



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



RAFAEL DE OLIVEIRA QUINTÃO

**DESENVOLVIMENTO DE UM CONTROLADOR ADAPTATIVO
DE TEMPERATURA PARA FORNOS ELÉTRICOS A
RESISTÊNCIA APLICADOS NA SOLDAGEM DE
COMPONENTES SMD.**

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

Ouro Preto, 2019

RAFAEL DE OLIVEIRA QUINTÃO

**DESENVOLVIMENTO DE UM CONTROLADOR ADAPTATIVO
DE TEMPERATURA PARA FORNOS ELÉTRICOS A
RESISTÊNCIA APLICADOS NA SOLDAGEM DE
COMPONENTES SMD.**

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

Orientador: Prof. João Carlos Vilela de Castro

Co-orientador: Prof. Vinícius Marinho Silva

Ouro Preto
Escola de Minas – UFOP
02/ 2019

Q71d

Quintão, Rafael de Oliveira.

Desenvolvimento de um controlador adaptativo de temperatura para fornos elétricos a resistência aplicados na soldagem de componentes SMD.
[manuscrito] / Rafael de Oliveira Quintão. - 2019.

73f.: il.: color; graf.; tabs.

Orientador: Prof. MSc. João Carlos Vilela de Castro.

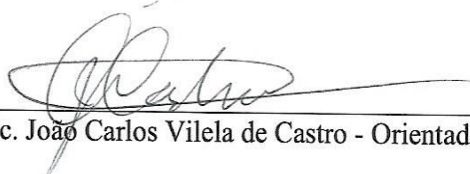
Coorientador: Prof. MSc. Vinícius Marinho Silva.

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

1. Controle adaptativo. 2. Controle de temperatura. 3. Forno resistivo. 4. Soldagem reflow. I. Castro, João Carlos Vilela de. II. Silva, Vinícius Marinho. III. Universidade Federal de Ouro Preto. IV. Título.

Catálogo: ficha.sisbin@ufop.edu.br CDU: 669.162.2

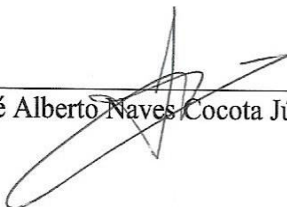
Monografia defendida e aprovada, em 20 de fevereiro de 2019, pela comissão avaliadora constituída pelos professores:



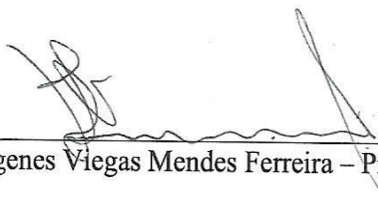
Prof. M. Sc. João Carlos Vilela de Castro - Orientador



Prof. – Professor Convidado - Co-Orientador



Prof. M. Sc. José Alberto Naves Cocota Júnior – Professor Convidado



Prof. M. Sc. Diógenes Viegas Mendes Ferreira – Professor Convidado

AGRADECIMENTO

Agradeço primeiramente aos meus pais, Jordelina e Danilo, por todo apoio e por me ensinarem a importância da educação na vida de uma pessoa. Aos meus irmãos, Arthur e Filipe, pela amizade e companheirismo. À minha namorada Natália, por todo o amor e incentivo. À todo o corpo docente da UFOP pelos ensinamentos. Ao meu professor e orientador João Carlos pela dedicação e paciência. À Fundação Gorceix e à PRACE pelos auxílios. À CAPES pela oportunidade no programa Ciência Sem Fronteiras. À República Taturrodano pelo acolhimento e pela amizade. Muito obrigado a todos que, de alguma maneira, participaram dessa minha trajetória e me ajudaram a tornar esse sonho possível.

“Seja curioso. Leia muito. Experimente coisas novas. O que as pessoas chamam de inteligência apenas se resume a curiosidade.”

Aaron Swartz

RESUMO

Com a evolução da eletrônica, produtos como celulares, computadores, dentre outros, se tornaram cada vez menores e mais leves. Dispositivos com tecnologia SMD (*Surface-Mount Devices*) são os grandes responsáveis por essa evolução, pois seu uso permite um melhor aproveitamento do espaço nas placas de circuito impresso. Devido às suas pequenas dimensões, torna-se difícil a soldagem desses dispositivos por meios convencionais, sendo muito usado para esse fim a solda por forno de refluxo. Neste último, para uma soldagem de qualidade e sem danos aos componentes, a placa com os componentes deve passar por diferentes perfis de temperatura por determinados períodos de tempo. Visto isso o controle realimentado da temperatura dentro deste forno se torna essencial. Esse controlador deve ser capaz de controlar a temperatura com certa precisão em diferentes perfis de temperatura. O problema deste método é que fornos de refluxo são dispositivos de preço relativamente elevado, quando comparados a um forno convencional. Neste trabalho desenvolveu-se um controlador de temperatura para um forno elétrico convencional, transformando-o em um forno de refluxo. Por questões de compatibilidade com os diversos modelos de fornos elétricos à resistência existentes no mercado, o controlador desenvolvido foi do tipo adaptativo, que é capaz de se ajustar automaticamente a diversos modelos sem grandes intervenções do usuário.

Palavras-chave

Controle Adaptativo, Controle de Temperatura, Forno Resistivo, Soldagem Reflow.

ABSTRACT

With the evolution of electronics, products like cellphones, computers, among others, became even smaller and lighter. Devices with SMD (Surface Mount Devices) technology are the main responsible for this evolution, because their use allows a more efficient use of the room on a printed circuit board. Due to their small dimensions it's difficult to solder this kind of device by the conventional ways, being very common the use of reflow soldering ovens. On those, for a good quality solder without damaging the components, the board must pass through different temperature profiles for a specific amount of time. For that, the feedback temperature control inside the oven becomes essential. The controller must be able to control the temperature with a good precision in different profiles. The problem with this method is that reflow soldering ovens are normally very expensive, when compared to a conventional oven. In this paper, a temperature controller was designed for a conventional electric oven, converting it into a reflow oven. For a matter of compatibility, due to the several kinds of existing ovens in the market, the controller designed was an adaptive controller, that is capable of adjusting automatically to different models of resistance ovens without much user intervention.

Keywords

Adaptive Control, Temperature Control, Resistance Oven, Reflow Soldering.

LISTA DE FIGURAS

Figura 2.1 – Tamanho do revestimento em relação a pastilha de silício.....	14
Figura 2.2 – Comparação entre o tamanho de componentes THT e componentes SMD.....	15
Figura 2.3 – Perfil de temperatura Reflow.....	16
Figura 2.4 – Arquitetura Von Neumann.....	19
Figura 2.5 – Arquitetura Harvard.....	19
Figura 2.6 – Comparação entre modelos de placa Arduino.....	20
Figura 2.7 – Diagrama de blocos de um sistema com escalonamento de ganho.....	28
Figura 2.8 – Diagrama de blocos MRAS.....	29
Figura 2.9 – Diagrama de Blocos STR.....	30
Figura 2.10 – Diagrama de blocos controlador MDPP.....	31
Figura 2.11 – Princípio de funcionamento do termopar.....	35
Figura 3.1 – Esquema da bancada de trabalho.....	37
Figura 3.2 – Esquema elétrico detector de passagem por zero.....	38
Figura 3.3 – Esquema elétrico acionamento do TRIAC.....	39
Figura 3.4 – Forma de onda da tensão na carga.....	39
Figura 3.5 – Esquema elétrico alimentação do circuito de controle.....	40
Figura 3.6 – Termopar tipo K e módulo sensor MAX6675.....	41
Figura 3.7 – Sinal PRBS.....	42
Figura 3.8 – Resposta temporal da temperatura do modelo escolhido (laranja) e da temperatura real medida (azul), para o patamar $T = 120^{\circ}\text{C}$, utilizando dados de validação.....	43
Figura 3.9 – Resposta temporal da temperatura do modelo escolhido (laranja) e da temperatura real medida (azul), para o patamar $T = 175^{\circ}\text{C}$, utilizando dados de validação.....	44
Figura 3.10 – Resposta temporal da temperatura do modelo escolhido (laranja) e da temperatura real medida (azul), para o patamar $T = 175^{\circ}\text{C}$, utilizando dados de validação.....	44
Figura 3.11 - Fluxograma Controle da tensão média.....	45
Figura 4.1 – Esquema da bancada de trabalho.....	46
Figura 4.2 – Face inferior da PCI.....	46
Figura 4.3 – Face superior da PCI.....	46

Figura 4.4 – Resposta da temperatura medida do forno real a uma referência em degrau unitário de 100°C.....	47
Figura 4.5 – Saída do controlador com cancelamento de zeros.....	48
Figura 4.6 – Saída do controlador sem cancelamento de zeros.....	48
Figura 4.7 – Convergência dos parâmetros para o controlador com cancelamento.....	49
Figura 4.8 – Convergência dos parâmetros para o controlador sem cancelamento.....	49
Figura 4.9 – Perfil de temperatura ‘ <i>reflow</i> ’.....	50

