções de tratamento dos eventos individuais e outra modalidade “manual”, na qual o programador deve incluir todo o código para o tratamento de interrupções.

A diferença entre as duas modalidades reside principalmente na otimização e facilidade de uso: a primeira é de uso mais simples, mas também gera um código muito maior, enquanto a segunda produz um código menor, mas também outorga ao programador a tarefa de realizar os diversos procedimentos necessários ao tratamento de interrupções (SOUZA, 2005).

2.3 Controle de tensão por corte do ângulo de fase

Um TRIAC (Tríodo de corrente alternada) é um componente eletrônico equivalente a dois retificadores controlados de silício ligados em antiparalelo e com o terminal de disparo ligados juntos. Este tipo de ligação resulta em uma chave eletrônica bidirecional que pode conduzir a corrente elétrica nos dois sentidos, obtendo assim um disparo controlado.

O uso do TRIAC é uma alternativa para realizar o controle de fase, onde em um dado semiciclo da rede, o tiristor é acionado em um determinado instante, fazendo com que a carga esteja conectada à rede por um intervalo de tempo menor ou igual a um semiciclo (ALMEIDA, 2003). Por conduzir em ambos os sentidos, o controle de fase pode ser feito tanto no semiciclo positivo quanto no semiciclo negativo, podendo ser usados para o controle de fase de tensão alternada, deixando de

conduzir no instante que a corrente cai a zero. Desta maneira o uso do TRIAC ajusta o nível de tensão aplicada à carga e consequentemente controla a potência fornecida a ela.

2.4 Modelagem de sistemas de primeira ordem

Apresenta-se nessa seção as características de um sistema de primeira ordem, assim como métodos matemáticos para realizar a sua modelagem, como mínimos quadrados e aproximação de padé.

2.4.1 Identificação de um sistema de primeira ordem com atraso

Para aproximar um sistema real por um sistema de primeira ordem, deve-se avaliar a sua resposta em malha aberta e observar se a mesma é compatível com respostas comuns de sistemas de primeira ordem, que tendem a ser parecidas com a figura 2.5.

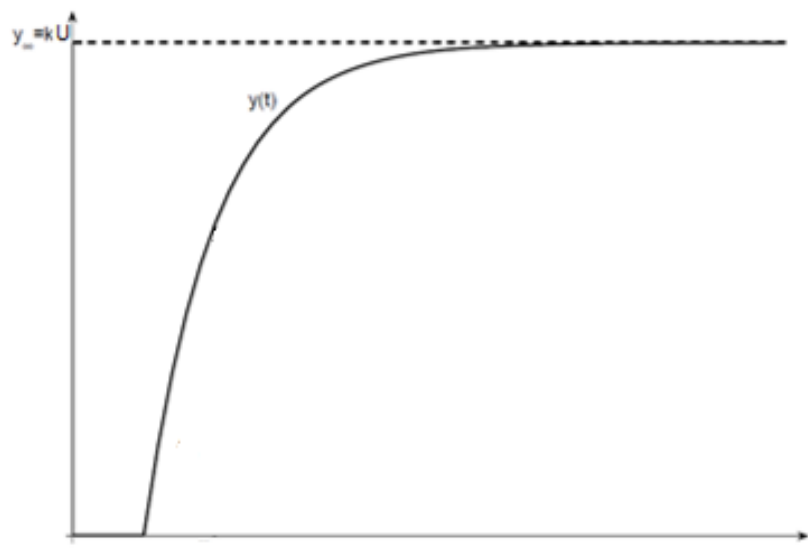


Figura 2.5 – Resposta ao degrau de um sistema de primeira ordem.
Fonte: Guerra (2006) (Adaptado).

Esses sistemas podem ser representados pela equação 2.1:

$$G(s) = \frac{k}{\tau s + 1} \quad (2.1)$$

onde k representa o ganho τ a constante de tempo. Em alguns sistemas ocorre o atraso de transporte que representa o tempo após a ocorrência de uma perturbação na entrada até que seja notada uma

mudança na saída. A resposta (típica) no tempo de sistemas de primeira ordem com atraso de transporte pode ser representada pela equação 2.2 (GUERRA, 2006):

$$G(s) = \frac{k}{\tau s + 1} * e^{-Ts} \quad (1.2)$$

Em que T representa o atraso de transporte. A correspondente resposta de $G(s)$, no domínio do tempo, a um degrau de amplitude U , aplicado em $t = 0$, está representada na figura 2.6, sendo dada pela equação 2.3:

$$y(t) = \begin{cases} 0, & t < T \\ kU \left(1 - e^{-\frac{1}{\tau}(t-T)} \right), & t \geq T \end{cases} \quad (2.2)$$

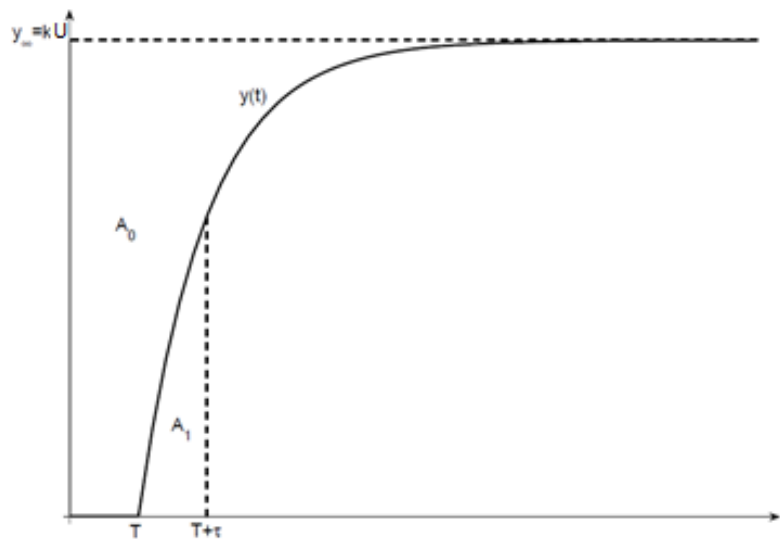


Figura 2.6 – Resposta ao degrau de um sistema de primeira ordem com atraso.
Fonte: Guerra (2006) (Adaptado).

Supondo-se que os parâmetros k , τ e T sejam desconhecidos, pode-se usar o método das áreas (ÅSTRÖM *et al.*, 1995) para descobrir esses valores. Nesse método, a determinação dos parâmetros k , τ e T pode ser feita de acordo com o seguinte algoritmo.

Algoritmo 2.1

- 1 . Aplicar um degrau de amplitude U e obter a resposta $y(t)$.
2. Encontrar a razão $K = y_{\infty}/U$.

3. Calcular numericamente a área A_0 e obter $T + \tau = A_0/kU = A_0/y_\infty$.
4. Calcular numericamente a área $A_1(0 \text{ a } T + \tau)$ e obter $\tau = A_1 e/kU$.
5. Calcular $T = (A_0 - A_1)/kU$.

2.4.2 Aproximação de Padé

Consideremos uma expansão do tipo representado na equação 2.4,

$$f(x) = \sum_{n=0}^N f_n x^n \quad (2.3)$$

Onde N não é necessariamente finito. Segundo Navarro *et al.* (1999) os aproximantes de Padé associados com essa expansão são funções racionais, ou seja, quocientes de dois polinômios, que representam a expansão. Esses aproximantes são caracterizados por dois inteiros positivos L e M, graus do numerador e denominador, respectivamente, da função racional, e são representados pela notação $\left[\frac{L}{M}\right]_{f(x)}$. Frequentemente, em benefício da notação, o índice f(x) é omitido quando o contexto é bem definido, eliminando qualquer possibilidade de confusão. Explicitamente, o aproximante de Padé [L/M] é definido por:

$$\left[\frac{L}{M}\right] = \frac{P_L(x)}{Q_M(x)}, \quad L, M \geq 0 \quad (2.4)$$

com

$$P_L(x) = p_0 + p_1 x + p_2 x^2 + \dots + p_L x^L \quad (2.5)$$

e

$$Q_M(x) = q_0 + q_1 x + q_2 x^2 + \dots + q_M x^M \quad (2.6)$$

Pode-se tomar $q_0 = 1$, de modo que o polinômio $Q_M(x)$ se expresse como

$$Q_M(x) = 1 + q_1 x + q_2 x^2 + \dots + q_M x^M \quad (2.7)$$

Se $f(x)$ e $\frac{P_L(x)}{Q_M(x)}$ forem expandidos em séries de MacLaurin, então os primeiros $L+M+1$ termos da expansão de $R(x)$ devem ser feitos iguais aos primeiros $L+M+1$ termos da expansão de $f(x)$, e então

$$f(x) - \frac{P_L(x)}{Q_M(x)} = c_{L+M+1}x^{L+M+1} + c_{L+M+2}x^{L+M+2} + \dots \quad (2.8)$$

Neste trabalho a expansão de Padé será utilizada para aproximar a função irracional e^{-Ts} por uma função racional $\frac{a(s)}{b(s)}$, onde $L = \text{gr}[a(s)] = 1$ e $M = \text{gr}[b(s)] = 1$, com $\text{gr}[\cdot]$ denotando grau. Para facilitar os cálculos, será considerado $T = 1$. Deste modo, é necessário encontrar $a(s) = a_0s + a_1$ e $b(s) = b_0s + b_1$ de forma que,

$$\epsilon = e^{-s} - \frac{a_0s + a_1}{b_0s + b_1} \quad (2.9)$$

seja mínimo. Portanto deve-se considerar a expansão em série de MacLaurin de e^{-s}

$$e^{-s} = 1 - s + \frac{s^2}{2!} - \frac{s^3}{3!} + \frac{s^4}{4!} - \dots \quad (2.10)$$

e de $\frac{a(s)}{b(s)}$:

$$\frac{a(s)}{b(s)} = \frac{a_0s + a_1}{b_0s + b_1} = a_1 + (a_0 - b_0a_1)s - b_0(a_0 - b_0a_1)s^2 + b_0^2(a_0 - b_0a_1)s^3 + \dots \quad (2.11)$$

Igualando os três primeiros termos das equações 2.11 e 2.12, temos:

$$a_1 = 1 \quad (2.12)$$

$$a_0 - b_0a_1 = -1 \quad (2.13)$$

$$-b_0(a_0 - b_0a_1) = \frac{1}{2} \quad (2.14)$$

Substituindo-se $a_1 = 1$ nas equações 2.14 e 2.15, encontra-se $b_0 = \frac{1}{2}$ e $a_0 = -\frac{1}{2}$. Deste modo:

$$e^{-s} \approx \frac{a_0 s + a_1}{b_0 s + b_1} = \frac{-\frac{1}{2}s + 1}{\frac{1}{2}s + 1} \quad (2.15)$$

Substituindo-se s por Ts , obtém-se a expressão geral para o aproximante de Padé (1,1):

$$e^{-Ts} = \frac{-\frac{Ts}{2} + 1}{\frac{Ts}{2} + 1} \quad (2.16)$$

2.4.3 Método dos mínimos quadrados

Este método é utilizado quando se tem uma distribuição de pontos e se quer ajustar a melhor curva a este conjunto de dados. Inicialmente, será analisado o caso em que a curva de ajuste é uma função linear (AGUIAR *et al.*, s.d):

$$y_i = a + bx_i \quad (2.17)$$

Para que esta seja a reta que melhor se ajusta aos dados, deve-se minimizar a soma das diferenças entre os valores de $f(x)$ tabelados e y_i e os valores da curva de ajuste $a + bx_i$ em cada ponto. Esta diferença pode ser tanto positiva quanto negativa, o que pode ocasionar em uma soma nula das diferenças mesmo com valores muito distantes da reta. Desta maneira, uma forma de evitar o cancelamento é minimizar o quadrado da diferença, o que deixa os cálculos menos complicados do que minimizar o módulo da diferença. Supondo que sejam p pontos tabelados, definimos a função:

$$S(a, b) = \sum_{i=1}^p (y_i - (a + bx_i))^2 \quad (2.18)$$

O problema agora é encontrar valores de a e b que minimizam $S(a, b)$. Usando notação matricial, com os resíduos definidos por:

$$r_i = y_i - (a + bx_i) \quad (2.19)$$

e definindo as matrizes

$$x = \begin{pmatrix} a \\ b \end{pmatrix}, y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix}, R = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_p \end{pmatrix}, A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_p \end{pmatrix} \quad (2.20)$$

Segue que $y_i = a + bx_i$ para todo i variando de 1 até p é o mesmo que $AX = Y$. Assim, como se quer minimizar

$$S(a, b) = \sum_{i=1}^p r_i^2 \quad (2.21)$$

em notação matricial tem-se que

$$\sum_{i=1}^p r_i^2 = R^T R \quad (2.22)$$

onde

$$R = Y - AX \quad (2.23)$$

Denotando $M = S(a, b)$, tem-se

$$M = (Y - AX)^T (Y - AX) = Y^T Y - X^T A^T Y - Y^T A X + X^T A^T A X \quad (2.24)$$

Deseja-se obter os parâmetros a e b ou, em notação matricial, o vetor X de modo a minimizar M . Para isso, o gradiente de M que representa a derivada primeira da função de duas variáveis M deve ser nulo:

$$\nabla M = -A^T Y - Y^T A + 2A^T A X = 0 \rightarrow A^T A X = A^T Y \quad (2.25)$$

Assim, para se encontrar “ a ” e “ b ” que faça com que a soma do quadrado das diferenças entre y_i e $a + bx_i$ seja mínima basta resolver o sistema linear

$$A^T A X = A^T Y \quad (2.26)$$

Como a matriz $A^T A$ é simétrica definida positiva, o sistema linear admite solução única e esta solução será o ponto crítico que será o ponto de mínimo. Efetuando os cálculos de $A^T A X$ e $A^T Y$ tem-se:

$$A^T A = \begin{pmatrix} \sum_{i=1}^p 1 & \sum_{i=1}^p x_i \\ \sum_{i=1}^p x_i & \sum_{i=1}^p x_i^2 \end{pmatrix}, \quad A^T Y = \begin{pmatrix} \sum_{i=1}^p y_i \\ \sum_{i=1}^p x_i y_i \end{pmatrix} \quad (2.27)$$

Este resultado pode ser generalizado para ajustar qualquer polinômio da forma

$$y = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n \quad (2.28)$$

Aos pontos (x_i, y_i) basta fazer:

$$r_i = y_i - (a_0 + a_1 x + \dots + a_n x_i^n) \quad (2.30)$$

$$X = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix}, R = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_p \end{pmatrix}, A = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_p & x_p^2 & \dots & x_p^n \end{pmatrix} \quad (2.291)$$

Assim, para encontrar os pontos a_0, a_1, \dots, a_n deve-se que resolver o mesmo sistema $A^T A X = A^T Y$. Efetuando os cálculos de $A^T A$ e $A^T Y$, tem-se:

$$\begin{pmatrix} \sum_{i=1}^p x_i^0 & \sum_{i=1}^p x_i^1 & \dots & \sum_{i=1}^p x_i^n \\ \sum_{i=1}^p x_i^1 & \sum_{i=1}^p x_i^2 & \dots & \sum_{i=1}^p x_i^{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^p x_i^n & \sum_{i=1}^p x_i^{n+1} & \dots & \sum_{i=1}^p x_i^{2n} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^p x_i^0 y_i \\ \sum_{i=1}^p x_i^1 y_i \\ \vdots \\ \sum_{i=1}^p x_i^n y_i \end{pmatrix} \quad (2.30)$$

2.5 Controladores PID

Esta seção descreve as principais ações de controle, a implementação digital utilizada no trabalho e alguns métodos de sintonia de controladores mais conhecidos na literatura.

2.5.1 Controle de temperatura

Para realizar o controle eficaz da temperatura é necessário desenvolver um controlador automático, o qual compara o valor real da saída do processo com o valor desejado, determina o desvio, e produz um sinal de controle que reduz o desvio a um valor nulo ou muito pequeno. A maneira pela qual o controlador automático produz o sinal de controle é denominada ação de controle (OGATA, 1970).

Esse processo é conhecido como controle de sistemas em malha fechada. A figura 2.7 ilustra o diagrama de blocos e fluxo de informações deste tipo de sistema em que $G_c(s)$ representa a função de transferência do controlador, $G_p(s)$ representa a função de transferência da planta ou sistema a ser controlado, $H(s)$ representa a função de transferência no caminho da realimentação, $U(s)$ a correção no sistema a ser controlado, $E(s)$ o erro do sistema, $C(s)$ o sinal de medição da saída do sistema e $R(s)$ a referência (*set point*).