LISTA DE TABELAS

Tabela 2.1 – Características do perfil <i>reflow</i>	16
Tabela 2.2 – Temperatura de pico solda com chumbo.....	17
Tabela 2.3 – Temperatura de pico solda sem chumbo.....	17
Tabela 2.4 – Tipos de termopares.....	34

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivo Geral.....	12
1.2	Objetivos Específicos.....	12
1.3	Justificativa.....	12
1.4	Metodologia.....	12
1.5	Estrutura do Trabalho.....	13
2	REVISÃO DE LITERATURA	14
2.1	Tecnologia SMD.....	14
2.1.1	<i>Perfil de soldagem reflow</i>	15
2.2	Microcontroladores.....	18
2.2.1	<i>Arquitetura dos microcontroladores</i>	18
2.2.2	<i>Arduíno</i>	20
2.2.3	<i>Conversor Analógico Digital</i>	21
2.2.4	<i>Portas de entrada e saída(I/O)</i>	21
2.2.5	<i>Interrupções</i>	22
2.3	Controle de fase com tiristor.....	22
2.4	Métodos de identificação de sistemas.....	23
2.4.1	<i>Modelagem matemática</i>	23
2.4.2	<i>Modelos ARX e ARMAX</i>	24
2.4.3	<i>Estimador de mínimos quadrados</i>	24
2.4.4	<i>Estimador recursivo de mínimos quadrados</i>	26
2.4.5	<i>Sinais binários pseudo-aleatórios</i>	27
2.5	Controle adaptativo.....	27
2.5.1	<i>Escalonamento de ganho</i>	28
2.5.2	<i>Modelo de referência sistemas adaptativo (MRAS)</i>	28
2.5.3	<i>Regulador de ajuste próprio (STR)</i>	29
2.5.4	<i>Projeto do controlador por alocação de polos</i>	31
2.5.5	<i>Algoritmo regulador auto ajustável indireto</i>	34
2.6	Medição de temperatura.....	34
2.6.1	<i>Termopares</i>	34
3	METODOLOGIA	37
3.1	Projeto do circuito de controle de potência em corrente alternada.....	38

3.2	Sistema de aquisição de temperatura.....	40
3.3	Identificação dos parâmetros do forno.....	41
3.3.1	<i>Testes dinâmicos e coleta de dados.....</i>	41
3.3.2	<i>Representação matemática e estrutura do modelo.....</i>	42
3.3.3	<i>Estimação de parâmetros e validação do modelo.....</i>	43
3.4	Projeto do regulador auto ajustável indireto.....	45
4	RESULTADOS.....	46
4.1	Placa de circuito impresso para controle de potência CA.....	46
4.2	Teste dos Controladores.....	47
4.3	Resultados da identificação dos parâmetros do forno.....	49
4.4	Perfil <i>reflow</i> projetado.....	50
5	CONCLUSÕES E RECOMENDAÇÕES.....	52
5.1	Conclusões.....	52
5.2	Recomendações para trabalhos futuros.....	52
	REFERÊNCIAS.....	53
	ANEXO 01 (FIRMWARE DO MICROCONTROLADOR).....	55
	ANEXO 02 (PROJETO DA PLACA DE CIRCUITO IMPRESSO).....	70

1. INTRODUÇÃO

Fornos de refluxo são muito usados na produção de placas de circuito impresso que fazem uso de componentes SMD ("Surface-Mount Devices"). Estes fornos trabalham controlando a temperatura em seus interiores de acordo com um perfil de soldagem que passa por diversos patamares, como visto em Goosens (2006). O processo se inicia na fase de pré-aquecimento, tornando o fluxo de solda líquido. O excesso de fluxo escorre e afasta-se dos terminais de soldadura, deixando os grãos de solda no local devido. Logo após, vem a fase de liquefação, onde a temperatura deve subir lentamente, de modo a se manter homogênea por toda a placa, até próximo ao ponto de fusão dos grãos de solda. Alcançada a temperatura, a placa e os componentes devem ser aquecidos até a temperatura máxima. Após esta etapa a solda está efetivamente feita, porém deve-se ainda resfriar o conjunto (fase de arrefecimento), o resfriamento não pode ser demasiadamente rápido, ao ponto de provocar diferenças grandes de temperatura e danificar componentes e soldas e nem demasiadamente lenta, uma vez que certos componentes só podem permanecer acima de uma temperatura crítica por um intervalo de tempo específico.

Um problema prático dos fornos de refluxo é seu preço, que pode torná-lo proibitivo para pequenas aplicações. Porém, um forno de refluxo é basicamente um forno elétrico convencional, acrescido de um controlador de temperaturas. Diversas metodologias de controle convencionais podem ser usadas no projeto e sintonia do controlador, como por exemplo o controle de um forno específico utilizando um controlador Proporcional Integral Derivativo (PID), como feito por Cordeiro (2016). Um problema do projeto convencional é que o controlador é projetado de acordo com o modelo matemático específico do forno, com isso, se o mesmo controlador for aplicado a outro forno, o funcionamento pode ser indesejado.

Neste trabalho deseja-se desenvolver um dispositivo para controle de um forno elétrico convencional de modo que este funcione como um forno de refluxo, porém, deseja-se também que este dispositivo possa funcionar em diversos modelos de fornos elétricos disponíveis no mercado. Para alcançar esta portabilidade, projeta-se um controlador utilizando-se técnicas de controle adaptativo, as quais identificam direta ou indiretamente os parâmetros do forno e ajustam os parâmetros do controlador para o forno em uso.

Diversas técnicas de controle de temperatura adaptativas têm sido usadas tanto para resfriamento quanto para aquecimento, dentre elas, técnicas de controle por redes neurais (LU & TSAI, 2008), técnicas por modelo de referência (WAIT, 2012), dentre outras

(ASTROM & WITTENMARK, 2008). Neste trabalho espera-se conseguir um controlador funcional que atenda aos requisitos de controle estabelecidos para diversos fornos de refluxo através de técnicas de controle adaptativas baseadas no algoritmo dos mínimos quadrados (AGUIRRE, 2007).

1.1 Objetivo Geral

Projetar e desenvolver um controlador de temperatura para fornos, capaz de identificar direta e/ou indiretamente os parâmetros de fornos elétricos convencionais de maneira a se auto-sintonizar, garantindo uma solda de qualidade em placas de circuito impresso que utilizem tecnologias SMD. Além do projeto e desenvolvimento do controlador proposto, tem-se também o objetivo de fazer um comparativo entre o controlador adaptativo proposto e controladores convencionais.

1.2 Objetivos Específicos

- Projetar e implementar o circuito de controle de potência em corrente alternada.
- Identificar o modelo matemático da planta a ser controlada utilizando um estimador de mínimos quadrados.
- Desenvolver um controlador adaptativo aplicado à um sistema de solda SMD.

1.3 Justificativa

Fornos dedicados a soldagem SMD são equipamentos caros, o que restringe o acesso, principalmente a pequenos projetos. O desenvolvimento de um equipamento de baixo custo que possa ser conectado a um forno elétrico convencional, transformando-o em um forno de soldagem, torna esse tipo de equipamento mais acessível, principalmente a pequenos projetistas, uma vez que reduz consideravelmente o custo de investimento em equipamentos.

1.4 Metodologia

- Desenvolver um sistema de controle de potência em corrente alternada.
- Estimar o modelo do forno a partir de dados em batelada.
- Implementar um algoritmo recursivo para identificação do modelo.
- Projetar um controlador adaptativo por alocação de polos.
- Implementar o firmware para o perfil de soldagem *reflow*.

1.5 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos. O capítulo um contextualiza o trabalho, trazendo uma introdução e apresentando os principais objetivos e justificativa. No capítulo dois é apresentado o referencial teórico da proposta, destacando as vantagens da utilização de componentes SMD e explicando o perfil de temperatura esperado assim como o microcontrolador utilizado, seus principais módulos, as técnicas de identificação de sistemas utilizadas, alguns tipos de controladores adaptativos e o sistema de aquisição de temperatura utilizado. O capítulo três apresenta a metodologia utilizada no desenvolvimento desse trabalho, focando no desenvolvimento do circuito de controle de potência em corrente alternada, no método de identificação de sistemas e no projeto do regulador auto ajustável. No capítulo quatro são apresentados os resultados experimentais e o capítulo final apresenta as conclusões e algumas sugestões para trabalhos futuros.

2. REVISÃO DE LITERATURA

2.1 Tecnologia SMD

A sigla SMD é utilizada para referir-se a componentes eletrônicos cujo os terminais de ligação são substituídos por pequenas superfícies de soldagem. É comum encontrar siglas como SMT (*Surface Mount Technology*) ou SMC (*Surface Mount Component*) que também fazem referência a esse tipo de tecnologia.

Na tecnologia de montagem tradicional, conhecida como THT (do inglês “*Through Hole Technology*”) os componentes possuem um revestimento muito maior do que os elementos ativos em seu interior, como é mostrado na figura 2.1, isso é feito para que esses componentes possam ser manuseados mais facilmente por um operador humano. Sendo assim, a maior parte da área ocupada por componentes que utilizam a tecnologia THT corresponde ao invólucro e terminais, ocorrendo assim um desperdício de material e uma ocupação desnecessária de espaço em uma montagem.

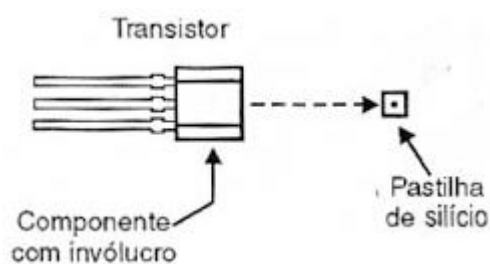


Figura 2.1 – Tamanho do revestimento em relação a pastilha de silício

Fonte: Instituto Newton C. Braga (2019).

Devido à necessidade de se colocar cada vez mais componentes em uma placa de circuito impresso (PCI) e a substituição da montagem humana pela montagem realizada por máquinas, despontou-se uma nova tecnologia de fabricação de componentes muito pequenos, destinados a serem usados por máquinas automáticas, permitindo uma diminuição significativa do tamanho de equipamentos eletrônicos nos dias atuais (INSTITUTO NEWTON C. BRAGA, 2019)

Dentre as vantagens da utilização de dispositivos SMD no lugar de dispositivos convencionais pode-se destacar: a redução do tamanho da PCB, uma vez que componentes SMD são significativamente menores do que os componentes convencionais e também permitem a utilização de ambas as faces da PCI, e aumento da velocidade de resposta do circuito, já que a distância entre os componentes no circuito também é reduzida. Na figura 2.2 é mostrada uma comparação entre o tamanho dos componentes com tecnologia THT e SMD.

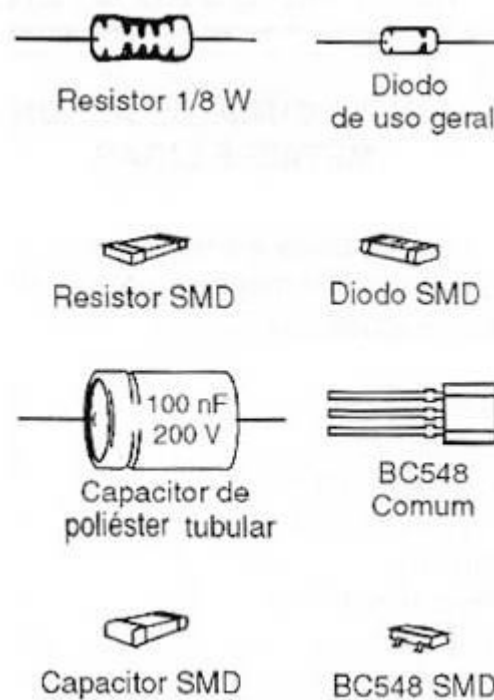


Figura 2.2- Comparação entre o tamanho de componentes THT e componentes SMD.

Fonte: Instituto Newton C. Braga (2019).

2.1.1 Perfil de soldagem *reflow*

O método de soldagem do tipo “*reflow*”, ou refluxo, é o mais comum na produção de PCIs atual pois é um método que permite a soldagem de todos os componentes de uma PCI em apenas um único ciclo termal, isso permite uma maior repetibilidade e eficiência do processo de soldagem. A principal exigência durante a soldagem por refluxo é providenciar valores adequados de temperatura e tempo (PAWLOWSKI, 2018). Na figura 2.3 é mostrado um gráfico com o perfil de temperatura para soldagem do tipo *reflow*, respeitando as restrições mostradas na tabela 2.1.

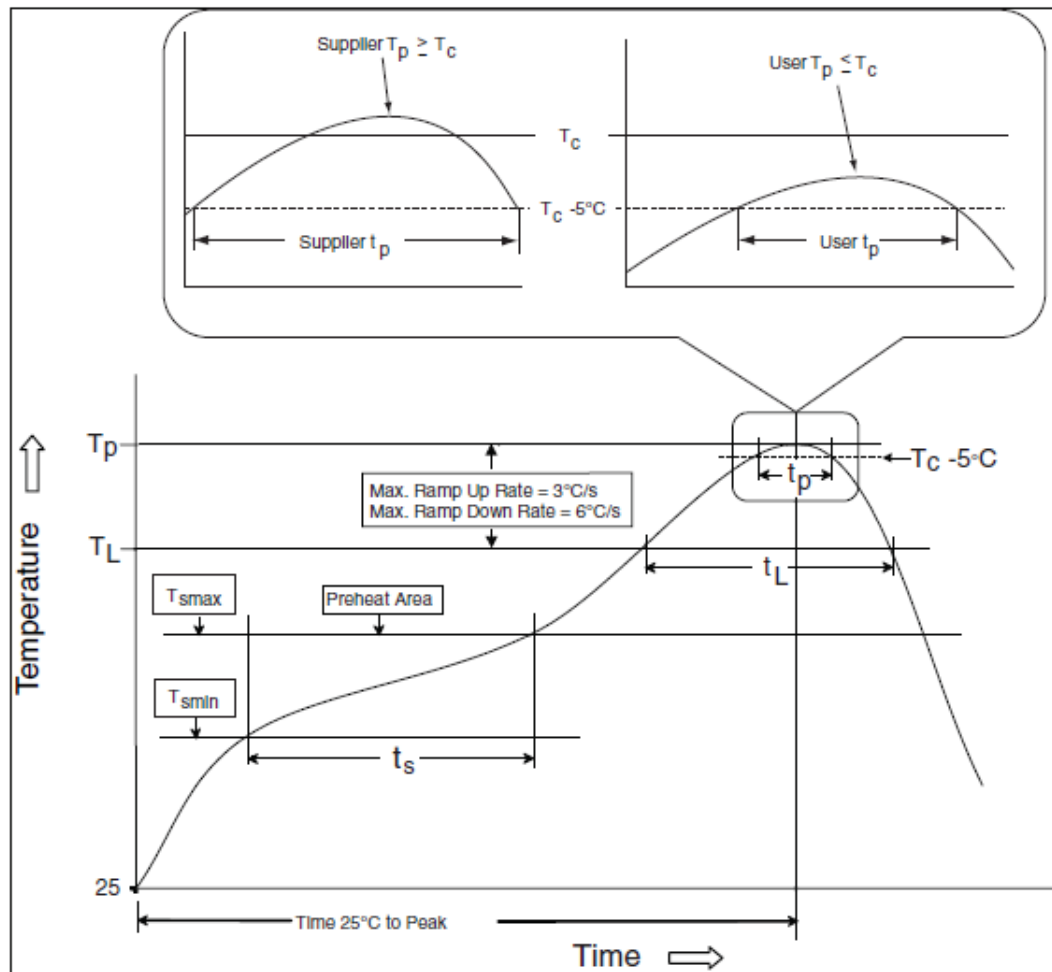


Figura 2.3 – Perfil de temperatura *reflow*.

Fonte: Altera (2011).

Característica do perfil	Solda SnPb	Solda sem chumbo
Taxa média de subida ($T_{S_{max}}$ até T_p)	3 °C/s no máximo	3 °C/s no máximo
Temperatura mínima de pré- aquecimento ($T_{S_{min}}$)	100 °C	150 °C
Temperatura máxima de pré- aquecimento ($T_{S_{max}}$)	150 °C	200 °C
Tempo de pré-aquecimento ($T_{S_{min}}$ até $T_{S_{max}}$)	60-120 segundos	60-120 segundos
Temperatura de liquefação (T_L)	183 °C	217 °C
Tempo com a temperatura acima de T_L	60-150 segundos	60-150 segundos

Temperatura de pico (T_P)	Ver tabela 2.2	Ver tabela 2.3
Tempo com a temperatura na faixa de $(T_P - 5) ^\circ\text{C}$	20 segundos	30 segundos
Taxa média de arrefecimento (T_P até $T_{S_{\max}}$)	6 $^\circ\text{C/s}$ no máximo	6 $^\circ\text{C/s}$ no máximo
Tempo para a temperatura ir de 25 $^\circ\text{C}$ até T_P	6 min no máximo	8 min no máximo

Tabela 2.1 – Características do perfil *reflow*

Fonte: Altera adaptado (2011)

Espessura do pacote	Volume $\text{mm}^3 < 350$	Volume $\text{mm}^3 \geq 350$
< 2.5 mm	235 $^\circ\text{C}$	220 $^\circ\text{C}$
≥ 2.5 mm	220 $^\circ\text{C}$	220 $^\circ\text{C}$

Tabela 2.2 – Temperatura de pico solda com chumbo

Fonte: Altera adaptado (2011)

Espessura do pacote	Volume $\text{mm}^3 < 350$	Volume mm^3 350 - 2000	Volume $\text{mm}^3 >$ 2000
< 1.6 mm	260 $^\circ\text{C}$	260 $^\circ\text{C}$	260 $^\circ\text{C}$
1.6mm - 2.5mm	260 $^\circ\text{C}$	250 $^\circ\text{C}$	245 $^\circ\text{C}$
> 2.5 mm	250 $^\circ\text{C}$	245 $^\circ\text{C}$	245 $^\circ\text{C}$

Tabela 2.3 – Temperatura de pico solda sem chumbo

Fonte: Altera adaptado (2011)

O processo de soldagem por refluxo é iniciado com a fase de pré-aquecimento, na qual temperatura do forno é incrementada até 125 $^\circ\text{C}$, temperatura em que o fluxo contido na solda se torna líquido, fazendo com que o excesso escorra e afaste-se dos terminais de solda, deixando os grãos de solda no local apropriado. Em seguida, na fase de liquefação, a temperatura sobe de forma um pouco mais lenta até os 175 $^\circ\text{C}$, próximo ao ponto de fusão dos grãos de solda. O motivo para incrementar a temperatura de forma lenta nesta etapa é que a placa e os componentes necessitam de tempo para ficarem na mesma temperatura. Passada a etapa de liquefação o forno deve aquecer a placa e os componentes até a temperatura máxima, geralmente por volta de 220-240 $^\circ\text{C}$. Nesta etapa os grãos de solda derretem e ligam-se ao metal em volta (GOOSENS, 2006).