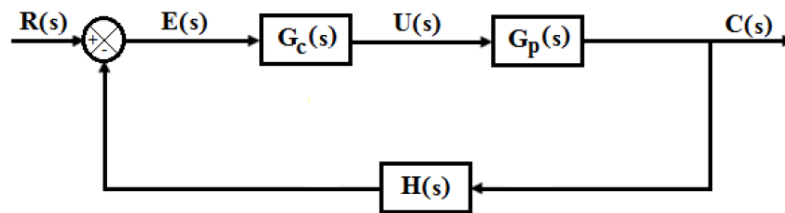


Figura 2.7 – Diagrama de blocos de um sistema de controle em malha fechada.

Fonte: Pesquisa direta (2016).

O tempo necessário para este controle atingir um valor estável varia com o objeto controlado e a tentativa de diminuir o tempo de resposta pode apresentar sobressinal, onde a temperatura ultrapassa a temperatura alvo, ou oscilação, em que a temperatura oscila em torno da temperatura alvo, como mostrado na figura 2.8.

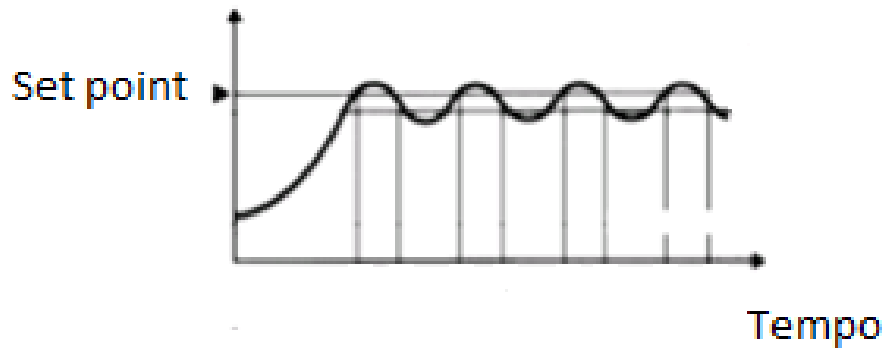


Figura 2.8 – Temperatura oscilando em torno do *set point*.
Fonte: Bezerra *et al.* (2010) (Adaptado).

Tendo em vista várias técnicas de sintonia de controladores existentes, o intuito é encontrar o tipo mais adequado ao sistema, ou seja, que apresentam uma rápida e boa resposta sem gerar sobressinal ou oscilação. As seções a seguir abordam estes tipos de controladores.

2.5.2 Ação de controle liga / desliga (ON/OFF)

Esta ação de controle é relativamente simples e barata, sendo extremamente utilizado em sistemas de controle industriais e domésticos. O elemento atuante possui apenas duas posições fixas que são em muitos casos, ligado e desligado, desta maneira esta ação de controle também é conhecida como controle de duas posições. Considere o sinal de saída do controlador $m(t)$ e o sinal erro atuante $e(t)$. Em um controle de duas posições, o sinal $m(t)$ permanece ou em um valor máximo ou em um valor mínimo, dependendo de o sinal erro atuante ser positivo ou negativo. O sinal de controle $m(t)$ para este tipo de controlador é dado pela equação 2.31 (OGATA, 1970).

$$\begin{aligned} m(t) &= M_1 \text{ para } e(t) > 0 \\ &= M_2 \text{ para } e(t) < 0 \end{aligned} \quad (2.31)$$

Onde M_1 e M_2 são constantes. O valor mínimo M_2 usualmente ou é zero ou $-M_1$.

A figura 2.9 ilustra a curva de resposta do sistema e a ação de controle ON-OFF respectivamente. Na prática existe uma zona diferencial de liga e desliga (histerese), que se deve às características mecânicas do próprio controlador, bem como as necessidades de se evitar manobras frequentes que poderiam desgastar elementos do sistema de controle (KEMPENICH, s.d.).

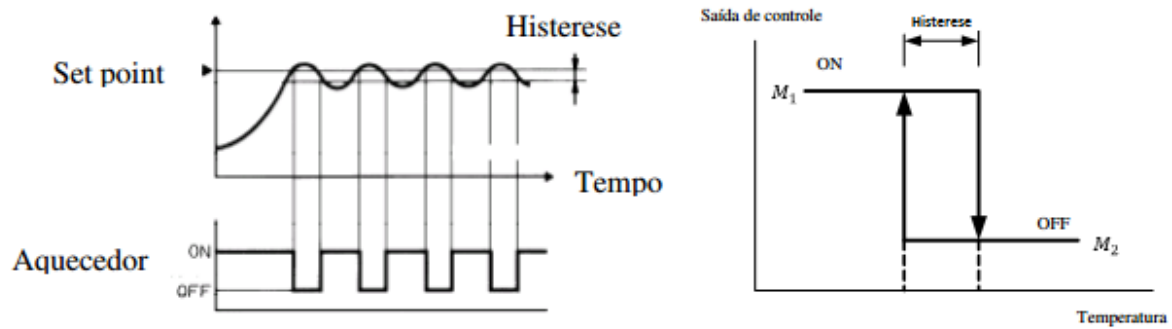


Figura 2.9 – Características da ação de controle ON/OFF.
Fonte: Bezerra *et al.* (2010).

2.5.3 Ação proporcional

Segundo OGATA (1970), a relação entre a saída do controlador $m(t)$ e o sinal do erro atuante $e(t)$ para um controlador com ação proporcional, pode ser definida pela equação 2.32,

$$m(t) = k_p e(t) \quad (2.32)$$

onde k_p é denominado sensibilidade proporcional ou ganho. Independente do mecanismo real, ou da forma da potência de operação, o controlador proporcional é essencialmente um amplificador com um ganho ajustável.

Na figura 2.10 vê-se a resposta de um sistema considerando-se a aplicação de uma ação proporcional. Quanto maior o ganho k_p menor o erro em regime permanente, isto é, melhor a precisão do sistema em malha fechada. Este erro pode se diminuído com o aumento do ganho, entretanto é difícil anular completamente o erro. Por outro lado, quanto maior o ganho, mais oscilatório tende a ficar o comportamento transitório do sistema em malha fechada. Na maioria dos processos físicos, o aumento excessivo do ganho proporcional pode levar o sistema a instabilidade (BAZANELLA *et al.*, 2000).

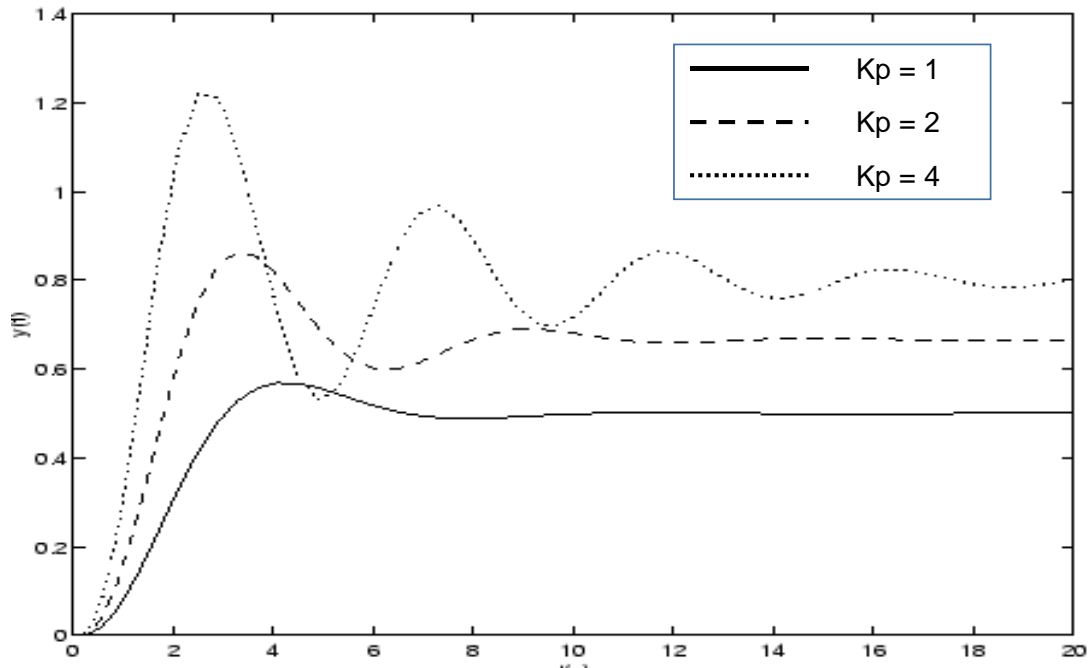


Figura 2.10 – Ação proporcional.
Fonte: Bazanella *et al.*, (2000) (Adaptado).

2.5.4 Ação proporcional-integral

Segundo Ogata (1970) a ação de controle integral varia o valor da saída do controlador $m(t)$ em uma taxa proporcional ao sinal erro atuante $e(t)$, podendo ser definida pela equação 2.34,

$$m(t) = k_i \int_0^t e(t) dt \quad (2.33)$$

onde k_i é uma constante ajustável. A função de transferência do controlador integral é

$$\frac{M(s)}{E(s)} = \frac{k_i}{s} \quad (2.34)$$

A ação integral fará com que o sistema possua mais um pólo na origem da função de transferência do controlador, obtendo erro nulo em regime permanente, desde que o controlador resulte em uma resposta estável. A ação integral está diretamente ligada à melhoria da precisão do sistema. Entretanto, a introdução de um pólo na origem na função de transferência em malha aberta, tende a piorar a estabilidade relativa do sistema em malha fechada ou mesmo torna-lo instável. Por este

motivo, esta ação de controle em geral não é aplicada de maneira isolada (BAZANELLA *et al.*, 2000).

Ainda segundo Bazanella *et al.* (2000) para contrabalançar este fato, a ação integral é em geral utilizada em conjunto com a ação proporcional constituindo-se o controlador PI, cujo sinal de controle é dado por:

$$u(t) = K_p(e(t) + \frac{1}{T_i} \int_0^t e(t)) \quad (2.35)$$

Onde T_i representa o tempo integral ou *reset-time* que corresponde ao tempo em que a parcela relativa a parte proporcional da ação de controle é duplicada. Aplicando-se a transformada de Laplace tem-se a equação 2.37 que representa a função de transferência para o controlador PI. Note que tem-se um zero em $-\frac{1}{T_i}$ que tende a compensar o efeito desestabilizador do polo na origem.

$$G_{pi}(s) = \frac{u(s)}{r(s)} = \frac{K_p(s + \frac{1}{T_i})}{s} \quad (2.36)$$

A ação proporcional deve ser reduzida sempre que esta esteja combinada com a ação integral. O controlador proporcional-integral é utilizado em sistemas com frequentes alterações de carga, sempre que o controlador proporcional, por si só, não seja capaz de reduzir o erro estacionário a um nível aceitável. Contudo o sistema deve ter alterações de carga relativamente lentas, para evitar oscilações induzidas pela ação integral (LOURENÇO, 1997).

2.5.5 Ação proporcional derivativa

A ação derivativa consiste na aplicação de um sinal de controle proporcional a derivada do sinal de erro (BAZANELLA *et al.*, 2000).

$$u(t) = T_d \frac{d e(t)}{dt} \quad (2.37)$$

Sua função de transferência é dada por:

$$G_c(s) = \frac{u(s)}{e(s)} = T_d s \quad (2.38)$$

Este tipo de função implica em um ganho que cresce com o aumento da frequência, desta maneira o sistema fica extremamente sensível a ruídos de alta frequência. Além disso a implementação analógica de um derivador puro é fisicamente impossível. Por estes motivos a implementação da ação derivativa dá-se com a introdução de um pólo em alta frequência que tem justamente a finalidade de limitar o ganho em alta frequência (BAZANELLA *et al.*, 2000).

A ação de controle de um controlador proporcional mais derivativo é definida pela seguinte equação (OGATA, 1970),

$$m(t) = k_p (e(t) + T_d \frac{d e(t)}{dt}) \quad (2.39)$$

em que T_d (tempo derivativo) é expresso em segundos ou minutos, sendo o período de tempo antecipado pela ação derivativa em relação à ação proporcional. O modo derivativo não afeta o erro estacionário, adiciona amortecimento ao sistema e permite o uso de valores de k_p mais elevados, o que implica um menor erro estacionário, porém este modo acentua o ruído em alta frequência presente no sinal do erro (LOURENÇO, 1997).

A ação derivativa quando combinada com a ação proporcional tem justamente a função de antecipar a ação de controle a fim de que o processo seja mais rápido, obtendo respostas transitórias mais rápidas, ou seja, melhora o comportamento dinâmico do sistema em malha fechada (BAZANELLA *et al.*, 2000).

2.5.6 Ação proporcional integral derivativa

A combinação da ação de controle proporcional, ação de controle derivativa e ação de controle integral é denominada ação de controle proporcional mais derivativa mais integral (PID). Esta ação combinada possui as vantagens de cada uma das três ações de controle individuais. A equação de um controlador com esta ação combinada é dada por (OGATA, 1970):

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2.40)$$

Em que $u(t)$ é o sinal de controle; K_p é o ganho proporcional do controlador; $e(t)$ é o sinal de erro do processo; T_i é denominado tempo integral do controlador e T_d é o tempo derivativo.

Um K_p elevado tem o efeito de reduzir o tempo de subida e o erro estacionário (sem nunca o eliminar). O controle integral terá como efeitos, por um lado, eliminar o erro estacionário e por outro piorar a resposta transitória, tornando-a mais oscilatória. Sempre que se utilize controle integral, deve-se testá-lo inicialmente com um K_p reduzido. A utilização do controle derivativo tem como principal consequência uma melhoria da estabilidade do sistema, reduzindo a sobre-elevação e melhorando a resposta transitória (LOURENÇO, 1997).

Os efeitos na resposta, do sistema em malha fechada, ao adicionar os modos proporcional, integral e derivativo são listados na próxima tabela 2.5.

Tabela 2.5 – Efeitos no sistema das ações proporcional, integral e derivativo.

Resposta CF	Tempo de Subida	Sobreelevação	Tempo de Estabelecimento	Erro Estacionário
Proporcional	Diminuição	Aumento	Sem alteração	Diminuição
Integral	Diminuição	Aumento	Aumento	Elimina
Derivativo	Sem alteração	Diminuição	Diminuição	Sem alteração

Fonte: Lourenço (1997).

A seleção do controlador deve depender das condições operativas do sistema e de especificações de performance tais como, o erro estacionário máximo, a sobre-elevação máxima e o tempo de estabelecimento permitido. Se o erro estacionário não é tolerado, então o modo integral deve ser incluído no controlador, uma vez que esta é a única ação que o permite eliminar ou reduzir. A necessidade da ação derivativa pode ser ditada por uma sobre-elevação máxima e/ou tempo de estabelecimento. Se um reduzido erro estacionário não é crítico para as condições operativas do sistema, então é possível omitir o modo integral, e o uso do modo derivativo depende entre outros fatores da necessidade ou não de adicionar ganho suplementar ao modo proporcional (LOURENÇO, 1997).

2.5.7 Controle Digital

A implementação de uma lei de controle em tempo contínuo em um microcontrolador ou em um computador digital, pode ser feita por meio de aproximações numéricas da derivada e da integral, no caso de um controle PID por exemplo.

Muitos controladores industriais têm somente a ação PI e em outros a ação derivativa pode ser desligada. Segundo Åström *et al.* (1995), isto ocorre porque o controle PI é adequado para todos os processos onde as dinâmicas são essencialmente de primeira ordem. Para descobrir se este é o caso, basta observar a resposta ao degrau do processo que deve assemelhar-se a um sistema de primeira ordem. Outra razão é quando o processo é projetado para que sua operação não exija controle fino, mesmo se o processo tem dinâmica de alta ordem, necessita apenas de uma ação integral para eliminar o erro de regime, e de uma resposta transiente adequada, proporcionada pela ação proporcional.

Similarmente, o controle PID é suficiente para processos onde a dinâmica dominante é de segunda ordem. Para tais processos, não há um ganho de benefício ao se usar um controlador mais complexo. Um caso típico de melhoramento pela ação derivativa é quando as dinâmicas são caracterizadas por constantes de tempo que diferem em magnitude. A ação derivativa pode ser usada para acelerar a resposta. Controle de temperatura é um caso típico. O controle derivativo é também benéfico quando se exige controle fino de um sistema de ordem superior. As dinâmicas de ordem superior limitariam o resultado do ganho proporcional para o bom controle. Com uma ação derivativa, a evolução da oscilação amortecida é evitada, daí um ganho proporcional alto pode ser usado para aumentar a resposta transiente (ÅSTRÖM *et al.*, 1995).