Segundo Goosens (2006) depois de atingida a temperatura máxima é necessário resfriar todo o conjunto, esta etapa é chamada de arrefecimento. Este resfriamento não deve ser

imediatamente, de forma a evitar diferenças muito grande de temperatura entre os componentes e a placa, que pode ocasionar a deformação dos componentes ou até mesmo quebrá-los. No entanto este resfriamento também não pode ser muito lento, principalmente no início, porque alguns componentes podem somente permanecer acima de uma determinada temperatura crítica por um dado intervalo de tempo.

De acordo com Goosens (2006) as pastas de solda podem ser divididas em dois grupos: As soldas baseadas em chumbo e as que não contém chumbo. Em grande parte das soldas que contém chumbo a liga é composta pelo chumbo em conjunto com estanho (SnPb), esta liga possui o ponto de fusão em torno de 183 °C. Já as pastas de solda sem chumbo normalmente são formadas por uma liga de estanho, prata e cobre (SnAgCu) e devido ao fato de não possuir chumbo na sua composição, esta liga geralmente possui um ponto de fusão por volta dos 217 °C.

2.2 Microcontroladores

Os microcontroladores são pequenos dispositivos eletrônicos providos de uma “inteligência” programável, que possuem processador, pinos de entrada/saída e memória. O microcontrolador é programável pois toda a sua lógica de operação é estruturada na forma de um programa, que é gravado dentro do componente e permite controlar o estado das suas saídas, de acordo com o que é desejado (SOUZA & SOUSA, 2012).

Normalmente, um microcontrolador é caracterizado por integrar no mesmo encapsulamento um microprocessador (com o propósito de interpretar as instruções de programa e processar dados), memória de programa (cuja finalidade é armazenar as instruções de programa), memória de dados (responsável por armazenar os valores das variáveis usadas durante a execução do programa), um conjunto de pinos de entrada/saída (responsável por realizar a comunicação do microcontrolador com o meio externo) e diversos periféricos como temporizadores, controladores de interrupção, WatchDog Timers, comunicação serial, geradores de modulação por largura de pulso (PWM), conversores analógico/digital, etc. (MARTINS 2005).

2.2.1 Arquitetura dos microcontroladores

Todos os microcontroladores atualmente são divididos em dois grupos: Os que utilizam a arquitetura Harvard e os que utilizam a arquitetura von-Neumann, sendo a principal

diferença entre elas a forma de trocar informações entre a Unidade Central de Processamento (CPU do inglês “*Central Process Unit*”) e a memória (SENA, s.d.).

Os microcontroladores que utilizam a arquitetura von-Neumann possuem somente uma zona de memória disponível e um barramento de 8 bits. Devido ao fato de todos os dados serem trocados utilizando estas 8 linhas, este barramento é sobrecarregado e a comunicação acaba se tornando extremamente lenta e ineficiente. Dessa forma o CPU pode ler uma instrução ou ler/escrever dados na memória, não sendo possível realizar ambas as operações de forma simultânea, pois as instruções compartilham o barramento com os dados (SENA, s.d.). Na figura 2.4 é mostrado o esquema de barramentos da arquitetura Von Neumann.

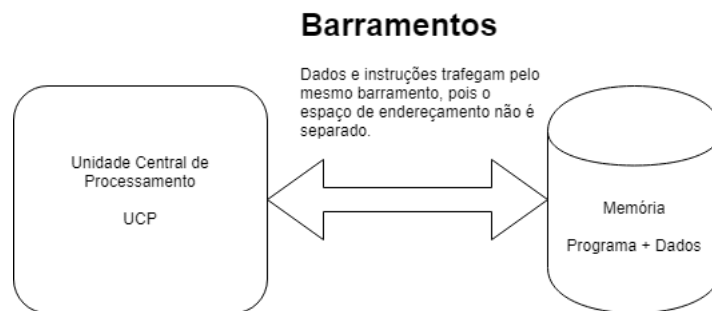


Figura 2.4 – Arquitetura Von Neumann

Fonte: Embarcados (2019)

Os microcontroladores que utilizam a arquitetura Harvard possuem dois barramentos internos, como mostrado na figura 2.5, um barramento de dados de 8 bits e um barramento de instruções, que pode ser de 12, 14 ou 16 bits. Essa arquitetura possibilita que, enquanto uma instrução está sendo executada, outra seja buscada na memória, aumentando assim a velocidade de processamento desejado (SOUZA & SOUSA, 2012).

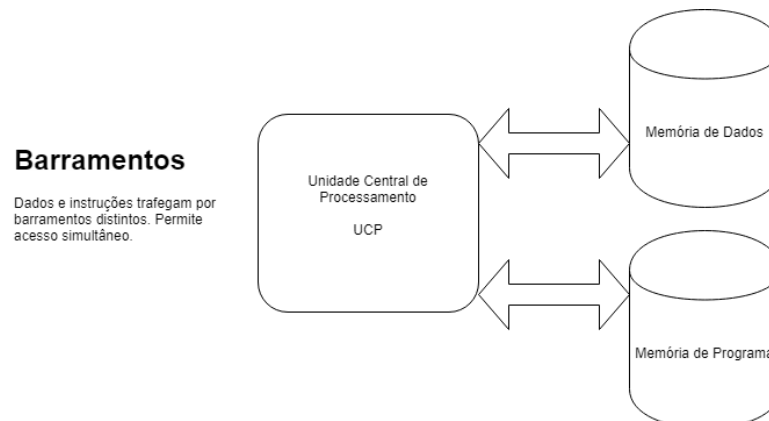


Figura 2.5 – Arquitetura Harvard

Fonte: Embarcados (2019)

2.2.2 Arduino

Arduino é uma plataforma aberta de prototipagem eletrônica, criada em 2005 por um grupo de cinco pesquisadores com o propósito de desenvolver um dispositivo barato, funcional e de fácil programação, sendo assim mais acessível a estudantes e projetistas amadores. Também foi adotado o conceito de hardware livre, onde qualquer um pode montar, modificar, melhorar e personalizar o Arduino, baseando-se no mesmo hardware básico.

Dessa forma foi criada uma placa composta por um microcontrolador Atmel que pode ser facilmente conectada a um computador e programada via IDE (do inglês “*Integrated Development Environment*”) usando uma linguagem baseada em C/C++, dispensando a necessidade de outros equipamentos além de um cabo USB (THOMSEN, 2014).

Hoje em dia é possível encontrar uma variedade de modelos de placas Arduino e a escolha da placa a ser utilizada depende do projeto a ser desenvolvido, levando em consideração o número de entradas/saídas, memória, etc. Na figura 2.4 é mostrada uma comparação entre modelos de placa Arduino com relação aos seus periféricos.




	Arduino Uno	Arduino Mega2560	Arduino Leonardo	Arduino Due	Arduino ADK	Arduino Nano	Arduino Pro Mini	Arduino Espora
								
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega168	ATmega32u4
Portas digitais	14	54	20	54	54	14	14	-
Portas PWM	6	15	7	12	15	6	6	-
Portas analógicas	6	16	12	12	16	8	8	-
Memória	32 K (0,5 K usado pelo bootloader)	256 K (8 K usados pelo bootloader)	32 K (4 K usados pelo bootloader)	512 K disponível para aplicações	256 K (8 K usados pelo bootloader)	16 K (ATmega168) ou 32K (ATmega328), 2 K usados pelo bootloader	16 K (2k usados pelo bootloader)	32 K (4 K usados pelo bootloader)
Clock	16 Mhz	16 Mhz	16 Mhz	84 Mhz	16 Mhz	16 Mhz	8 Mhz (modelo 3.3v) ou 16 Mhz (modelo 5v)	16 Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial / Módulo USB externo	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Tensão de operação	5v	5v	5v	3.3v	5v	5v	3.3v ou 5v, dependendo do modelo	5v
Corrente máxima portas E/S	40 mA	40 mA	40 mA	130 mA	40 mA	40 mA	40 mA	-
Alimentação	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	3.35 - 12 V (modelo 3.3v), ou 5 - 12 V (modelo 5v)	5v

Figura 2.6 – Comparação entre modelos de placa Arduino.

Fonte: Luthortronics (2019).

Nesse trabalho, por uma questão de disponibilidade, foi utilizado o modelo Arduino Mega 2560, que possui um microcontrolador ATmega2560, 54 portas digitais e 16 portas analógicas. Esse modelo é ideal para projetos mais elaborados, onde é necessário um número grande de entradas e saídas.

2.2.3 Conversor analógico digital

No mundo real grande parte das grandezas não são de natureza elétrica, de forma que, para trazer esse mundo para o microcontrolador é necessário transformar essas grandezas em sinais elétricos. Os equipamentos que fazem essa transformação são conhecidos como sensores ou transdutores. Esses sensores transformam grandezas como pressão, temperatura, luminosidade, vazão, etc. em sinais elétricos, podendo esses sinais serem lineares e proporcionais à amplitude das grandezas aferidas, ou então não lineares com curvas conhecidas e que podem ser compensadas de alguma forma posteriormente.

Uma vez que essas grandezas são transformadas em sinais elétricos, esses sinais, que em geral são analógicos e contínuos no tempo, precisam ser transformados em sinais digitais que poderão ser tratados e processados pelo programa dentro do microcontrolador. Essa segunda transformação é feita por um periférico conhecido como Conversor Analógico/Digital.

Um conversor A/D transforma um sinal analógico e contínuo no tempo em um sinal digital, discreto no tempo, quantizado dentro de um intervalo finito de valores inteiros. Esse intervalo é determinado pela resolução característica do conversor, podendo ser de 8, 10, 12 bits ou superior (PUHLMANN, 2015). O Arduino Mega 2560, por exemplo, possui um conversor de 10 bits, dessa forma o sinal da entrada analógica é transformado em amostras com valores de 0 a 1023. No trabalho o conversor A/D foi utilizado para poder transformar o sinal do termopar de mV para um valor de temperatura em °C.

2.2.4 Portas de entrada e saída (I/O)

As portas de entrada e saída de um microcontrolador são os periféricos responsáveis pela “comunicação” com o meio externo, sendo responsáveis por receber os sinais de sensores que serão tratados pelo processador (entrada) e por enviar sinais de comando aos atuadores (saída). As portas de I/O podem ser classificadas em digitais ou analógicas. No trabalho as portas de entrada e saída são usadas para acionar os tiristores utilizados para controlar a tensão do forno e também para detectar a passagem por zero da rede.

2.2.5 Interrupções

As interrupções, assim como o nome sugere, servem para interromper o programa imediatamente, de forma a possibilitar a tomada de atitudes instantâneas. As interrupções são ações tratadas diretamente pelo hardware, o que faz com que sejam muito rápidas e disponíveis em qualquer ponto do sistema. Dessa forma, quando ocorre uma interrupção o programa é interrompido e uma função específica, denominada Rotina de Serviço de Interrupções (ISR do inglês “*Interruption Service Routine*”), é executada e em seguida o programa retorna sua execução a partir do ponto onde foi interrompido.

As ISRs são funções que possuem algumas limitações que as diferem das outras funções, por exemplo, uma ISR não recebe argumentos e não devem retornar nada. Normalmente uma ISR deve ser o mais curta e rápida possível, de forma que o tempo gasto para tratá-la não seja demasiadamente longo. Somente uma ISR pode ser executada por vez, sendo que, na eventual ocorrência simultânea de mais de uma interrupção, as ISRs correspondentes a cada interrupção serão executadas em ordem de prioridade. Como as ISRs não podem receber ou retornar argumentos, variáveis globais são usadas para transmitir dados entre uma ISR e o programa principal (SOUZA & SOUSA, 2012). No trabalho uma rotina de interrupção é utilizada para que o TRIAC seja acionado no momento em que há uma passagem por zero na rede.

2.3 Controle de fase com tiristor

O controle de fase com tiristor é uma técnica utilizada para controlar a potência em corrente alternada (CA) através da variação do valor eficaz de tensão aplicada na carga. Essa variação é feita através de um chaveamento, para isso são utilizadas chaves eletrônicas, podendo ser realizada por meio de dois métodos distintos: O controle por ângulo de disparo e o controle por ciclos inteiros. Normalmente as chaves eletrônicas utilizadas são dois tiristores em ligação antiparalela para cargas com uma potência maior, na ordem de kW, ou um TRIAC para cargas menores (KOMATSU, 2017).

Segundo Komatsu (2017), no controle por ângulo de disparo o ângulo de disparo dos tiristores é variado entre $0 \leq \alpha \leq \pi$ para cada semiciclo, assim o valor de tensão eficaz aplicado na carga decresce à medida que o valor de α aumenta. O valor eficaz da tensão na carga pode ser calculado como:

$$V_c = \left(\frac{1}{\pi} \int_{\alpha}^{\pi} (\sqrt{2} V_r)^2 \sin^2 \theta d\theta \right)^{\frac{1}{2}} = V_r \left(\frac{2(\pi - \alpha) + \sin 2\alpha}{2\pi} \right)^{\frac{1}{2}} \quad (2.1)$$

De acordo com Komatsu (2017) no controle por ciclos inteiros os tiristores são sempre disparados com $\alpha = 0^\circ$. A cada N ciclos completos da tensão da rede são aplicados K ciclos completos na carga, onde $K \leq N$. Assim, o valor eficaz da tensão aplicada na carga varia proporcionalmente à relação de N e K , podendo ser calculada da seguinte maneira:

$$V_c = V_r \sqrt{\frac{K}{N}} \quad (2.2)$$

2.4 Métodos de identificação de sistemas

2.4.1 Modelagem matemática

Modelagem matemática é a área do conhecimento que estuda formas de desenvolver e implementar modelos matemáticos de sistemas reais. Um modelo matemático de um sistema é uma aproximação matemática que representa algumas das características observadas nesse sistema. Existem diversas maneiras de classificar técnicas de modelagem, uma delas reúne os métodos em três categorias: modelagem caixa branca, modelagem caixa preta e modelagem caixa cinza.

Na modelagem do tipo caixa branca necessita-se conhecer bem o sistema estudado, assim como as leis da física que o descrevem. No entanto, devido ao conhecimento e tempo necessários para modelar um sistema partindo do equacionamento dos fenômenos envolvidos, nem sempre é viável esse tipo de modelagem (AGUIRRE, 2007).

Segundo Aguirre (2007), a área de conhecimento que estuda técnicas alternativas de modelagem matemática é a identificação de sistemas. A principal característica dessas técnicas é que é necessário pouco ou nenhum conhecimento prévio do sistema, esses métodos são denominados modelagem caixa preta ou modelagem empírica. A motivação para o estudo de técnicas de identificação de sistemas vem com o fato de que constantemente não são conhecidas as equações envolvidas no funcionamento de um determinado sistema, ou então são conhecidas, mas devido a limitações de tempo e de recursos torna-se impraticável levantar essas equações e estimar seus respectivos parâmetros.

2.4.2 Modelos ARX E ARMAX

Segundo Aguirre (2007), existem algumas representações matemáticas que são particularmente adequadas para a identificação de sistemas, utilizando-se de algoritmos conhecidos para estimação de parâmetros. Considerando o modelo geral:

$$A(q)y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}v(k) \quad (2.3)$$

Onde $y(k)$ e $u(k)$ representam respectivamente a saída e a entrada do sistema e $v(k)$ é um ruído branco com média nula. $A(q)$, $B(q)$, $C(q)$, $D(q)$ e $F(q)$ são polinômios definidos da seguinte maneira:

$$A(q) = 1 - a_1 q^{-1} - \dots - a_{ny} q^{-ny} \quad (2.4)$$

$$B(q) = b_1 q^{-1} + \dots + b_{nu} q^{-nu} \quad (2.5)$$

$$C(q) = 1 + c_1 q^{-1} - \dots - c_{nv} q^{-nv} \quad (2.6)$$

$$D(q) = 1 + d_1 q^{-1} - \dots - d_{nd} q^{-nd} \quad (2.7)$$

$$F(q) = 1 + f_1 q^{-1} - \dots - f_{nf} q^{-nf} \quad (2.8)$$

Sendo q^{-1} o operador de atraso, ou seja, $y(k)q^{-1} = y(k-1)$.

O modelo auto-regressivo com entradas exógenas (ARX do inglês *autoregressive with exogenous inputs*) é obtido a partir do modelo (2.3) fazendo-se $C(q) = D(q) = F(q) = 1$.

$$A(q)y(k) = B(q)u(k) + v(k) \quad (2.9)$$

$A(q)$ e $B(q)$ dependem da estrutura do modelo da planta a ser identificada.

O modelo auto-regressivo com média móvel e entradas exógenas (ARMAX) pode ser obtido a partir do modelo geral (2.1) onde $D(q) = F(q) = 1$.

$$A(q)y(k) = B(q)u(k) + C(q)v(k) \quad (2.10)$$

Nesse caso obtém-se o modelo do erro na equação $C(q)$ como um processo de média móvel.