2.5.8 Implementação Digital do Controlador PID

A implementação do controlador PID pode ser feita com aproximações numéricas da integral que aparecem na lei de controle (ÅSTRÖM *et al.*, 1995). Desta forma, é possível descrever cada uma das ações por uma equação de diferenças. A equação de diferenças descreve a operação matemática a ser programada no microcontrolador onde será implementado o PID digital. A figura 2.11 ilustra um sistema de controle onde um processador (ou microcontrolador) toma o bloco de controle responsável por realizar a compensação e atuar sobre a planta a ser controlada.

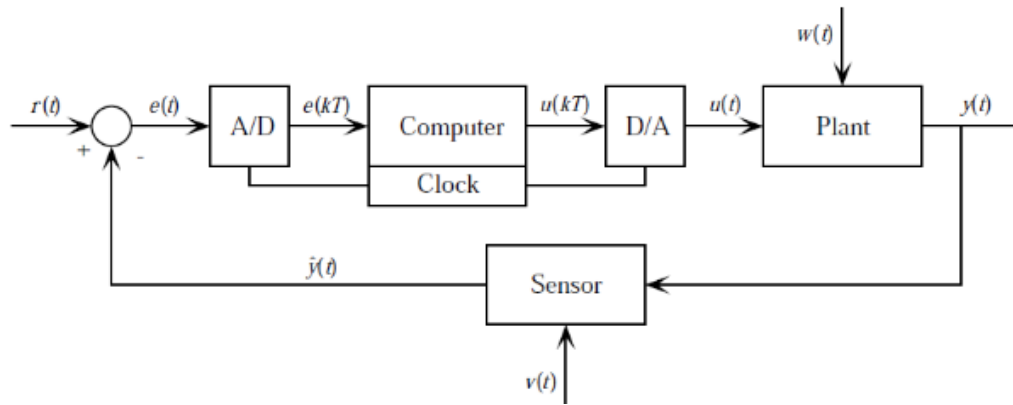


Figura 2.11 – Sistema de controle digital.
Fonte: Neves (2014).

O sinal de erro chega ao controlador por meio de uma conversão A/D. O valor numérico é então processado gerando outro sinal, que é passado para um conversor Digital/Analógico (D/A), onde retorna sua forma analógica e controla a planta.

A função do controlador PID no domínio do tempo pode ser dada pela equação 2.43.

$$u(t) = K_p \times e(t) + K_i \times \int_0^t e(t) d\tau + K_d \times \frac{de(t)}{dt} \quad (2.42)$$

Onde:

$u(t)$: é o sinal de saída do controlador no instante t ;

$e(t)$: é o sinal de erro na entrada do controlador no instante t ;

K_p : é o coeficiente da ação proporcional;

K_i : é o coeficiente da ação integral;

K_d : é o coeficiente da ação derivativa;

t : é o instante do estado a ser processado.

Para implementar esta função no microcontrolador foi feito uma análise da relação de cada coeficiente (K_p , K_i e K_d) com a entrada A/D do PIC. Para esta análise, foi considerado um tempo de amostragem de 1 segundo.

O termo proporcional pode ser computado tomando a multiplicação da constante proporcional pela variável de erro que chega a um instante t :

$$K_p \times e(t) \Leftrightarrow K_p \times e[n] \quad (2.43)$$

Onde n denota o instante da amostragem da variável de erro recebida pelo conversor A/D. Como as amostras chegam igualmente espaçadas por uma diferença de 1 novo valor, elas estão em um domínio do tempo discreto.

O conceito de derivada pode ser definido como:

$$\frac{d(t)}{d_t} = \lim_{\Delta t \rightarrow 0} \frac{f(t) - f(t - \Delta t)}{\Delta t} \quad (2.44)$$

Se Δt for suficientemente pequeno, pode-se obter uma aproximação muito similar à derivada pela definição. Colocando este valor como período de amostragem do conversor A/D, tem-se:

$$\Delta t = t_{amostragem\ A/D} = \frac{1}{f_{amostragem\ A/D}} \quad (2.45)$$

Substituindo a equação 2.45 em 2.44, tem-se:

$$\begin{aligned} \frac{d(t)}{d_t} &= \lim_{\Delta t \rightarrow 0} \frac{f(t) - f(t - \Delta t)}{\Delta t} \rightarrow \\ \Delta t = t_{amostragem\ A/D} &= \frac{f(t) - f(t - \Delta t)}{\Delta t} \end{aligned} \quad (2.46)$$

Passando para o domínio de tempo discreto, tem-se:

$$\frac{d[n]}{d_t} \approx \frac{f[n_t] - f[n_t - 1]}{t} \quad (2.47)$$

Adicionando K_d à equação e considerando $t=1s$, tem-se:

$$K_d \times \frac{d(t)}{d_t} = K_d \times (e[n] - e[n - 1]) \quad (2.48)$$

O termo integral segue o mesmo princípio numérico, porém de forma inversa, ao invés de uma subtração, o termo integrador irá realizar uma somatória. A definição da integral no tempo contínuo pode ser dada pela equação 2.49:

$$\int_t^{t+\Delta t} f(t)dt = \lim_{\Delta t \rightarrow 0} \sum_{n=t}^{\Delta t+t} f(n) \times \Delta t \quad (2.49)$$

A equação 2.49 pode ser desmembrada em duas amostras, visto que para um sistema de controle interessam os estados atuais e o de uma amostra anterior. Desta maneira tem-se:

$$\int_t^{t+\Delta t} f(t)dt = \lim_{\Delta t \rightarrow 0} \sum_{n=t}^{\Delta t+t} f(n) \times \Delta t = \Delta t \times \{f(t) + f(t + \Delta t)\} \quad (2.50)$$

Da mesma forma como foi feito para o termo derivativo, o tempo de amostragem do conversor A/D será considerado suficientemente pequeno para Δt , resultando na equação 2.51:

$$\int_n^{n+\Delta n} f[n]dn = f[n] + f[n - 1] \quad (2.51)$$

Adicionando K_i a equação, tem-se:

$$K_i \times \int_{t=0}^t e(t)dt = K_i \times f([n] + f[n - 1]) \quad (2.52)$$

Juntando as equações 2.43, 2.48 e 2.52 obtém-se o valor do estado que será enviado ao conversor D/A em função da entra $e(t)$, sendo representado pela equação 2.53.

$$u[n] = K_p \times e[n] + K_d \times (e[n] - e[n - 1]) + K_i \times f([n] + f[n - 1]) \quad (2.53)$$

2.6 Métodos práticos de sintonia de controle PID

Existem vários métodos para ajuste de controladores PID, sendo que estes requerem algum tipo de informação sobre a dinâmica do processo a ser controlado. Para um método prático de ajuste, deve

ser possível obter informações a partir de ensaios simples sobre o processo, ao mesmo tempo em que estas informações devem ser suficientes para possibilitar um ajuste adequado do controlador.

Em 1942, J.G. Ziegler e N. B. Nichols apresentaram um conjunto de regras para sintonia de controladores PID para determinação dos valores dos parâmetros K_p , T_i e T_d , baseados na característica da resposta temporal de uma determinada planta mediante a um sinal degrau de entrada (OGATA, 1970).

2.6.1 Método da resposta ao degrau

O procedimento normal no ajustamento dos parâmetros por este método consiste na abertura da malha para que não haja realimentação e na obtenção da sua resposta a um degrau de amplitude M na entrada da referência. A resposta deverá ter uma forma de S , em situação contrária o método não é aplicável (LOURENÇO, 1997).

A curva com o formato em S pode ser caracterizada por duas constantes, o atraso L e a constante de tempo T , sendo determinadas ao passar uma tangente pelo ponto de inflexão da curva, como mostra a Figura 2.12. A partir dos pontos onde a tangente intercepta o eixo das abscissas e a linha horizontal com valor K , que é o valor que o sistema possui de saída quando se estabiliza, obtêm-se T e L , respectivamente.

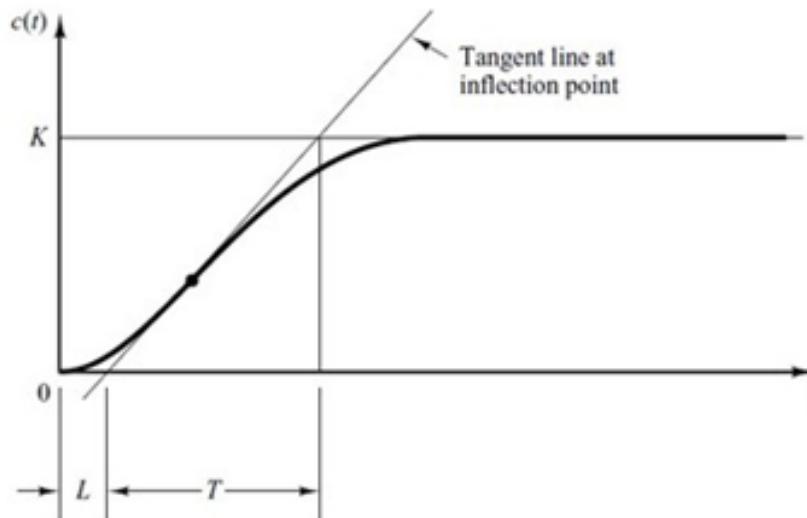


Figura 2.12 – Curva de resposta a um degrau.
Fonte: Ogata (1970).

Uma vez obtidos os parâmetros L , T e K , pode-se recorrer à tabela 2.6 para determinar os valores dos parâmetros dos controladores.

Tabela 2.6 – Cálculo dos ganhos PID (primeiro método).

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{KL}$	∞	0
PI	$0,9 \frac{T}{KL}$	$3,3L$	0
PID	$1,2 \frac{T}{KL}$	$2L$	$0,5L$

Fonte: Ibrahim (2006).

2.6.2 Método do ganho crítico

Este método baseia-se no ajuste de uma malha fechada até se obter oscilações com amplitude constante. Utiliza-se um conjunto de fórmulas para determinar os parâmetros do controlador, as quais requerem duas medidas do sistema: o Ganho crítico (G_u , ou seja, o ganho mínimo que torna o processo criticamente estável) e o período de oscilação correspondente, P_u (LOURENÇO, 1997), conforme mostra a figura 2.13.

Primeiramente é definido $T_i = \infty$ e $T_d = 0$. Utiliza-se somente a ação de controle proporcional, K_p que é aumentada de 0 ao valor crítico K_{cr} , onde a saída apresenta uma oscilação sustentada pela primeira vez. Então, são obtidos experimentalmente o ganho crítico K_{cr} e o período P_{cr} , de acordo com a Figura 2.13 (OGATA, 1970).

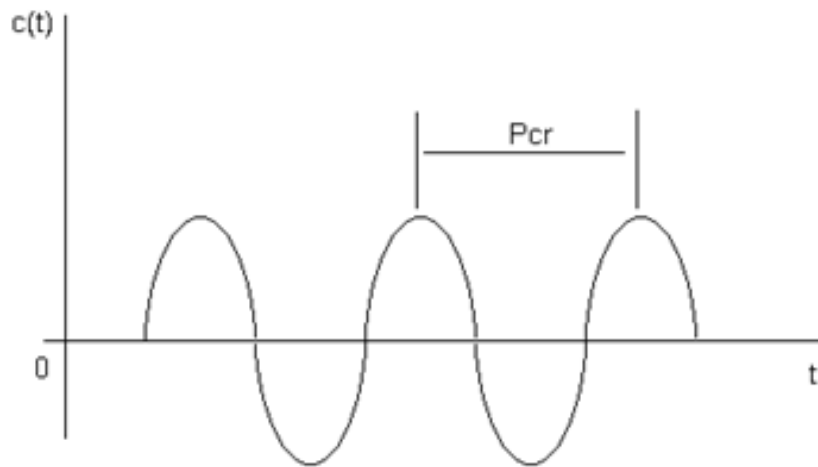


Figura 2.13 – Resposta do sistema ganho crítico.
Fonte: Ogata (1970).

Uma vez obtidos estes parâmetros, recorre-se à **Erro! Fonte de referência não encontrada.**7 para obter os parâmetros dos controladores.

Tabela 2.7 – Tabela para cálculos dos ganhos PID (segundo método).

<i>Tipo de Controlador</i>	K_p	T_i	T_d
P	$0,5 K_{cr}$	∞	0
PI	$0,4 K_{cr}$	$0,8 P_{cr}$	0
PID	$0,6 K_{cr}$	$0,5 P_{cr}$	$0,125 P_{cr}$

Fonte: Ogata (1970).

2.7 Medição de temperaturas

Esta seção tem como objetivo descrever o sensor de temperatura utilizado e suas características.

2.7.1 Termopares

Os termopares são sensores de temperatura bastante utilizados devido ao seu baixo custo e larga faixa de operação. O modelo matemático do termopar pode ser descrito pela seguinte equação (GUERRA, 2006):

$$v(t) = K_t \theta(t) + b, \quad (2.54)$$

Onde $v(t)$ denota a tensão fornecida pelo termopar, K_t o ganho, $\theta(t)$ representa a temperatura no interior do forno e b é uma constante.

Os termopares são constituídos de dois metais distintos que unidos por suas extremidades formam um circuito fechado e geram uma tensão elétrica de acordo com o diferencial de temperatura entre as duas juntas. Esta tensão também é conhecida como tensão de *Seebeck*.

No trabalho de Lourenço (2008), tem-se que a relação entre a temperatura na junção e a tensão gerada é dada por:

$$\Delta V_c = S(T_{tc})T_{tc} \quad (2.55)$$

Em que $S(T_{tc})$ é o coeficiente de Seebeck que depende do tipo de termopar e é função da temperatura T_{tc} . A dependência de S com a temperatura provoca uma certa não linearidade na característica tensão *versus* temperatura, porém, para uma pequena faixa de temperatura, a equação 2.55 pode ser simplificada por:

$$\Delta V_c = S * T_{tc} \quad (2.56)$$

No qual S é uma constante que depende do tipo de termopar. A tabela 2.8 apresenta exemplos de termopares com os materiais que constituem cada tipo, a faixa de operação de cada um e exemplo de utilização.

Tabela 2.8 – Termopares

Tipo	Composição	Range	Características
T	Cobre/Cobre - Níquel (+) / (-)	200 a 370°C	Podem ser usados em atmosferas oxidantes, redutoras, inertes e no vácuo. Adequados para Mediações abaixo de zero grau. Apresenta boa precisão na sua faixa de utilização.
J	Ferro/Cobre - Níquel (+) / (-)	-40 a 760°C	Utilizados em atmosferas oxidantes, redutoras, inertes e no vácuo. Não deve ser usados em atmosfera sulfurosas e não se recomenda o uso em temperatura abaixo de zero grau. Apresenta baixo custo.
E	Níquel-Cromo/Cromo - Níquel (+) / (-)	-200 a 870°C	Próprio para atmosfera oxidante e inertes. Em ambientes redutores ou vácuo perde suas características termoeletricas. Adequado para uso em temperatura abaixo de zero.
K	Níquel-cromo/Níquel - alumínio (+) / (-)	-200 a 1260°C	Recomendáveis em atmosfera oxidante ou inertes. Ocasionalmente pode ser usado abaixo de zero grau. Não deve ser utilizado em atmosfera redutoras e sulfurosas. Seu uso no vácuo é por curto período de tempo
S R	Platina-10% Ródio/Platina (+) / (-)	0 a 1600°C	Recomendáveis em atmosfera oxidante ou inertes. Não deve ser usado abaixo de zero grau, no vácuo em atmosfera redutoras ou com vapores metálicos. Apresenta boa precisão em temperaturas elevadas
	Platina-13% Ródio/Platina	0 a 1600°C	
B	Platina-30% Ródio/ Platina-6% Ródio (+) / (-)	600 a 1700°C	Recomendáveis em atmosfera oxidante ou inertes. Não deve ser usado abaixo de zero grau, no vácuo em atmosfera redutoras ou com vapores metálicos. É mais adequado a temperaturas mais elevadas que os tipos S/R
N	Níquel-Cromo-Silício (+) Níquel-Silício (-)	-200 a 1260°C	Excelente resistência a oxidação até 1200°C, curva FEMxTemp, similar ao tipo K, porém possui menor potência termoeletrica, apresenta maior estabilidade e menor drift tempo

Fonte: CR resistências

2.7.2 Compensação da junta de referência

A junta de referência, ou junção fria, é a que conecta o termopar ao instrumento de aquisição de sinais. Todas as tabelas normalizadas dão os valores da tensão de saída do termopar considerando que a segunda junção do termopar (junta de referência) é mantida exatamente zero graus Celsius. Antigamente isto conseguia-se conservando a junção em gelo fundente. Contudo a manutenção do gelo nas condições necessárias não era fácil, logo optou-se por medir a temperatura da junção fria e compensar a diferença para o zero graus Celsius (PROTOLAB, s.d).