2.4.3 Estimador de Mínimos Quadrados

Um dos métodos mais conhecidos e utilizados em diversas áreas da ciência e da tecnologia é o estimador de mínimos quadrados, isso porque esse estimador minimiza uma função de custo, que é definida como o somatório do quadrado dos erros entre as medições e as predições de um passo à frente, sendo assim esse estimador possui boas propriedades de robustez ao ruído presente nos dados (AGUIRRE, 2007). Considerando o modelo (2.9) com $ny = nu = 2$, tem-se

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + b_1 u(k-1) + b_2 u(k-2) + v(k) \quad (2.11)$$

Transformando (2.11) em uma equação matricial para N amostras de dados temos:

$$Y = X\hat{\theta} + \xi \quad (2.12)$$

Onde,

$$Y = \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-N) \end{bmatrix} \quad X = \begin{bmatrix} y(k-1) & y(k-2) & u(k-1) & u(k-2) \\ y(k-2) & y(k-3) & u(k-2) & u(k-3) \\ \vdots & \vdots & \vdots & \vdots \\ y(k-N-1) & y(k-N-2) & u(k-N-1) & u(k-N-2) \end{bmatrix}$$

$$\hat{\theta} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} \quad \xi = \begin{bmatrix} v(k) \\ v(k-1) \\ \vdots \\ v(k-N) \end{bmatrix} \quad (2.13)$$

Sendo Y o vetor das saídas observadas, X a matriz de regressores, $\hat{\theta}$ o vetor de parâmetros estimados e ξ o vetor de erros cometidos ao tentar explicar y como $X\hat{\theta}$, ou seja, $v(k) = y(k) - \hat{y}(k)$.

A ideia em torno do método de mínimos quadrados consiste em estimar os parâmetros θ de forma que o erro de estimação seja mínimo, define-se então a função de custo J_{MQ} como o somatório do quadrado dos erros:

$$J_{MQ} = \sum_{i=1}^N \xi(i)^2 = \xi^T \xi = \|\xi\|^2 \quad (2.14)$$

Substituindo (2.10) em (2.12), tem-se

$$J_{MQ} = (Y - X\hat{\theta})^T (Y - X\hat{\theta})$$

$$= Y^T Y - Y^T X\hat{\theta} - \hat{\theta}^T X^T Y + \hat{\theta}^T X^T X \hat{\theta} \quad (2.15)$$

Para minimizar a função de custo, iguala-se a zero a derivada parcial de J_{MQ} em relação à $\hat{\theta}$, ou seja, $(\partial J_{MQ} / \partial \hat{\theta}) = 0$.

$$\frac{\partial J_{MQ}}{\partial \hat{\theta}} = -(Y^T X)^T - X^T Y + (X^T X + X^T X) \hat{\theta}$$

$$0 = -XY - X^T Y + 2X^T X \hat{\theta}$$

$$\hat{\theta} = [X^T X]^{-1} X^T Y \quad (2.16)$$

Dessa forma é possível estimar os valores de θ , minimizando o somatório do quadrado dos erros e utilizando somente os dados de entrada e saída do sistema.

2.4.4 Estimador Recursivo Estendido de Mínimos Quadrados

Segundo Aguirre (2007) em casos onde o ruído é autocorrelacionado (colorido) e o modelo inclui regressores da saída, o estimador de mínimos quadrados tende a ser polarizado, ou seja, alguns parâmetros tendem para valores diferentes dos parâmetros reais do sistema.

O fato do ruído ser autocorrelacionado significa que existe uma correlação entre os valores do vetor de ruído, portanto existe um modelo que representa esse ruído. O estimador estendido de mínimos quadrados consegue contornar o problema de polarização estendendo a matriz de regressores X e o vetor de parâmetros, incluindo regressores e parâmetros do resíduo. Obtém-se então uma nova matriz de regressores X^* e um novo vetor de parâmetros $\hat{\theta}^*$.

$$X^* = \begin{bmatrix} & e(k) & e(k-1) \\ & e(k-1) & e(k-2) \\ & & \vdots \\ X & & \vdots \\ & & \vdots \\ & e(k-N) & e(k-N-1) \end{bmatrix} \quad \hat{\theta}^* = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \\ c_1 \\ c_2 \end{bmatrix} \quad (2.17)$$

Dessa forma, tem-se

$$\hat{\theta}^* = [X^{*T} X^*]^{-1} X^{*T} Y \quad (2.18)$$

Segundo Astrom (2008) em controladores adaptativos, devido ao fato das observações serem feitas sequencialmente em tempo real, é desejável que a estimação de parâmetros seja feita de forma recursiva, de forma a economizar tempo de computação.

Segundo Aguirre (2007) o estimador estendido de mínimos quadrados pode ser calculado de forma recursiva utilizando o seguinte algoritmo:

$$\begin{aligned} K_k &= P_{k-1} X_k [X_k^T P_{k-1} X_k + 1]^{-1} \\ \hat{\theta}_k &= \hat{\theta}_{k-1} + K_k [y(k) - X_k^T \hat{\theta}_{k-1}] \\ P_k &= P_{k-1} - K_k X_k^T P_{k-1} \\ \xi &= y(k) - X_k^T \hat{\theta}_k \end{aligned} \quad (2.19)$$

Onde K_k é uma matriz de ganhos, P_k é a matriz de covariância. É importante observar que o vetor de regressores X_k contém valores somente até o instante $k-1$. Vale ressaltar também que durante a primeira iteração o vetor de regressores não contém resíduos e a matriz de covariância P_0 recebe valores suficientemente elevados para que a condição inicial de $\hat{\theta}_0$ não afete a qualidade da predição de um passo à frente.

2.4.5 Sinais binários pseudo-aleatórios

Quando se busca um modelo matemático que relacione dinamicamente duas variáveis de um sistema assume-se que existe uma correlação significativa entre essas variáveis de forma a justificar o modelo (AGUIRRE, 2007).

Segundo Lennox (2001 apud AGUIRRE, 2007) a escolha das variáveis de entrada e a determinação correta de atrasos de transporte são extremamente importantes na modelagem de processos. A função de correlação cruzada é apresentada como uma ferramenta capaz de determinar a existência de correlação significativa entre duas variáveis candidatas a compor um modelo.

Segundo Aguirre (2007) para se obter a resposta ao impulso de um sistema $h(k)$ a partir da função de autocorrelação de entrada $r_{uu}(k)$ e da função de correlação cruzada entre a entrada e a saída, a entrada deve ser tal que sua função de autocorrelação seja diferente de zero somente para o atraso $\tau = 0$. Os sinais que satisfazem essa condição são sinais com espectro de frequência branco ou quase branco, ou seja, sinais aleatórios ou pseudo-aleatórios.

Os sinais binários pseudo-aleatórios (PRBS) são bastante utilizados na identificação de sistemas lineares pois são relativamente fáceis de serem gerados. Esses sinais possuem somente dois valores possíveis, $+V$ e $-V$, e qualquer mudança de valor pode ocorrer somente em instantes bem definidos $t = 0, T_b, 2T_b, \dots$. Devido ao fato de que as mudanças entre valores é determinada de forma determinística o mesmo sinal pode ser gerado repetidamente.

O tipo mais comum de sinal PRBS é a sequência de comprimento máximo, ou sinais de sequência m . Para gerar sinais de sequência m é necessário executar a operação OU-Exclusivo entre alguns bits específicos do registro de deslocamento, ressaltando que são poucas as combinações que resultam em sinais de sequência m . Estes sinais têm períodos de amostragem $T = NT_b$, onde $N = 2^{n-1}$ com n sendo o número de bits do registro de deslocamento. Para se gerar sinais PRBS de sequência m que possam ser usados com êxito é necessário a escolha adequada dos parâmetros V, n e T_b (AGUIRRE, 2007).

2.5 Controle adaptativo

Segundo Astrom (2008) “adaptar-se” significa mudar de comportamento para se adequar às novas condições. Sendo assim é possível dizer que um controlador adaptativo é aquele

capaz de modificar seu comportamento em resposta a variações na dinâmica do sistema, possuindo tanto parâmetros ajustáveis assim como mecanismos para ajustá-los.

A abordagem mais comum no desenvolvimento de sistemas de controle consiste em encontrar um modelo linear para o processo, em uma determinada condição de operação, e projetar um controlador com parâmetros constantes. Embora seja uma solução bastante eficaz, existem alguns mecanismos que fazem emergir variações na dinâmica do processo, como atuadores com características não-lineares, etc. (ASTROM & WITTENMARK, 2008). A seguir serão descritos três tipos de sistemas adaptativos: Escalonamento de ganho, Sistema adaptativo modelo-referência e regulador auto sintonizável.

2.5.1 Escalonamento de Ganho

Em alguns casos é possível encontrar variáveis mensuráveis que se correlacionam bem com as mudanças na dinâmica do processo e essas variáveis podem ser usadas para mudar os parâmetros do controlador. Essa abordagem é conhecida como escalonamento de ganho pois era originalmente usada para medir o ganho e então mudar o controlador para compensar as mudanças no ganho do processo (ASTROM & WITTENMARK, 2008). Na figura 2.7 é mostrado o diagrama de blocos de um sistema com escalonador de ganho.

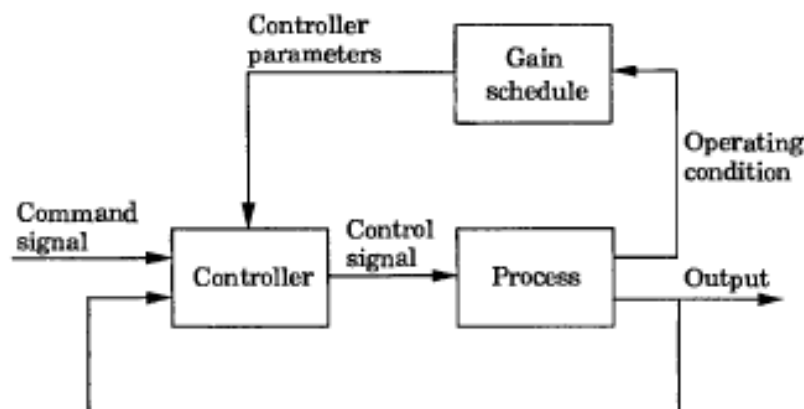


Figura 2.7 – Diagrama de blocos de um sistema com escalonamento de ganho

Fonte: Astrom(2008)

2.5.2 Modelo de Referência Sistemas Adaptativo (MRAS)

Segundo Astrom (2008) o Modelo de Referência Sistemas Adaptativo (MRAS do inglês “*model-reference adaptive system*”) foi originalmente proposto para solucionar

problemas nos quais as especificações de performance são dadas em termos de modelo de referência. Esse modelo mostra como a saída do processo deveria responder a um sinal de comando. Este controlador consiste em dois ‘*loops*’, como é mostrado na figura 2.8.

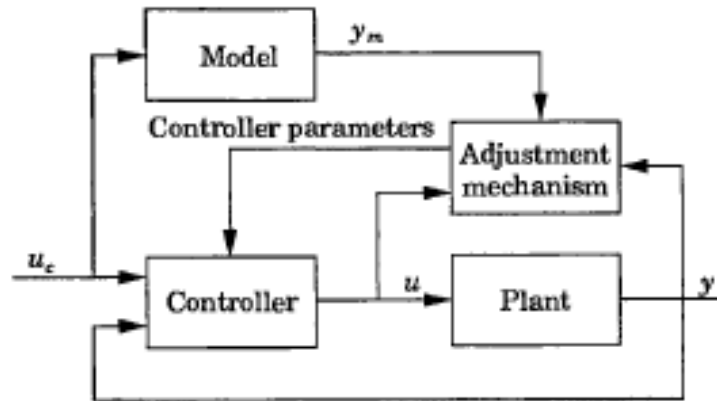


Figura 2.8 – Diagrama de blocos MRAS

Fonte: Astrom (2008)

O *loop* interno é uma realimentação simples composta pela planta do processo e pelo controlador. Já o *loop* externo ajusta os parâmetros do controlador de forma que o erro, diferença entre a saída do processo (y) e a saída do modelo (y_m), seja pequeno. O principal problema com o MRAS é determinar um mecanismo de ajuste de forma que um sistema estável seja obtido (ASTROM & WITTENMARK, 2008).

2.5.3 Regulador de Ajuste Próprio (STR)

O Regulador de Ajuste Próprio (STR do inglês *Self-tuning regulator*) é o sistema adaptativo obtido quando as estimações dos parâmetros do processo são atualizadas e os parâmetros do controlador são obtidos a partir da solução de um problema de projeto utilizando os parâmetros estimados. O controlador adaptativo pode ser arranjado como um conjunto de dois *loops*, como é mostrado na figura 2.9, de forma análoga ao MRAS. Os parâmetros do controlador são ajustados no *loop* externo, que é composto por um estimador recursivo dos parâmetros do processo e um estimador para os parâmetros do controlador (ASTROM & WITTENMARK, 2008).

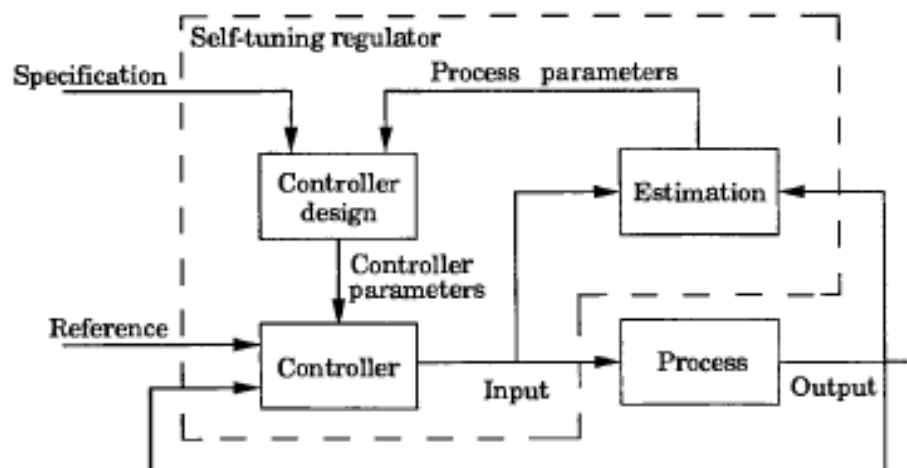


Figura 2.9 – Diagrama de Blocos STR

Fonte: Astrom (2008)

Segundo Astrom (2008) é importante ressaltar que este sistema deve ser visto como uma forma de automatização da modelagem do processo e do projeto do controlador, onde os parâmetros de ambos são atualizados a cada período de amostragem, fazendo com que o controlador sintonize automaticamente seus parâmetros de forma a obter as propriedades desejadas em um sistema de malha-fechada.

A estimação de parâmetros pode ser feita tanto de forma contínua quanto em batelada. Se tratando de implementações digitais, que no caso são as mais usuais, podem-se usar períodos de amostragem diferentes para o estimador e para o controlador, sendo possível inclusive utilizar esquemas híbridos, onde o controle é feito de forma contínua e a estimação de parâmetros é feita de forma discreta. Embora diversos métodos de estimação possuam incertezas nos valores dos parâmetros estimados, essas incertezas normalmente não são consideradas quando se projeta o controlador, os parâmetros estimados são tratados como se fossem os valores verdadeiros da planta. Essa assunção é denominada princípio da equivalência.

O controlador apresentado na figura 2.9 possui uma estrutura bastante rica, onde somente algumas possibilidades foram analisadas. A escolha da estrutura do modelo e da sua parametrização é um problema importante quando se implementa um regulador de ajuste próprio. Uma abordagem direta consiste em estimar os parâmetros da função de transferência do processo, isso nos dá um algoritmo adaptativo indireto, ou seja, os parâmetros do controlador não são atualizados diretamente e sim indiretamente, através da estimação do modelo do processo.

Outro tipo de abordagem consiste em reparametrizar o modelo de forma que os parâmetros do controlador possam ser estimados de forma direta, fornecendo um algoritmo adaptativo direto. A ideia básica nos dois tipos de algoritmos é identificar alguns parâmetros que são relacionados com o processo e com as especificações do sistema em malha fechada (ASTROM & WITTENMARK, 2008).

2.5.4 Projeto do controlador por alocação de polos

Essa técnica de controle consiste essencialmente em projetar um controlador que forneça determinados polos em malha fechada, fazendo com que o regime transiente do sistema se comporte da maneira desejada, com um sobressalto e um tempo de acomodação especificados de acordo com as necessidades do sistema a ser controlado. Este método apesar de ser bastante simples, quando propriamente aplicado, é capaz de resultar em controladores bastante úteis além de fornecer uma boa compreensão da ideia do controle adaptativo.

Para o projeto desse controlador assume-se que o modelo do processo pode ser descrito por (2.9), portanto o diagrama de blocos para o controlador adaptativo por alocação de polos se dá como mostrado na figura 2.10.