A leitura desta segunda temperatura é utilizada para o cálculo da temperatura verificada na extremidade do termopar, podendo ser representada pela equação 2.57. A figura 2.14 ilustra a leitura de um termopar tipo J (LOURENÇO, 2008):

$$V = V_1 + V_3 - V_2 = V_I(\theta_1) + V_1(\theta_3) - v_2(\theta_2) \quad (2.57)$$

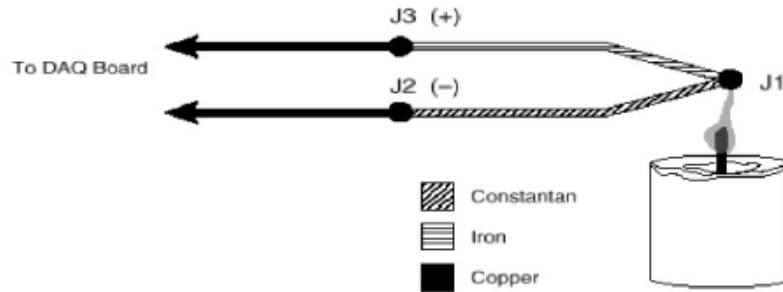


Figura 2.14 – Leitura de um termopar tipo j.

Fonte: Lourenço (2008).

Esta análise pode ser simplificada para a figura 2.15.

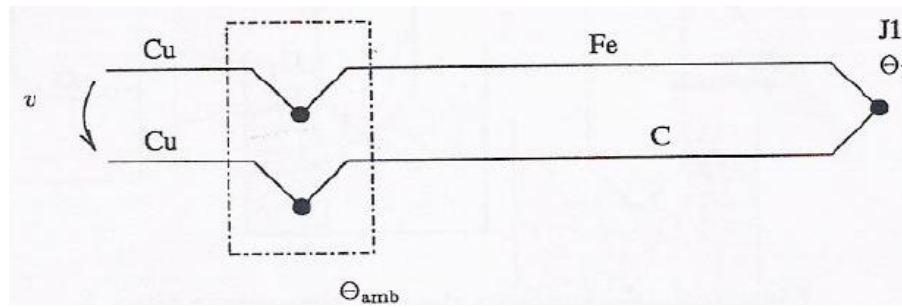


Figura 2.15 – Modelo simplificado de leitura de termopar.

Fonte: Lourenço (2008).

Desta maneira, a temperatura (θ_1) pode ser facilmente calculada pela equação 2.58, conhecendo $V(\text{lido})$ e $V_{tp}(\theta_{amb})$ (por meio de θ_{amb}).

$$V = V_{tp}(\theta_1) - V_{tp}(\theta_{amb}) \quad (418)$$

2.7.3 Linearização de dados

A tensão de saída dos termopares é altamente não linear. Desta maneira, é necessário aproximar a curva tensão vs temperatura dos termopares por polinômios ou usando-se uma tabela. A aproximação por polinômios é dada pela seguinte equação (LOURENÇO, 2008):

$$T = a_0 + a_1 * v + a_2 * v^2 + \dots + a_n * v^n \quad (2.59)$$

Onde, v é a tensão do termopar em volts, T é a temperatura em graus Celsius e a_0 até a_n são os coeficientes que são específicos de cada tipo de termopar. Pode-se obter a tensão equivalente por um polinômio similar a equação 2.60.

$$V = c_0 + c_1 * T + c_2 * T^2 + \dots + c_n * T^n \quad (2.60)$$

Onde, v é a tensão em volts, T a temperatura em graus Celsius e c_0 até c_n são os coeficientes que são específicos de cada tipo de termopar.

3 METODOLOGIA

Este capítulo tem a finalidade de apresentar as etapas realizadas até chegar no projeto do controlador, que tem como objetivo seguir um perfil específico de temperatura.

Para modelagem do forno, foi utilizado um Variac o qual fornecia diferentes tensões de alimentação. As respectivas curvas de temperatura foram obtidas com um termopar tipo K e amplificadas por uma placa amplificadora de tensão. O sinal foi enviado para o módulo USB-6009 e tratadas no software Labview.

O controlador foi implementado digitalmente por meio de programação no microcontrolador da família PIC18F4550. O circuito de controle foi responsável pela geração de pulsos que serviu para acionar um TRIAC o qual controlou a tensão enviada ao forno. A figura 3.1 ilustra a bancada de trabalho.

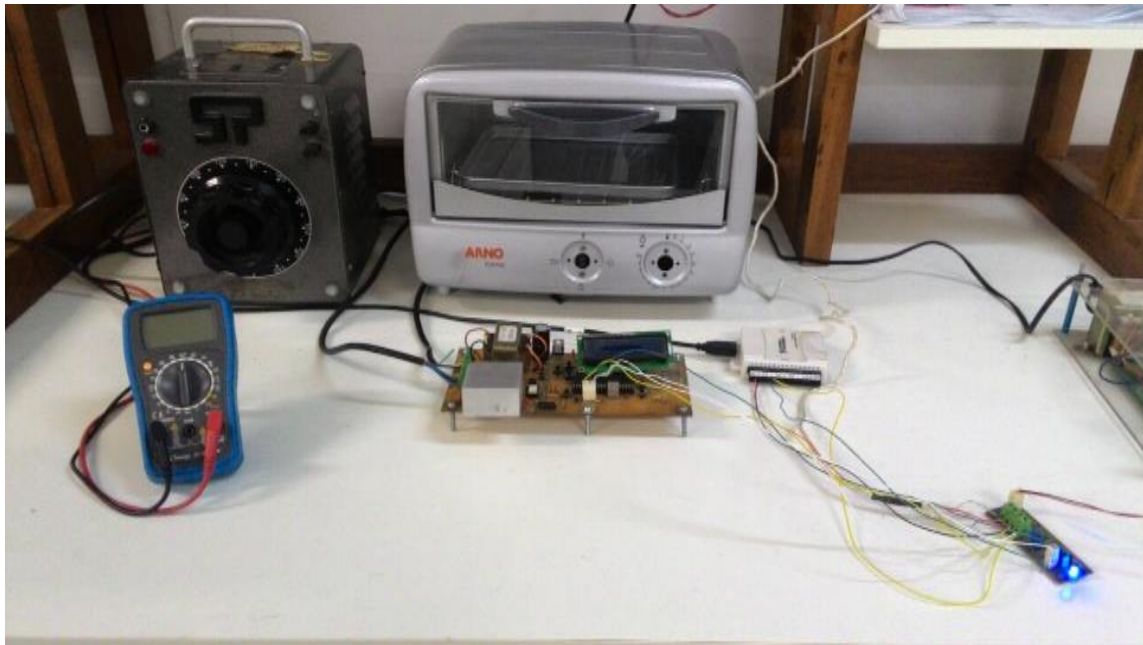


Figura 3.1 – Bancada de trabalho.
Fonte: Pesquisa direta (2016).

Este trabalho foi desenvolvido no Laboratório de Protótipos e Desenvolvimento de Novas Tecnologias, pertencente ao Departamento de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto, em Ouro Preto, Minas Gerais.

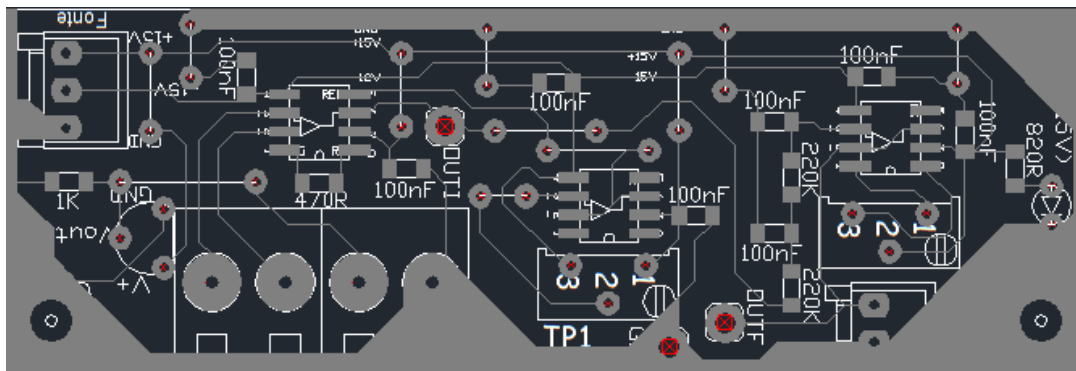
3.1 Projeto do sistema de aquisição de temperatura

De acordo com o trabalho de Basílio *et al.* (2002), o termopar tipo K pode ter como parâmetros $K_t = 41.2\mu V/^{\circ}C$ e $b = -0.985mV$. Sabendo que a temperatura máxima do termopar utilizado é $1.260^{\circ}C$, e utilizando a equação 2.50, a tensão máxima fornecida será de $0,042V$. A entrada para a placa de aquisição de dados USB-6009 registra valores entre $-0,3V$ e $0,8V$ (menor escala), tornando-se necessário utilizar um circuito amplificador para multiplicar a tensão fornecida pelo termopar.

A figura 3.2 ilustra placa amplificadora de tensão. Esta placa foi desenvolvida no ambiente AutoCAD.



(a)



(b)

Figura 3.2 – (a) Placa amplificador de tensão; (b) Projeto no AutoCAD.
Fonte: Pesquisa direta (2016).

Os componentes da placa amplificadora de tensão estão listados na tabela 3.1:

Tabela 3.1 – Componentes placa amplificadora de tensão.

Componente	Valor	Quantidade	Encapsulamento
INA 128	-	1	SOIC-8
OP07	-	2	SOIC-8
Resistor	470	1	0805
Resistor	820	1	0805
Resistor	1k	1	0805
Resistor	15k	1	0805
Resistor	47k	1	0805
Capacitor	100nF	7	0805
Capacitor	47nF	1	0805
Conector 2 pinos	-	1	Molex
Conector 3 pinos	-	1	Molex
Conector KRE	-	1	-
Pino torneado	-	3	2.54mm
LM35	-	1	TO-92
Trimpot	20k	1	3296Y
Trimpot	100k	1	3296Y
LED azul	-	1	2mm

Fonte: Pesquisa direta (2016).

O componente INA128 filtra e realiza a amplificação do sinal obtido pela ponta do termopar. Foi utilizado dois amplificadores operacionais (OP07), onde um realiza o ajuste de zero (offset), enquanto o outro é utilizado para fazer o Buffer de tensão.

O sensor de temperatura LM35 é utilizado para fazer o papel da junta de referência. Ao ser alimentado com uma tensão de 5 volts no terminal positivo e zero volts no terminal negativo, produz na saída um nível de tensão que varia entre zero e 1,5 volt para temperaturas entre zero e 150°C, ou seja, cada 10 milivolts na saída do sensor corresponde a 1°C (TEXAS INSTRUMENTS, 2013). Assim, com as temperaturas registradas pela ponta do termopar e pelo LM35, que representa a temperatura ambiente, é possível obter a real temperatura dentro do forno de acordo com a equação 2.54.

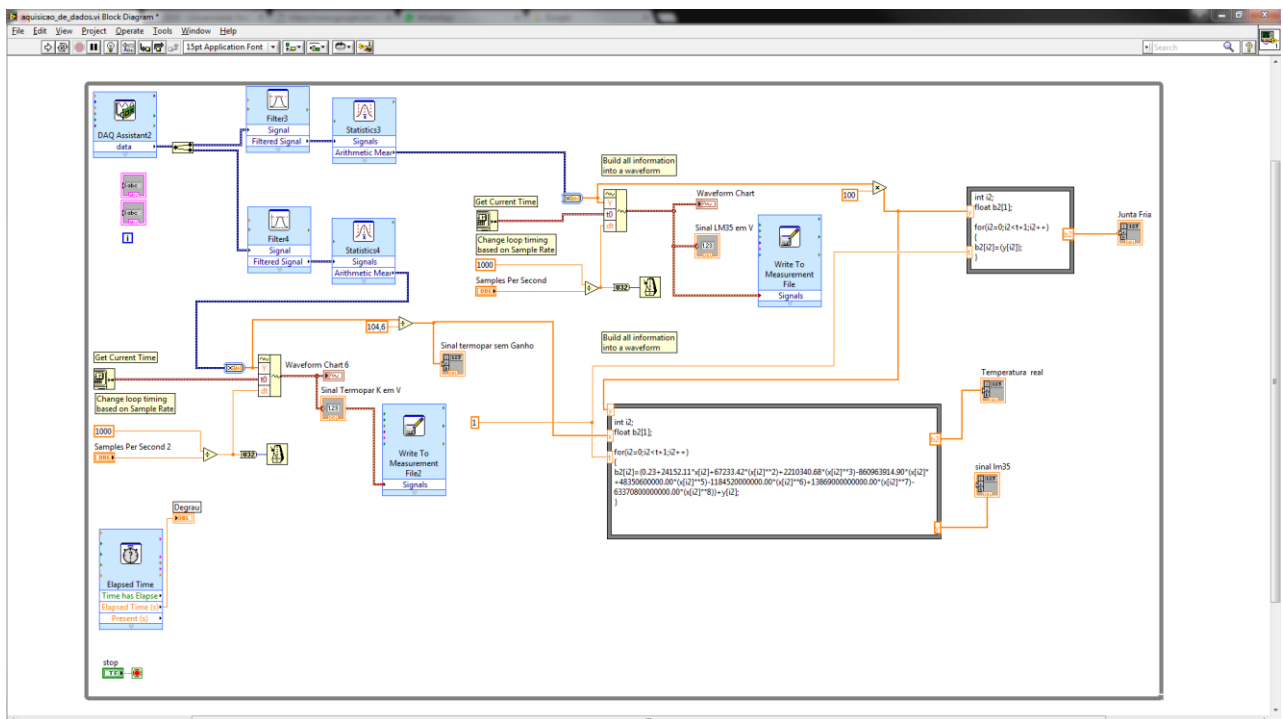
Os sinais referentes a ponta do termopar e do LM35 são registrados pela placa de aquisição USB-6009 da National Instruments. A escolha deste equipamento foi devido a facilidade de integração

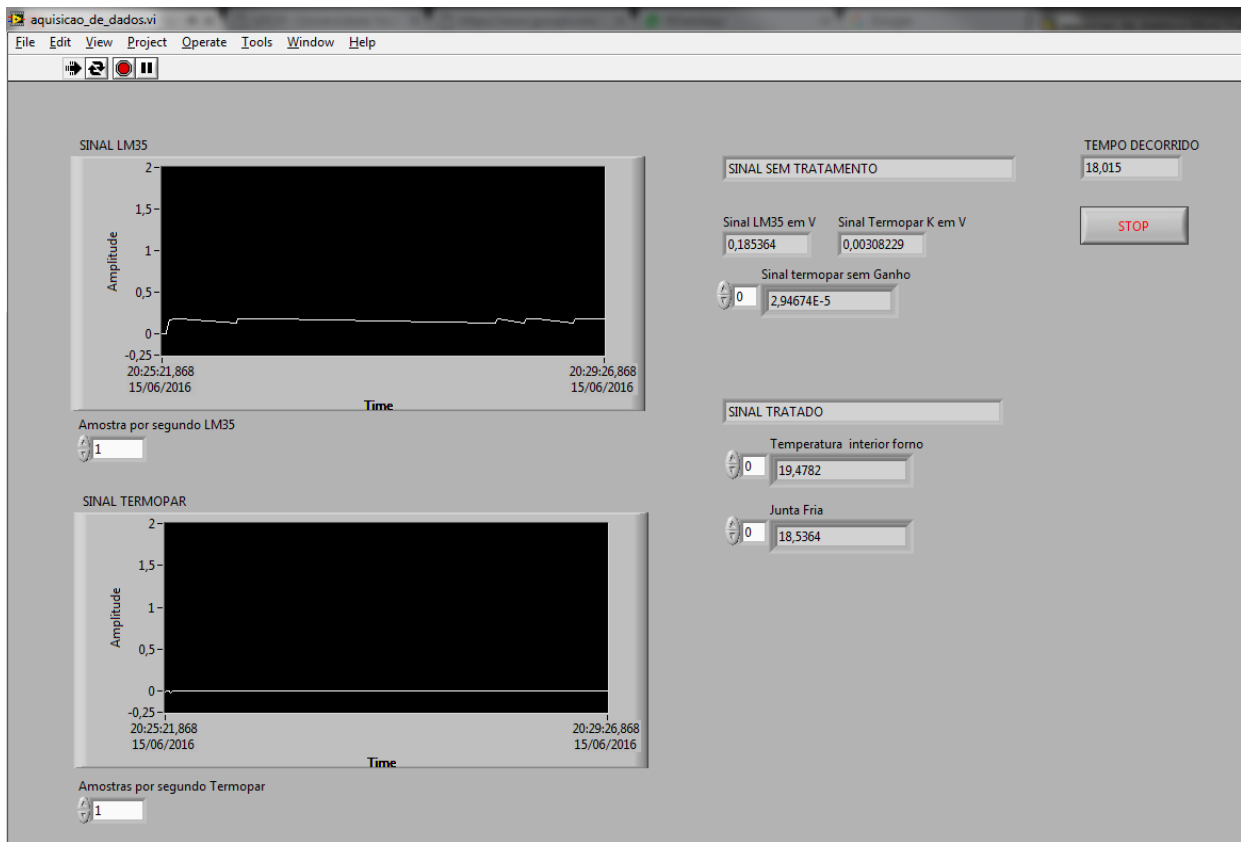
com software LabVIEW, também da National Instruments, assim como por apresentar portas analógicas, taxa de amostragem (48 kS/s) e rapidez na aquisição de dados satisfatória. A figura 3.3 ilustra o módulo USB-6009 da National Instruments.



Figura 3.3 – USB-6009.
Fonte: National Instruments (2014).

A programação no *software* LabVIEW pode ser visualizada na figura 3.4:





(b)

Figura 3.4 – (a) Interface de programação; (b) Interface Visual.
Fonte: Pesquisa direta (2016).

Toda a lógica da programação é colocada dentro de um *loop while* para que o programa fique operando o tempo todo. O bloco *DAQ assistance* registra dois sinais de tensão diferentes, um referente a ponta do termopar amplificado e outro referente ao LM35, com uma taxa de amostragem de 1000 amostras por segundo. Cada um desses sinais será tratado de forma semelhante. Primeiramente é aplicado um filtro para reduzir os ruídos e depois é realizado uma média aritmética dos sinais registrados. Esses dados foram enviados para um arquivo do Microsoft Office Excel a cada 1s.

3.2 Aquisição das temperaturas

Para identificar os parâmetros do modelo do forno, foi fornecido ao sistema diferentes valores de degrau de entrada com o Variac. É necessário calcular os diferentes ganhos DC, tempos de atrasos e constantes de tempo para que se identifique possíveis não linearidades no sistema.

Pelo fato do forno ser puramente resistivo, uma excitação com uma tensão senoidal $v(t) = V_m \sin(\omega t)$, corresponde a uma excitação por um degrau de amplitude $\frac{V_m}{\sqrt{2}}$ (valor eficaz de $v(t)$) (BASILIO *et al.*, 2002). Desta maneira, o comportamento do forno a resistência para diferentes degraus de diferentes amplitudes pode ser obtido experimentalmente da seguinte forma.

Procedimento 3.1

- 1 . Aplicar ao forno tensões senoidais de diferentes amplitudes utilizando um variac (gerador de tensão variável alternada).
2. Para cada tensão aplicada, registrar as curvas de resposta de temperatura.
3. Após atingir o valor de temperatura de regime permanente, desligar o variac e registrar novamente a temperatura.

3.3 Identificação dos parâmetros do forno

Pode-se notar que, durante o período de aquecimento, o forno possui uma resposta livre típica de um sistema de segunda ordem superamortecido. É comum modelar um sistema de segunda ordem superamortecido como um sistema de primeira ordem com atraso (ÅSTRÖM *et al.*, 1995).

Os parâmetros necessários para modelagem do forno como constante de tempo e atraso podem ser estimados traçando uma reta tangente ao ponto de inflexão da curva obtida no procedimento 3.1, como ilustrado na figura 2.12 (seção 2.6.1) referente ao primeiro método de Ziegler-Nichols. O ponto de inflexão pode ser encontrado calculando a derivada de todos os valores da curva. O valor com maior derivada, representa este ponto. O termopar apresenta uma resposta de saída com muita oscilação, assim, encontrar o correto ponto de inflexão ficou muito complicado como ilustra a figura 3.5. Desta maneira, foi necessário recorrer a outros métodos para encontrar os valores corretos dos parâmetros.

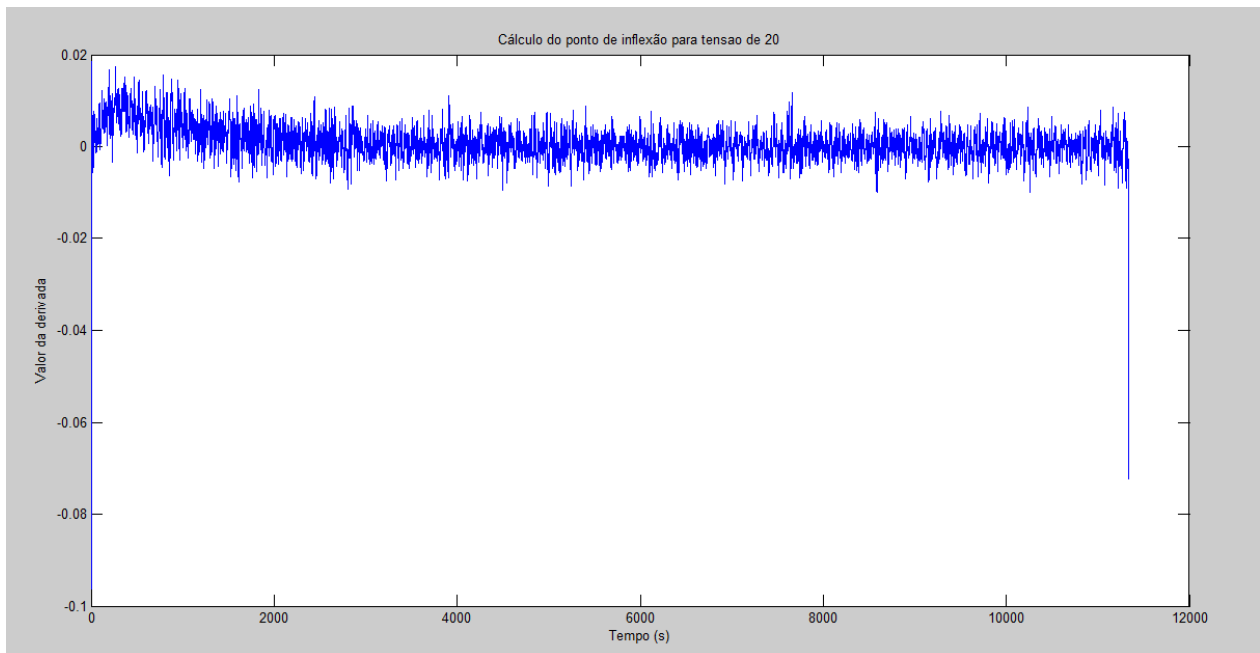


Figura 3.5 – Cálculo do ponto de inflexão para tensão de 20.
Fonte: Pesquisa direta (2016).

Conforme mostrado no capítulo 2, seção 2.3.1 (algoritmo 2.1), para encontrar os parâmetros do modelo adotado, pode-se utilizar o método das áreas. Deve-se observar, porém que o método das áreas supõe que a resposta inicial do sistema é nula ($y_{\infty} = 0$). Desta maneira, para que se possa calcular o valor do ganho, deve-se utilizar a variação de temperatura e não o valor do regime permanente. Procedendo desta forma, obtém-se os parâmetros apresentados na tabela 3.2. Os valores de tensão aplicada foram de 20 a 180 por fornecerem uma temperatura suficiente para soldagem dos componentes SMD e estarem dentro da faixa de segurança do forno.

Tabela 3.2 – Dados para modelagem do sistema.

Tensão Aplicada (U)	Ganho (k)	Constante de tempo (τ)	Atraso (T)	Variação de Temperatura ($^{\circ}\text{C}$)
20	0,488	1061	133,1487	9,7594
40	0,7187	768,2146	67,0501	28,7477
60	0,9699	689,1225	44,6396	58,1946
80	1,1587	616,667	43,915	92,695
100	1,2458	472,8427	42,6071	124,5775
120	1,3552	440,9713	36,0064	162,6197
140	1,4511	405,8304	29,2317	203,1495
160	1,501	356,6937	23,4814	240,1539
180	1,5549	342,0024	13,519	279,8747

Fonte: Pesquisa direta (2016).

Após aquisição dos dados, utiliza-se o método dos mínimos quadrados, para ajustar os parâmetros encontrados, em função da tensão aplicada, por um polinômio (BASÍLIO, 2004) e por meio da aproximação de padé é possível modelar o efeito de atraso do sistema por uma função racional. Deste modo, consegue-se representar o forno pelo modelo adotado, por meio de uma única função de transferência. Com o auxílio do software MATLAB, os seguintes polinômios foram encontrados, o qual descrevem a variação do ganho (k_1), das constantes de tempo (τ_1) e do atraso (T) em função da tensão aplicada (U).

$$\begin{aligned}
 k_1(U) &= 1.51136364 \times 10^{-7}U^3 - 8.3980519481 \times 10^{-5}U^2 + 1.8026482683983 \times 10^{-2}U \\
 &\quad + 0.149426190476187 \\
 \tau_1(U) &= -7.22986269 \times 10^{-4}U^3 + 2.71852853084 \times 10^{-1}U^2 - 32.854181525974U \\
 &\quad + 1654.806697619057 \\
 T(U) &= 1.5039015 \times 10^{-6}U^4 - 7.171282407 \times 10^{-4}U^3 + 1202668377525U^2 \\
 &\quad - 8.5496049031987U + 260.6455444444443
 \end{aligned} \tag{3.1}$$

Assim, de acordo com os polinômios em 3.1, um modelo do forno para simulação é dado por:

$$G_{aq}(s) = \frac{K_1(U)e^{-T(U)s}}{\tau_1(U)s + 1} \tag{3.2}$$

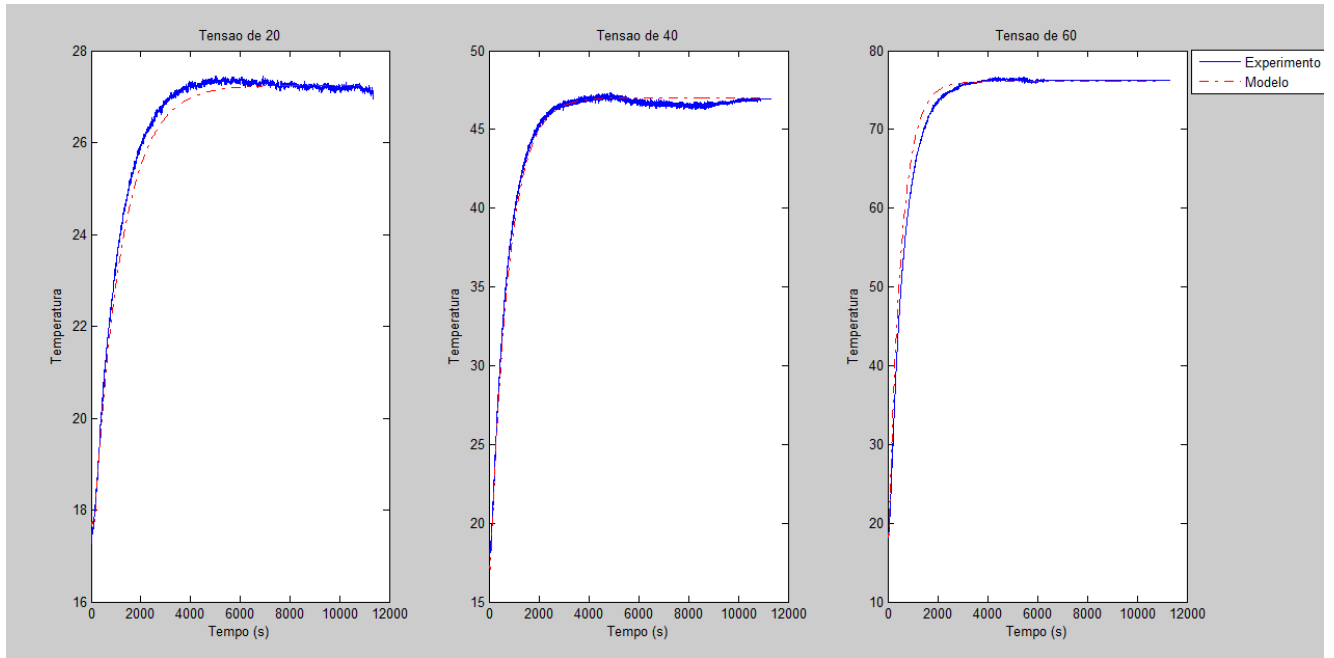
Onde $K_1(U)$, $T(U)$ e $\tau_1(U)$ são descritos pelas equações em 3.1 e $G_{aq}(s)$, definido na equação 3.2 denota a função de transferência para o período de aquecimento. Para representar o atraso de tempo ($e^{-T(U)s}$) da função de transferência foi utilizado a função do MATLAB que representa o aproximante de padé.

3.4 Validação do modelo

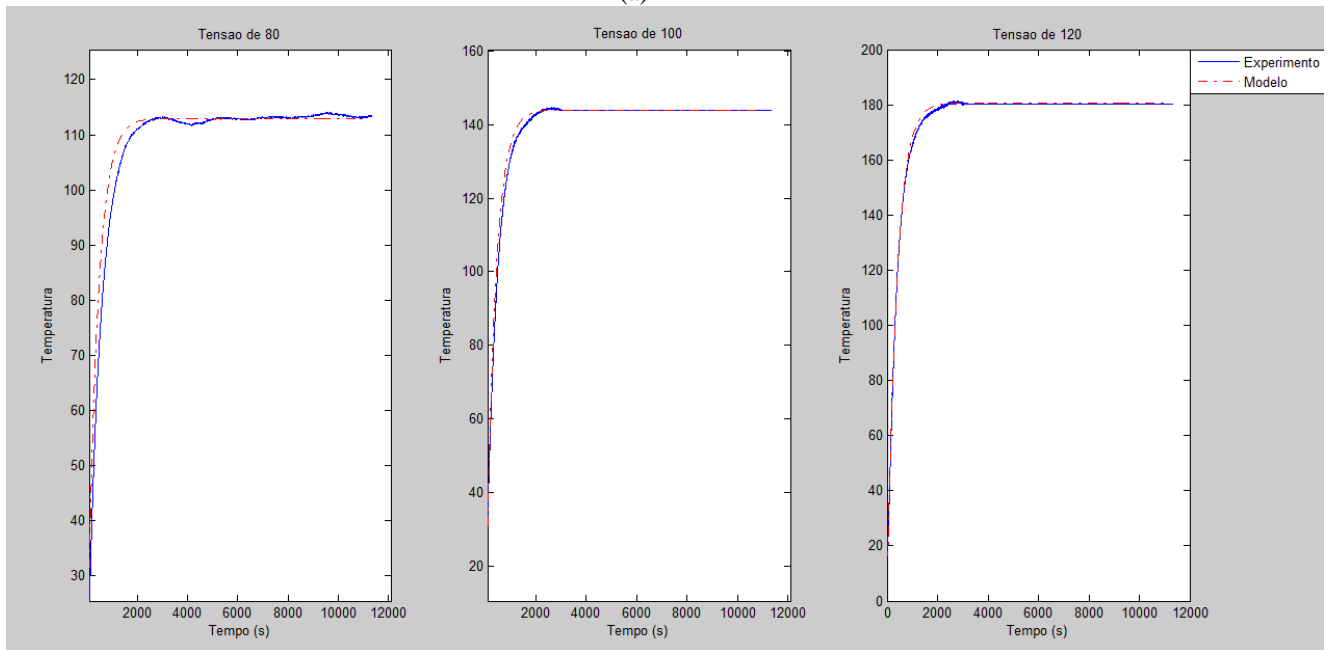
O modelo obtido deve ser capaz de representar corretamente o forno para que se possa realizar o projeto do controlador de maneira adequada. Deste modo, deve-se validar o modelo encontrado antes de passar para ao projeto do controlador.

Neste trabalho, a validação foi feita utilizando-se o ambiente Simulink do software MATLAB. Foi aplicado ao modelo encontrado, os mesmos sinais que foram anteriormente utilizados no sistema

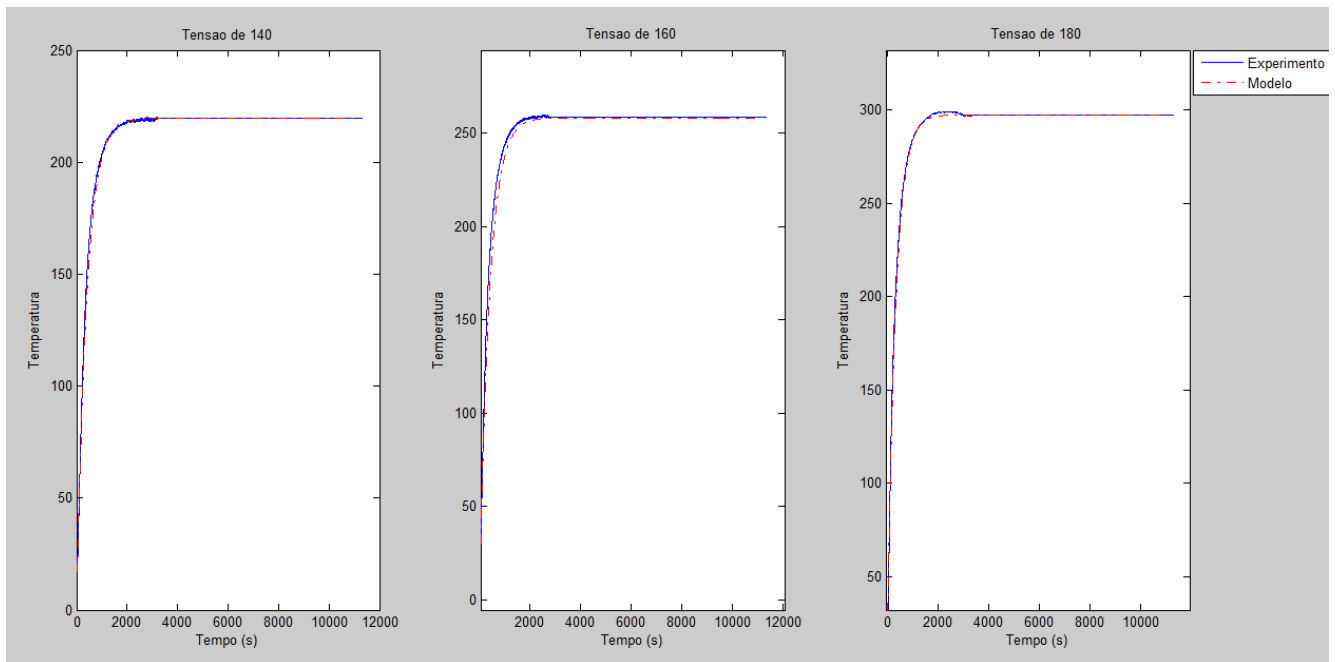
real e registrou-se as respostas. Estas respostas foram comparadas com as curvas de temperatura obtidas no procedimento 3.1. É importante ressaltar que embora a tensão a ser aplicada ao forno seja do tipo senoidal, na simulação o sinal a ser aplicado foi do tipo degrau, com amplitude igual ao valor eficaz da tensão senoidal que foi aplicada no sistema real. A figura 3.6 ilustra as curvas dos resultados experimentais, comparados com o modelo matemático.



(a)



(b)



(c)

Figura 3.6 – Resultado experimental vs modelo matemático para as tensões de 20 a 60 (a); 80 a 100 (b); 120 a 180 (c).
Fonte: Pesquisa direta (2016).

3.5 Projeto do controlador

O controlador consiste em um microcontrolador (PIC18F4550) que lê as temperaturas do LM35 e do termopar tipo K, medida pela placa amplificadora de tensão. Por se tratar de um sistema de soldagem de componentes SMD, as temperaturas alvo já foram definidas na programação de acordo com o tipo de solda selecionada no menu. A partir do erro entre os valores lidos pela placa amplificadora de sinal e as temperaturas alvo, um controlador PI calcula a razão cíclica para controlar um TRIAC a partir de um sinal de *PWM*. O TRIAC é responsável pelo ajuste da tensão de alimentação que o forno irá receber. Desta forma é controlada a temperatura no interior do forno.

Para gerar o *clock* foi montado um circuito com dois capacitores de 22pF e um cristal de 20MHz conectados aos pinos OSC1 e OSC2. Foi disponibilizada na placa as portas A0, A1, A2 e A3 que correspondem as entradas AN0, AN1, AN2 e AN3 do módulo conversor A/D como opções para a leitura das temperaturas LM35 e do termopar tipo K. As portas B1, B5, B6 e B7 foram escolhidas para serem as entradas das teclas seleciona(*enter*), volta(*c*), baixo (*b*) e cima(*a*), respectivamente. A porta B1 usa uma interrupção externa enquanto as portas B6 a B7 usa a interrupção RB. As portas D4 à D7 fazem a comunicação de 4 bits com o mostrador LCD e as portas D0 e D1 são as

saídas dos sinais E, *enable* que inicia leitura e gravação de dados, e RS, *registers select* que seleciona registradores para configurar o LCD. A conexão USB é realizada pelas portas C4(D-) e C5(D+). As portas B0 e E2 foram conectadas com o foto-acoplador 817B e o foto-triac MOC3021 respectivamente.

O esquema elétrico foi feito no *software* PROTEUS. As conexões das portas do microcontrolador é ilustrado na figura 3.7.

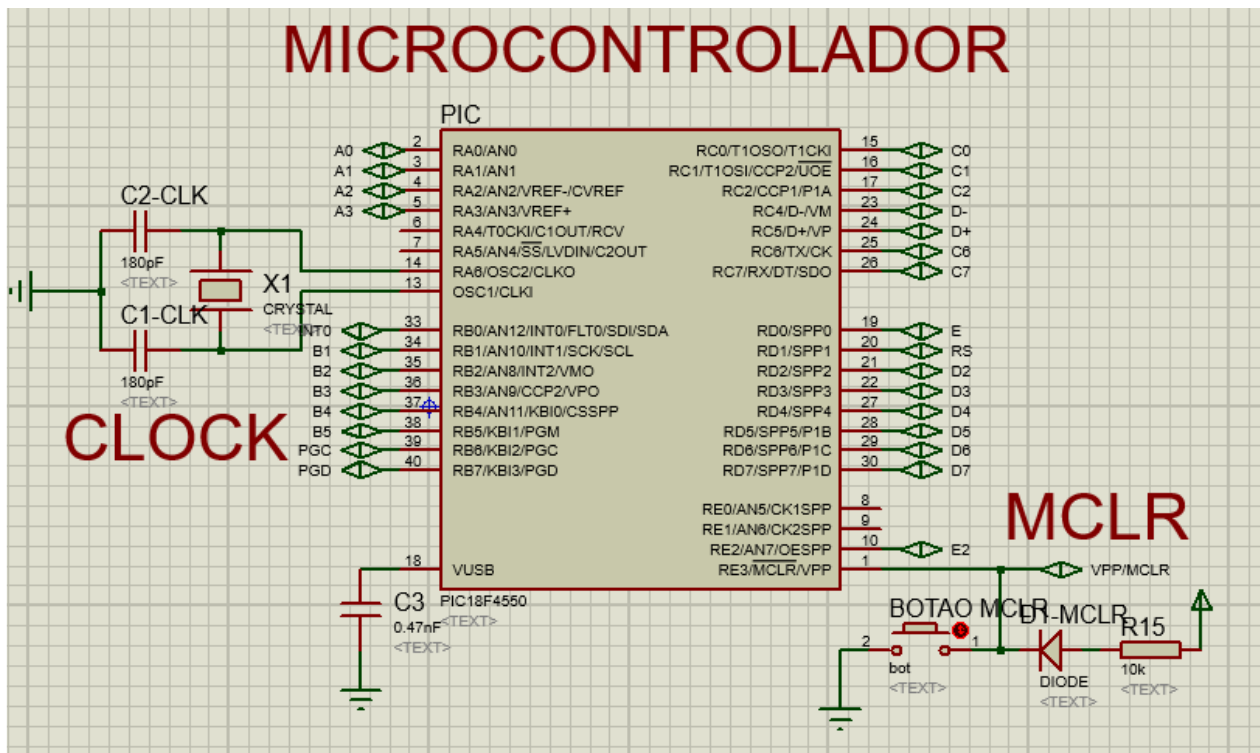


Figura 3.7 – Esquema elétrico microcontrolador.

Fonte: Pesquisa direta (2016).

O circuito realiza a função de controlar a tensão eficaz aplicada a uma carga por meio do corte em ângulo de fase. Para que tal tarefa seja possível deve-se fazer a sincronismo da senoide da rede com as tarefas executadas pelo microcontrolador. A sincronia é feita por meio de um circuito chamado “*zero-cross detection*”, que é composto por uma ponte de diodos que gera um sinal contínuo pulsante contendo a parte positiva da senoide e invertendo a parte negativa. Este sinal que possui uma frequência de 120HZ é um “espelho” da senoide original onde este sinal chega em 0 exatamente quando a senoide cruza o 0 (*zero-cross*). O sinal contínuo pulsante é aplicado em um acoplador óptico (817B) que vai fazer a conexão deste circuito de potência com o controle onde serão gerados pulsos de 5V de amplitude e frequência de 120Hz que definem a posição da senoide.

O esquema de montagem do LCD é ilustrado na figura 3.9. Por ser escolhida a comunicação de 4 bits, os pinos D0 à D3 do LCD não são utilizados. Por não ser necessária a leitura do LCD, o pino R/W, *read/write*, é aterrado. Um potenciômetro de 10k Ω é utilizado para variar o nível de tensão do pino que regula o contraste do LCD.

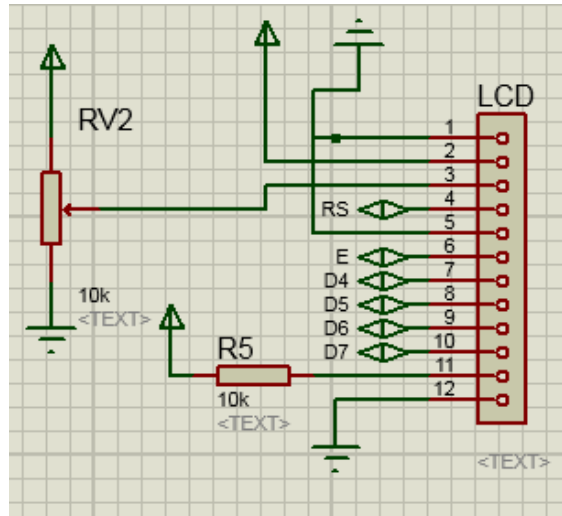


Figura 3.9 – (a) Circuito elétrico do LCD.
Fonte: Pesquisa direta (2016).

3.5.1 Firmware do controlador

O firmware implementado no microcontrolador foi escrito em linguagem de programação C, utilizando o compilador CCS por meio do ambiente de programação PCW. Inicialmente é declarado os bits da porta ADC, os fuses e as variáveis utilizadas. São configuradas três interrupções, sendo elas a interrupção externa, o timer 0 e o timer 1.

No microcontrolador o disparo do TRIAC ocorre através da variação de um valor de “set-point” que é definido através da variável inteira *INTENSITY* (0 a 250) que depende de uma tensão de 0 e 5V aplicada no conversor A/D. O microcontrolador através de seu timer 0 decrementa a variável *TRIGGER* a cada 33,3 μ s ou 250 vezes dentro de um semiciclo. Toda vez que ocorre o pulso de zero-cross a interrupção externa é executada e com isso *TRIGGER*=250 e o programa volta a ser executado. Assim o microcontrolador basicamente faz o seguinte: decrementa *TRIGGER*, compara *TRIGGER* com *INTENSITY*, a partir do momento que *TRIGGER* se torna menor ou se iguala a *INTENSITY* o microcontrolador dispara uma sequência de pulsos no *gate* do TRIAC pelo pino E2 até que o semiciclo da senoide acabe, repetindo essa sequência a cada semiciclo da senoide, ou seja, a cada 8,33 ms.

A variável *TRIGGER* conta de 0 a 250 dentro de cada semiciclo, então a variação de *INTENSITY* determina o ponto(ângulo) exato onde o TRIAC deve ser disparado. Ao disparar o TRIAC em diferentes ângulos o circuito realiza o controle da tensão eficaz aplicada à carga que vale:

$$VO_{ef} = \sqrt{\frac{1}{\pi} \int_{\alpha}^{\pi} (V_i \cdot \sin(\theta))^2 \cdot d\theta} = V_i \cdot \sqrt{\frac{1}{2} - \frac{\alpha}{2\pi} + \frac{\sin(2\alpha)}{4\pi}} \quad (3.3)$$

No timer 01 é configurado a rotina para controlar o tempo de cada fase de soldagem. O perfil de soldagem a ser seguido é configurado no menu “Solda” que mudar os valores dos set-points e tempo de cada fase de soldagem de acordo com o tipo de solda selecionada (SnPb ou SnAgCu). A programação completa do firmware pode ser visualizada no Anexo 01.

3.5.2 Placa de Circuito impresso

Finalizado o projeto do circuito eletrônico, deu-se segmento à confecção da parte física do projeto por meio da aplicação ARES, que faz parte do *software* Proteus. A figura 3.10 ilustra o projeto da placa e a figura 3.11 ilustra o resultado da placa após os componentes serem colocados e soldados.

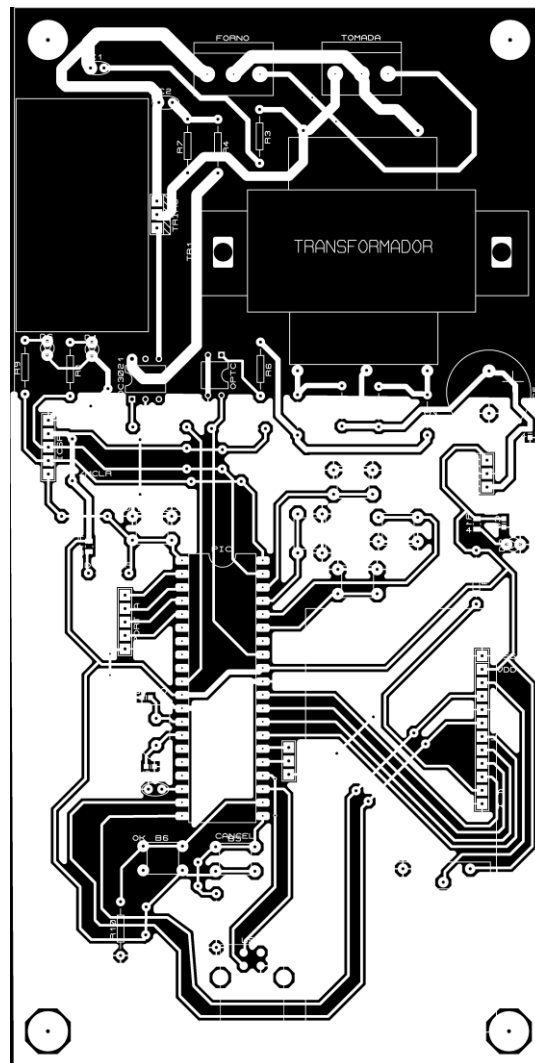


Figura 3.10 – Projeto da PCI feito no Ares – Proteus.
Fonte: Pesquisa direta, 2016.



Figura 3.11 – Placa controladora.
Fonte: Pesquisa direta (2016).

3.5.3 Parâmetros PID do controlador

O diagrama de blocos da figura 3.12 mostra os elementos básicos de um controlador PID em malha fechada. A planta e o controlador são conectados em um elo de realimentação negativa.

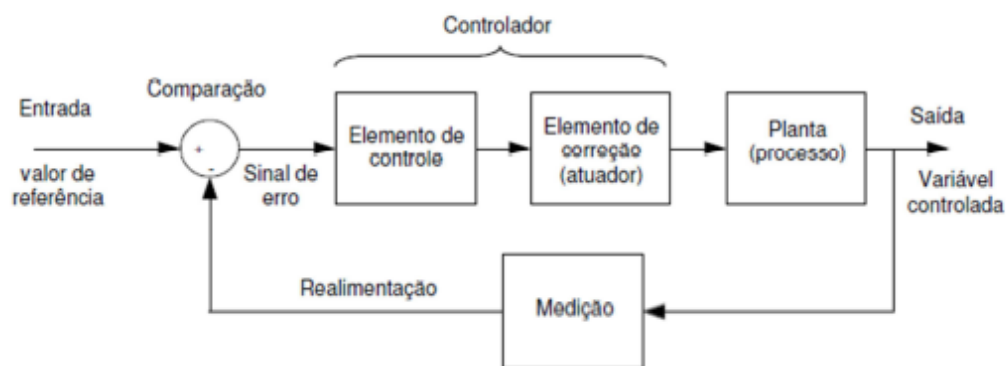


Figura 3.12 – Diagrama de blocos do controlador PI em malha fechada.
Fonte: Bezerra (2010).

O forno deve operar em diferentes pontos de operação para seguir o perfil *reflow* desejado. De acordo com as recomendações da Altera (2011), figura 2.2 e tabela 2.1 da seção 2.1.1, os pontos de operação devem ser 100°C, 150°C, 183°C e 220°C, respectivamente e devem ser respeitados os limites de tempo e inclinação da curva de aquecimento para cada fase de aquecimento.

A temperatura do forno no primeiro ponto de operação tem que ser 100°C. Para uma temperatura ambiente igual a 21°C, o ganho $K(U)$ deverá ser igual a

$$K(U) = \frac{\Delta\theta}{U} = \frac{100 - 21}{U} = \frac{79}{U} \quad (3.4)$$

Substituindo a equação 3.4 na equação 3.1(a) e resolvendo-se para B, obtém-se $U = 73,2427 \text{ Volts}$. O forno foi alimentado com uma tensão de 73,2427 *Volts*. Aplicando um degrau de 10% a 20% do ponto de operação e utilizando o método das áreas (capítulo 2, seção 2.3.1) foram obtidos os parâmetros da curva de resposta, dentre eles a constante de tempo e o atraso. Utilizando a tabela de Ziegler e Nichols (capítulo 2, seção 2.5.1), foram obtidos os parâmetros para um controle PI e PID:

Parâmetros de controle PI:

$$K_P: 10.7792$$

$$K_I: 0.0850$$

Parâmetros de controle PID:

$$K_P: 13.5071$$

$$K_I: 0.1538$$

$$K_d: 296.5834$$

4 RESULTADOS

Este capítulo tem a finalidade apresentar os resultados obtidos quanto aos controladores utilizados e do perfil *reflow* atingido. Também é feita uma análise das etapas obtidas durante a soldagem *reflow* e é justificada a escolha do melhor controlador para este sistema.

4.1 Teste dos controladores

Foi analisado o controle PI e PID para a primeira fase de soldagem. O controle PI conseguiu alcançar e manter a temperatura do *set point*, porém o tempo de resposta excedia o tempo limite para a primeira fase, que foi estipulado de acordo com a descrição feita no item 2.1.1. O controle PID teve um overshoot de 2%, que não influencia no resultado da solda, desta maneira ele foi o melhor controlador conseguindo alcançar e manter a temperatura do *set point* dentro do limite de tempo estabelecido. A figura 4.1 ilustra o controle PI e PID na primeira fase de soldagem.

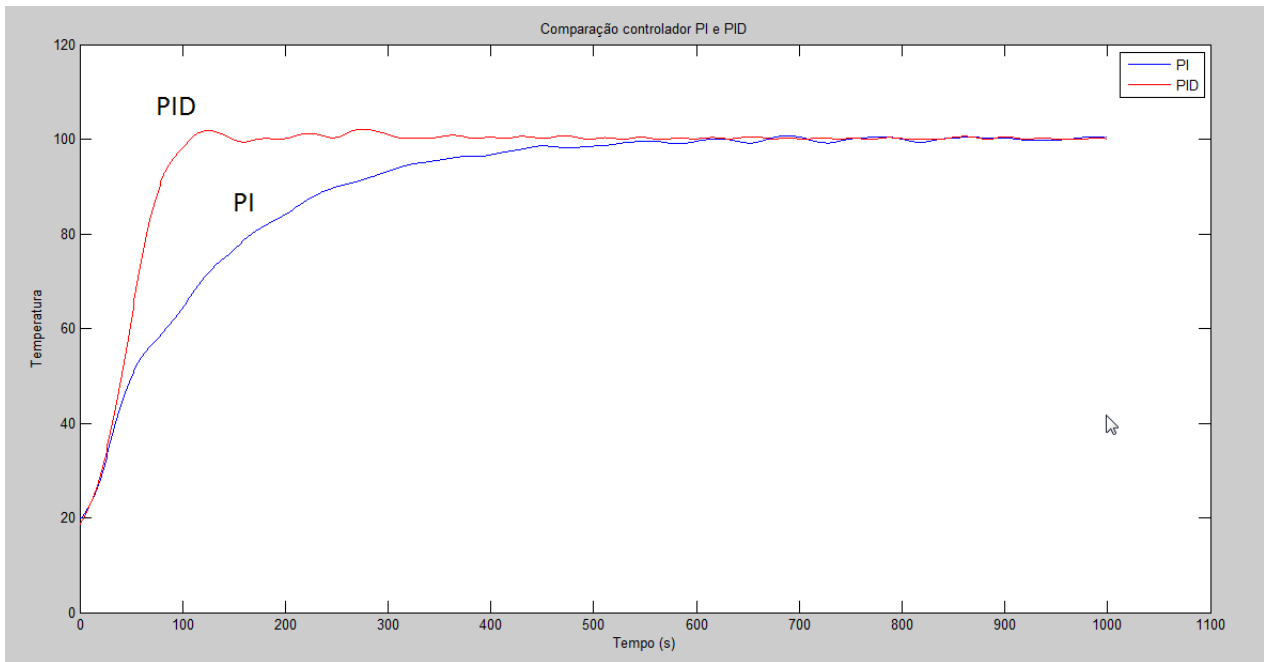


Figura 4.1 – Comparação entre o controle PI e PID.
Fonte: Pesquisa direta (2016).

Para descobrir os parâmetros dos outros pontos de operação, pode-se utilizar o mesmo procedimento feito na seção 3.5.2, porém como visto no trabalho de Guerra (2008), os controladores projetados para tensões menores são mais trabalhosos e seus parâmetros também serviram para os controladores em tensões mais elevadas. Uma evidência disto, é que o modelo

matemático encontrado representou melhor tensões mais elevadas, como pode-se ver na figura 3.6. Desta maneira o controlador projetado para 100°C foi utilizado nos demais ponto de operação.

4.2 Perfil reflow projetado

O controlador projetado neste trabalho foi capaz de conseguir seguir o correto perfil de temperatura para soldagem eficaz de componentes SMD. O perfil *reflow* projetado foi para uma solda composta por estanho e chumbo (SnPb), ou seja, as etapas deveriam respeitar as restrições da segunda coluna da tabela 2.1. A imagem 4.2 ilustra a curva de aquecimento obtida pelo controlador.

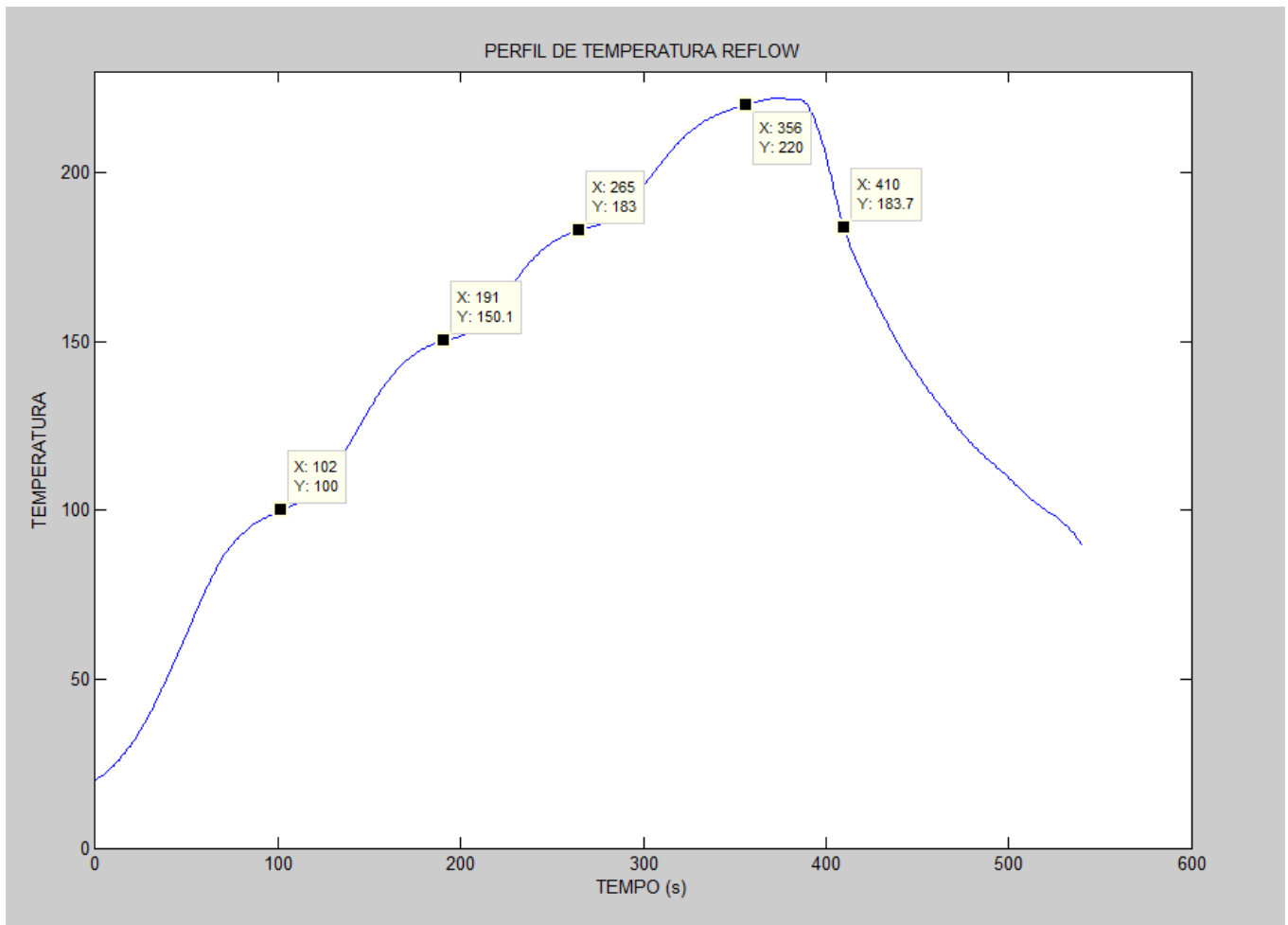


Figura 4.2 – Perfil de temperatura obtido com o controlador PID.
Fonte: Pesquisa direta, 2016.

A tabela 4.1 faz uma comparação entre as principais características do perfil reflow recomendado e o perfil reflow obtido.

Tabela 4.1 – Comparação entre perfil reflow recomendado e obtido.

Características	Perfil reflow recomendado	Perfil reflow projetado
Temperatura mínima na fase de pré-aquecimento	100°C	100°C
Temperatura máxima na fase de pré-aquecimento	150°C	150°C
Tempo durante fase de pré-aquecimento	60-120 segundos	90 segundos
Temperatura durante fase de liquefação	183°C	183°C
Tempo acima da fase de liquefação	60-150 segundos	144 segundos
Temperatura de Pico	220°C	221.9°C
Tempo entre a temperatura ambiente e temperatura máxima	360 segundos	356 segundos

Fonte: Pesquisa direta, 2016.

5 CONCLUSÕES E RECOMENDAÇÕES

Este capítulo mostra as conclusões obtidas com a finalização do trabalho e sugestão para trabalhos realizados no futuro.

5.1 Conclusões

O forno adotado neste trabalho foi adequado para a proposta principal, já que conseguiu chegar nas temperaturas adequadas respeitando o limite de tempo para soldagem dos componentes conforme evidenciados nos resultados apresentados.

Também se concluiu que um controlador PI não foi o suficiente, visto que sua resposta excedia o tempo máximo ao qual os componentes podiam ficar expostos à temperatura, sendo necessário projetar um controlador PID para conseguir controlar o forno de modo a obter o perfil *reflow* desejado.

O presente trabalho conseguiu com sucesso adaptar um forno resistivo para realizar a soldagem de componentes SMD, seguindo o perfil *reflow*, deixando um equipamento que irá auxiliar na confecção de placas de circuito impresso.

5.2 Recomendações para trabalhos futuros

Para trabalhos futuros sugere-se o desenvolvimento de um supervisor que se comunique com a PCI. Desta maneira, será mais simples acompanhar e analisar os perfis de temperatura obtidos, além de ser mais prático modificar as referências de temperatura e os parâmetros PID.

O firmware do controlador realiza o perfil de soldagem somente para solda com chumbo e sem chumbo (SnPb ou SnAgCu). Desta maneira, recomenda-se modificar o firmware do microcontrolador de forma que possa ser definida as temperaturas alvo e o intervalo de tempo entre cada uma delas, para ampliar a aplicabilidade do sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGUIAR, F. L. A., JUNIOR W. I. M. **Ajuste de curvas por quadrados mínimos lineares**. s.d.
- ALMEIDA, J. L. A. **Dispositivos Semicondutores: TIRISTORES**. 8. ed. São Paulo, 2003.
- ALTERA CORPORATION, T. **SMT Board Assembly Process Recommendations**. Outubro 2011.
- ÅSTRÖM, K. J.; HÄGGLUND, T. **PID Controllers: Theory, Design, and Tuning**. 2ª Ed. Carolina do Norte: Instrument Society of America, 1995.
- BASILIO, J. C. **Laboratório de Sistemas de Controle I, 3ª edição**. UFRJ, Escola Politécnica, Departamento de Eletrotécnica, Rio de Janeiro, 2004.
- BASILIO, J. C., MANHÃES, R. R., ROLIM, L. G. B. **Controle de temperatura de um forno elétrico a resistência utilizando a função PID de um controlador lógico programável**, Anais do 14º Congresso Brasileiro de Automática, p.2184-2189, Natal, Brasil, 2002.
- BAZANELLA, S. A., SILVA, J. M. G. **Ajuste de Controladores PID**. Universidade Federal do Rio Grande do Sul, Curso de Extensão, s.d.
- CR resistências. **Tabela de modelos e cores**. Disponível em <<http://cr-resistencias.com.br>> Acesso em Março de 2016.
- GOOSENS, P. **Controlador para forno de componentes SMD**. Elektor, Setembro de 2006.
- GUERRA, L. N. **Uso de compensador PID no controle da taxa de variação de temperatura em um forno elétrico a resistência**. Universidade Federal do Rio de Janeiro, Outubro de 2006. 44p.
- IBRAHIM, D. **Advanced PIC Microcontroller Projects in C: From USB to Zigbee with the PIC 18F Series**. Amsterdam; Boston: Newnes/Elsevier, 2008. 560p.
- IBRAHIM, D. **Microcontroller Based Applied Digital Control**. Chinchester: John Wiley & Sons, Ltd, 2006. 313p.
- KEMPENICH, G. **Curso de Projetos de Instrumentação**. Instituto Mauá de Tecnologia, São Caetano do Sul, SP. Apostila, 22p. s.d.

LOURENÇO, I. M. L. S. **Metodologia Analítica para medir e especiar o carbono: Controle de Temperatura**. Universidade de Aveiro, 2008. 59p

LOURENÇO, J. **Sintonia de controladores PID**. Escola Superior de Tecnologia, 1997.

MIYADAIRA, A. N. **Microcontroladores PIC18: Aprenda e Programe em Linguagem C**. 1ª Ed. São Paulo: Érica, 2009. 400p

MONTEIRO, M. A. **Introdução à organização dos computadores**. 3ª Ed. Rio de Janeiro: LCT, 1996.

NATIONAL INSTRUMENTS CORPORATION. **Datasheet USB-6009**. 2014. Disponível em <<http://www.ni.com/datasheet/pdf/en/ds-218>>. Acesso em Março de 2016.

NAVARRO, M. C. K. A., NAVARRO, C. A., FERREIRA, R. C., TERAMON N. **Os Aproximantes de Padé**. Matemática Universitária nº26/27 – junho/dezembro 1999 – pp. 49-66.

NEVES, F. **Controlador PID digital: Uma modelagem prática para microcontroladores**. Maio de 2014. Disponível em:<<http://www.embarcados.com.br/controlador-pid-digital-uma-modelagem-pratica-para-microcontroladores-parte-1/>> Acesso em: Junho de 2016.

OGATA, K. **Engenharia de Controle Moderno**. 4ª Ed. São Paulo: Prentice Hall, 2003.

OGATA, K. **Engenharia de Controle Moderno**. Rio de Janeiro: Prentice Hall, 1982.

PEREIRA, F. **Microcontroladores PIC: Técnicas Avançadas**. 1ª Ed. São Paulo: Érica, 2002. 358p

PROTOLAB. **Termopar**. Disponível em: <<http://www.protolab.com.br/Termopar.pdf>> Acesso em: Março de 2016.

SENA, A. S. **Microcontroladores**. Microchip, 253p. s.d.

SOUZA, D; LAVINIA N. **Conectando o PIC – RECURSOS AVANÇADOS**. 4ª Ed. Editora Érica, 2005. 380p.

TEXAS INSTRUMENTS. **Datasheet LM35**. 2013. Disponível em <[HTTP://www.ti.com](http://www.ti.com)>. Acesso em Março de 2016.

ANEXO 01
(FIRMWARE DO MICROCONTROLADOR)

```

// PROGRAMAÇÃO FORNO REFLOW
// Desenvolvedor: Daniel Augusto Salles Cordeiro
// Universidade Federal de Ouro Preto

//////// DEFINIÇÃO DO MICROCONTROLADOR , BIBLIOTECAS USADAS E PORTS //////////

#include <18F4550.H>
#define ADC = 10
#define delay (clock=48000000)
#define fuses
HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGEN,NO
PUT,NOBROWNOUT,NOCPU

#include "LCD4B.c"
#include <stdlib.h>

#define fast_io(a)
#define fast_io(b)
#define fast_io(c)
#define fast_io(d)
#define fast_io(e)

#define porta=0xF80 /*endereço das portas, retirados do datasheet, pág. 68 no Adobe Reader: no
caminho Memory Organization, Data Memory Organization, Special Function Registers*/
#define portb=0xF81
#define portc=0xF82
#define portd=0xF83
#define porte=0xF84

//////////////////// DECLARAÇÃO DAS VARIÁVEIS //////////////////////

#define LOWTOHIGH TRUE
#define TR_DELAY delay_us(5) /* Delay para disparo triac */

```

```

#define ZERO_IN      PIN_B0      /* Pino de detecção de "zero cross" */
#define TRIAC        PIN_E2      /* Pino de saída para o controle de fase/ disparo do triac
*/

#define ON            output_high /* Seta a saída em nível alto */
#define OFF           output_low  /* Seta a saída em nível baixo */

// Variaveis Dimmer
int TRIGGER;          // declara variável TRIGGER como int
float INTENSITY;

//Termopar Tipo K e LM35
float temp_atual1;
float temp_atual2;
float valor_real;

//Controlado Proporcional-Integral
int8 Ho,h0,h1,h2,h3,h4,hT;    // SET-POINT
float PID = 0;
float PID_AUX = 0;
float ErroAnterior = 0;
float Kp = 13.5071;    //Ganho proporcional
float Ki = 0.1538;    //Ganho integral
float Kd = 296.5834 ; //Ganho Derivativo
float erro = 0; //Erro
float soma_erro = 0;

//Timer 1
int8 contaT1_1 = 0;
signed int16 contaT1_2 = 0; // contaT1_2 = 0;
int8 contaT1_3 = 0;
int8 contaT1_4 = 0;

```

```

int8 aux_cont_fase2 = 0;
int8 aux_cont_fase5 = 0;

//Variaveis Auxiliares
float AUX_TOTAL, AUX;
int i;
float pid2;
int etapa;
int aux1,aux3,aux_flag=1,aux_limp=0;
int aux_sel_solda = 0, aux_finaliza=0;
int Inicia_contagem = 0, volta_tela=0;

// Variáveis da interrupção RB
unsigned int now=0,last=0, enter=0, a=0, b=0, c=0;

// Funções chamadas no programa
void Menu_Solda();
void Menu_Tela();
void Limpeza();
void Menu_Cancel();

// Variáveis para fazer o debounce
#define t_filtro 15
long int filtro0=t_filtro; // inicia filtro do bot1
long int filtro1=t_filtro; // inicia filtro do bot2
int flags=0;
#define bt_UP_press = flags.0
#define bt_DOWN_press = flags.1
////////////////////// FUNÇÕES ////////////////////////
void Limpeza()
{

```

```

    if(aux_limp==1)
    {
        lcd_pos_xy(1,1);
        printf (lcd_escreve,"          ");
        lcd_pos_xy(1,2);
        printf (lcd_escreve,"          ");
    }
    aux_limp=0;
}

void Controlador()
{
    Erro = Ho - valor_real; //Ho -> SetPoint // sensorPin3 -> Leitura da altura do tanque1 em
tempo real
    if (erro>0)
    {
        soma_erro = soma_erro + erro;
        if (soma_erro>=255)
        soma_erro=255;
        if (soma_erro<=0)
        soma_erro=0;
        PID = PID + ((Ki*soma_erro)+(Kp*Erro)+kd*(erro-ErroAnterior));
        PID_AUX = PID;
        if(PID<0) PID=0;
        if(PID>255) PID=255;
        ErroAnterior=Erro;
        pid2 = PID/255;
        INTENSITY = pid2*250;
    }
    else
    {
        INTENSITY = 0;
    }
}

```

```

    soma_erro = 0;
}
}

void Leitura_AD() //Função que realiza leituras dos canais AD do uC.
{
    //ADC LM35
    set_adc_channel(1);
    delay_us(10);
    AUX_TOTAL=0;
    for(i=0; i<50; i++)
    {
        AUX = read_adc();    // inicia conversão AD
        AUX_TOTAL += AUX;
    }
    temp_atual1 = (((AUX_TOTAL/50)*500)/1023);
    //-----
    //ADC TERMOPAR
    set_adc_channel(3);
    delay_us(10);
    AUX_TOTAL=0;
    for(i=0; i<50; i++)
    {
        AUX = read_adc();    // inicia conversão AD
        AUX_TOTAL += AUX;
    }
    temp_atual2 = (((AUX_TOTAL/50)*1168.22)/1023);
    valor_real = temp_atual2 + temp_atual1;
    // EXPLICAÇÃO DA ROTINA:
    // O For realiza um loop e pega 50 vezes o valor.
    // Depois esse é feita um media e realizado o calculo.

```

```

// O calculo é realizado da seguinte forma:
// Temp = ( Valor lido / 1023 ) x 1168.22
// Como o Termopar gera aproximadamente 41uV / °C e o ganho fornecido eh 104.6099
// Temos que : (104.6099)*(0.041mV) = 0.00428V
// 5V / 0.00428V = 1168.22
}
void fase()
{
if(etapa==1) ho=h0;
if(etapa==2) ho=h1;
if(etapa==3) ho=h2;
if(etapa==4) ho=h3;
if(etapa==0) ho=h4;
}

////////////////////// INT RB ////////////////////////
#INT_RB
void trata_RB()
{
    now = input_b();
    #if LOWTOHIGH
    if ((!bit_test(last,5))&&(bit_test(now,5)))
    {
        a=0;
        b=0;
        c=1;
    }
    if ((!bit_test(last,7))&&(bit_test(now,7)))
    {
        a=0;
        b=1;
        c=0;
    }
}

```

```

    }
    if ((!bit_test(last,6))&&(bit_test(now,6)))
    {
        a=1;
        b=0;
        c=0;
    }
    #endif
    last=now;
}

////////////////////////////////// INT EXT ////////////////////////////////////
#INT_EXT
void ext_isr()
{
    TRIGGER=250;
}
#INT_EXT1
void TRATA_EXT1()
{
    enter=1;
    delay_ms(200);
}

////////////////////////////////// TIMER 0 ////////////////////////////////////
#INT_TIMER0 // Frequencia em 30 kHz
void tmr0_isr()
{
    set_timer0(156+ get_timer0());
    TRIGGER--;
    if (TRIGGER<=INTENSITY)
    {

```



```

    ON(TRIAC);
    TR_DELAY;
    OFF(TRIAC);
  }
}

////////////////////////////////// TIMER 1 ////////////////////////////////////
#int_timer1
void trata_t1 ()
{
  // reinicia o timer 1 em 15536 mais a contagem que já passou
  set_timer1(15536 + get_timer1());
  contaT1_1 += 1;
  If (contaT1_1 == 30) //Se já passou um segundo:
  {
    contaT1_2 = contaT1_2 + 1;
    if(aux_cont_fase2==1)
    contaT1_3++;
    if(aux_cont_fase5==1)
    contaT1_4++;
    contaT1_1 = 0;
    //Leitura_AD();
    //Controlador();
    //fase();
  }
  if(Inicia_contagem == 1)
  {
    //COMEÇANDO A SOLDAR
    if(contaT1_4<=hT)
    {
      If(contaT1_2<h0 ) etapa=1;
      //2 FASE

```

```

if(contaT1_2>=h0 && valor_real<h1) {etapa=2;aux_cont_fase2=1;}
// 3 FASE
If(valor_real >= h1 && valor_real <h2 && contaT1_3>=60)
{etapa=3;aux_cont_fase2=0;contaT1_3=0;}
// 4 FASE
If(valor_real >= h2 && valor_real < h3) { etapa=4;aux_cont_fase2=1;}
// ULTIMA FASE
If(valor_real >= h3){aux_cont_fase5=1;}
}

else
{
etapa = 0;
INTENSITY=0;
Inicia_contagem = 0;
aux1=0;
aux3=0;
}
}
}

//////////MUDANÇA DE ESTADO APERTANDO OS BOTÕES//////////
void Menu_Principal()
{
int aux1=1;

while(aux1)
{

if(aux_finaliza==0)
{
printf (lcd_escreve,"%f");

```

```
lcd_pos_xy(1, 1);
printf (lcd_escreve," INICIAR SOLDAGEM ");
delay_ms(250);
}
if(enter==1)
{
Inicia_contagem = 1;
volta_tela=0;
Menu_tela();
}
if(b==1)
{
int aux3=1;
while(aux3)
{
a=0;
b=0;
printf (lcd_escreve,"\f");
lcd_pos_xy(1, 1);
printf (lcd_escreve,"SELECIONAR SOLDA");
delay_ms(250);
if(enter==1)
{
enter=0;
aux1=0;
aux3=0;
Menu_Solda();
}
if(a==1)
{
a=0;
```

```

    aux1=1;
    aux3=0;
    }
    }
    }
    }
    }

void Menu_Solda()
{
int aux1=1;
if(aux_sel_solda==0)
b=0;
else
b=1;
While(aux1)
{
if(aux_flag==1&&b==0)
{
printf (lcd_escreve,"\f");
lcd_pos_xy(1, 1);
printf (lcd_escreve,"[X]SnPb" );
lcd_pos_xy(1, 2);
printf (lcd_escreve,"[ ]SnAgCu" );
delay_ms(250);
}
if(c==1)
{
c=0;
aux1=0;
}
}

```

```

if(enter==1)
{
    enter=0;
    c=1;
    aux_flag=0;
    aux_sel_solda=0;
    printf (lcd_escreve, "\f");
    lcd_pos_xy(1, 1);
    printf (lcd_escreve, "CONFIGURADO !" );
    h0=100;
    h1=150;
    h2=183;
    h3=220;
    h4=0;
    hT=20;
    delay_ms(500);
}
if(b==1)
{
    int aux2=1;
    while(aux2)
    {
        a=0;
        b=0;
        if(aux_flag==1)
        {
            printf(lcd_escreve, "\f");
            lcd_pos_xy(1, 1);
            printf (lcd_escreve, "[ ]SnPb" );
            lcd_pos_xy(1, 2);
            printf (lcd_escreve, "[X]SnAgCu" );

```

```
        delay_ms(250);
    }
    if(c==1)
    {
        c=0;
        aux1=0;
        aux2=0;
    }
    if(enter==1)
    {
        enter=0;
        c=1;
        aux_flag=0;
        aux_sel_solda=1;
        printf (lcd_escreve,"%f");
        lcd_pos_xy(1, 1);
        printf (lcd_escreve,"CONFIGURADO !" );
        h0=150;
        h1=200;
        h2=217;
        h3=245;
        h4=0;
        hT=30;
        delay_ms(500);
    }
    if(a==1)
    {
        a=0;
        aux1=1;
        aux2=0;
    }
```

```

    }
  }
}
}

```

```
void Menu_tela()
```

```
{
```

```
    int aux1=1;
```

```
    aux_limp=1;
```

```
    Limpeza();
```

```
    enter=0;
```

```
    c=0;
```

```
    b=0;
```

```
    a=0;
```

```
    While(aux1)
```

```
    {
```

```
        lcd_pos_xy (1, 1);
```

```
        printf (lcd_escreve,"TEMP: ");
```

```
        printf (lcd_escreve,"%1.1f ", valor_real );
```

```
        lcd_pos_xy(13,1);
```

```
        lcd_escreve(0xDF);
```

```
        lcd_escreve(0x43);
```

```
        lcd_pos_xy(1,2);
```

```
        printf (lcd_escreve,"FASE: ");
```

```
        lcd_pos_xy(8,2);
```

```
        printf (lcd_escreve,"%d",etapa);
```

```
        if(volta_tela==1)
```

```
            aux1=0;
```

```

    if(c==1)
    {
        c=0;
        Menu_cancel();
    }
}

void Menu_cancel()
{
    int aux1=1;

    While(aux1)
    {
        if(aux_flag==1)
        {
            printf (lcd_escreve,"\f");
            lcd_pos_xy(1, 1);
            printf (lcd_escreve," Cancelar? " );
            lcd_pos_xy(1, 2);
            printf (lcd_escreve,"[X]SIM [ ]NAO" );
            delay_ms(250);
        }
        if(c==1)
        {
            c=0;
            aux1=0;
        }
    }
    if(enter==1)
    {
        c=0;
        enter=0;
    }
}

```



```

    aux1=0;
    Inicia_contagem = 0;
    volta_tela=1;
}
if(b==1)
{
    int aux2=1;
    while(aux2)
    {
        a=0;
        b=0;
        if(aux_flag==1)
        {
            printf(lcd_escreve, "\f");
            lcd_pos_xy(1, 1);
            printf (lcd_escreve, " Cancelar? " );
            lcd_pos_xy(1, 2);
            printf (lcd_escreve, "[ ]SIM [X]NAO" );
            delay_ms(250);
        }
    }
}
if(c==1)
{
    c=0;
    aux1=0;
    aux2=0;
}
if(enter==1)
{
    aux_limp=1;
    Limpeza();
    enter=0;
}

```

```

    c=0;
    aux1=0;
    aux2=0;
}
if(a==1)
{
    a=0;
    aux1=1;
    aux2=0;
}
}
}
}

////////////////////////////////////////ROTINA PRINCIPAL////////////////////////////////////////

void main()
{
    #priority EXT, Timer0,Timer1

    // Configuração do ADC
    setup_adc_ports(AN0_TO_AN3);
    setup_adc(ADC_CLOCK_INTERNAL);
    set_tris_a(0b11111111);
    set_tris_b(0b11111111);
    set_tris_c(0b11000001);
    set_tris_d(0b00001100);
    set_tris_e(0b00000011);
    porta=0x00;      // limpa porta
    portb=0x00;      // limpa portb
    portc=0x00;      // limpa portc

```

```

portd=0x00;      // limpa portd
porte=0x00;      // limpa porte

disable_interrupts(int_RB);
disable_interrupts(INT_TIMER0);
disable_interrupts(INT_TIMER1);
disable_interrupts(INT_EXT);
disable_interrupts(int_EXT1);
disable_interrupts(GLOBAL);

//Configuração do Timer 1
setup_timer_1( T1_INTERNAL | T1_DIV_BY_8 );
set_timer1(15536);
setup_timer_0(RTCC_INTERNAL | RTCC_DIV_8| RTCC_8_BIT);
set_timer0(156);
//Habilita interrupções
enable_interrupts(int_RB);
enable_interrupts(INT_TIMER0); // Liga as interrupções do TIMER 0
enable_interrupts(INT_TIMER1);
enable_interrupts(INT_EXT); // Liga as interrupções externas
enable_interrupts(int_EXT1);
ext_int_edge(L_TO_H); // Configura as interrupções externas para ocorrerem no degrau de
subida
enable_interrupts(GLOBAL); // Libera o registrador de interrupção global e permite que as
interrupções funcionem

//Configuração do LCD
lcd_ini();
delay_ms (2); // aguarda 2 ms
lcd_pos_xy (1, 1);
printf (lcd_escreve," FORNO REFLOW ");
lcd_pos_xy (1, 2);

```

```
printf (lcd_escreve,"          ");
delay_ms (1000);
lcd_pos_xy (1, 1);
printf (lcd_escreve,"          ");
printf (lcd_escreve,"\n          ");
//Inicia loop infinito
while (true)
{
    Menu_Principal();
    aux_flag = 1;
}
}
```