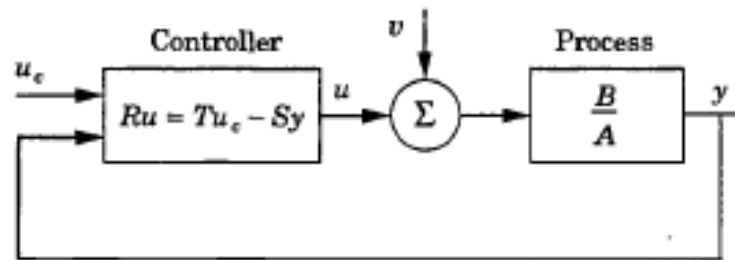


Figura 2.10 – Diagrama de blocos controlador MDP

Fonte: Astrom (2008)

Sendo u_c o valor desejado da saída (Set Point) e v um ruído branco de média nula. É importante ressaltar que os polinômios A e B devem ser relativamente primos, ou seja, não podem ter termos em comum (ASTROM & WITTENMARK, 2008). O controlador linear geral é descrito por:

$$Ru(k) = Tu_c(k) - Sy(k) \quad (2.20)$$

Onde R , S e T são polinômios em função do operador de atraso de fase (q^{-1}). Substituindo $u(k)$ de (2.9) em (2.20) temos:

$$\begin{aligned} y(k) &= \frac{BT}{AR+BS} u_c(k) + \frac{R}{AR+BS} v(k) \\ u(k) &= \frac{AT}{AR+BS} u_c(k) - \frac{S}{AR+BS} v(k) \end{aligned} \quad (2.21)$$

A equação característica do sistema em malha fechada é dada por:

$$AR + BS = A_c \quad (2.22)$$

Sendo assim, o ponto chave desse método consiste em especificar os polos desejados em A_c . A equação (2.22) é denominada '*Equação Diofantina*', e sempre possui uma solução se os polinômios A e B não possuem parâmetros em comum. No caso em que os polinômios possuem parâmetros aproximados a solução é pobremente condicionada. Os polinômios R e S do controlador podem ser obtidos através da solução das equações lineares obtidas através de (2.22).

Uma vez que a partir da equação diofantina é possível determinar somente os polinômios R e S , outras condições devem ser introduzidas para se determinar T no controlador (2.20). Para isso precisa-se que a resposta da saída do sistema a um sinal de controle u_c seja descrita pela seguinte dinâmica:

$$A_m y_m(k) = B_m u_c(k) \quad (2.23)$$

A partir de (2.21) a seguinte condição deve ser mantida:

$$\frac{BT}{AR+BS} = \frac{BT}{A_c} = \frac{B_m}{A_m} \quad (2.24)$$

Essa condição para adaptação de parâmetros nos diz que a resposta a um sinal de comando do sistema em malha fechada pode ser especificada pelo modelo (2.23). A condição para que a adaptação de parâmetros possa ser alcançada depende do modelo, do sistema e do sinal de comando. Caso seja possível fazer o erro ser igual a zero para qualquer sinal de comando, então diz-se que a adaptação de modelo perfeita é alcançada.

Da equação (2.24) implica-se que existem o cancelamento de fatores de BT e A_c . Fatorando o polinômio B_m temos:

$$B = B^+ B^- \quad (2.25)$$

Onde B^+ é um polinômio mônico, em que os zeros são estáveis e tão bem amortecidos que eles podem ser cancelados pelo controlador, e B^- corresponde aos fatores instáveis e pobremente amortecidos que não podem ser cancelados (ASTROM & WITTENMARK, 2008). Determina-se então que B^- deva ser um fator de B_m , sendo assim temos que:

$$B_m = B^- B'_m \quad (2.26)$$

Como B^+ é cancelado, ele deve ser um fator de A_c . Além disso, determina-se a partir de (2.24) que A_m também deva ser um fator de A_c . Sendo assim, a equação característica em malha fechada tem a seguinte forma:

$$A_c = A_o A_m B^+ \quad (2.27)$$

Uma vez que B^+ é um fator de B e de A_c , segue-se a partir de (2.22) que ele também divide R , dessa forma temos que:

$$R = R' B^+ \quad (2.28)$$

E a equação diofantina (2.22) é reduzida a:

$$AR' + B^- S = A_o A_m = A'_c \quad (2.29)$$

Substituindo as equações (2.25), (2.26) e (2.27) na equação (2.24) obtém-se:

$$T = A_o B'_m \quad (2.30)$$

Para se obter um controlador causal no tempo discreto deve-se impor as seguintes condições:

$$\begin{aligned} \deg S &\leq \deg R \\ \deg T &\leq \deg R \end{aligned} \quad (2.31)$$

A equação diofantina (2.22) possui várias soluções porque se R^0 e S^0 são soluções da equação, então temos que:

$$\begin{aligned} R &= R^0 + QB \\ S &= S^0 - QA \end{aligned} \quad (2.32)$$

Onde Q é um polinômio arbitrário. Uma vez que existem diversas soluções para a equação diofantina (2.22), devemos escolher a solução que nos fornece um controlador com o menor grau. Este método é chamado de solução de grau mínimo. Já que $\deg A > \deg B$, o termo de maior ordem no lado esquerdo da equação (2.22) é AR , então:

$$\deg R = \deg A_c - \deg A \quad (2.33)$$

Devido à equação (2.32), existe sempre uma solução em que $\deg S < \deg A = n$. Podemos então encontrar uma solução em que o grau de S é no máximo igual a $\deg A - 1$. Essa solução é denominada solução de grau mínimo para a equação diofantina. A condição $\deg S \leq \deg R$ implica que:

$$\deg A_c \geq 2 \deg A - 1 \quad (2.34)$$

Seguindo da equação (2.29) em que a condição $\deg T \leq \deg R$, implica-se que:

$$\deg A_m - \deg B'_m \geq \deg A - \deg B^+ \quad (2.35)$$

Somando-se $\deg B^-$ em ambos os lados, constata-se que (2.35) é equivalente a $\deg A_m - \deg B_m \geq d_0$. Isso significa que no caso discreto o atraso de tempo do modelo deve ser no mínimo igual ao atraso de tempo do processo, o que é uma condição bastante natural. Dessa forma a condição de causalidade (3.31) pode ser escrita da seguinte maneira:

$$\begin{aligned} \deg A_c &\geq 2 \deg A - 1 \\ \deg A_m - \deg B_m &\geq \deg A - \deg B = d_0 \end{aligned} \quad (2.36)$$

É normal escolher uma solução em que o controlador possui o menor grau possível. No caso discreto também é razoável pleitear que o controlador não possua nenhum atraso extra, isso implica que os polinômios R , S e T devem ser de mesmo grau (ASTROM & WITTENMARK, 2008).

2.5.5 Algoritmo Regulador Auto Ajustável Indireto

Combinando o estimador recursivo estendido de mínimos quadrados (RELS do inglês *'Recursive Extended Least Squares'*) dado pela equação (2.19) com o método de alocação de polos com grau mínimo (MDPP do inglês *'Minimum-degree pole placement'*) descrito na seção anterior obtém-se um regulador auto ajustável simples. O algoritmo para o projeto deste regulador é dado a seguir:

- I) Calcular os polos desejados da função de transferência em malha fechada B_m/A_m e o polinômio do observador A_o .
- II) Estimar os coeficientes dos polinômios A , B e C da eq. (2.10) utilizando o método recursivo estendido de mínimos quadrados dado por (2.19).
- III) Estimar os coeficientes dos polinômios R , S e T da lei de controle através das equações (2.22) e (2.30);
- IV) Calcular a saída do controlador através da lei de controle (2.20).

Os passos II, III e IV se repetem a cada período de amostragem. É importante ressaltar que podem haver variações no algoritmo dependendo do cancelamento de zeros do processo (ASTROM & WITTENMARK, 2008).

2.6 Medição de temperatura

2.6.1 Termopares

Um termopar é um sensor de temperatura composto por dois fios de metais diferentes unidos em uma das extremidades. O funcionamento de um termopar é fundamentado pelo chamado “Efeito Seebeck”, onde uma diferença de potencial surge como um efeito da

diferença de temperatura entre as junções de condutores de materiais diferentes, como é mostrado na figura 2.11.



Figura 2.11 – Princípio de funcionamento do termopar

Fonte: Embarcados (2016)

O modelo matemático de um termopar pode ser descrito a partir da seguinte equação:

$$V(t) = K_t \theta(t) + b \quad (2.37)$$

Onde $V(t)$ é a tensão fornecida pelo termopar, K_t o ganho, $\theta(t)$ é a temperatura na junta quente e b é uma constante (GUERRA, 2006).

Segundo Lourenço (2008 apud CORDEIRO, 2016) a relação entre a temperatura na junção e a tensão gerada é dada por:

$$\Delta V_c = S(T_{tc})T_{tc} \quad (2.38)$$

Onde $S(T_{tc})$ é o coeficiente de Seebeck, que depende do tipo de termopar e é função da temperatura T_{tc} . Essa dependência do coeficiente de Seebeck com a temperatura causa uma certa não-linearidade na relação tensão x temperatura, no entanto, para uma pequena faixa de temperatura a equação (2.38) pode ser simplificada como:

$$\Delta V_c = S * T_{tc} \quad (2.39)$$

Sendo S agora uma constante que depende do tipo de termopar. Na Tabela 2.4 são mostrados os diferentes tipos de termopares com os materiais que constituem cada tipo juntamente com faixa de temperatura de cada um.

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 usado em atmosferas 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 atmosferas oxidantes e inertes. Em ambientes redutores ou vácuo perde suas características termoeletrônicas. Adequado para uso em temperatura abaixo de zero.
K	Níquel-cromo/Níquel- alumínio (+) / (-)	-200 a 1260°C	Recomendáveis em atmosferas oxidantes ou inertes. Ocasionalmente pode ser usado abaixo de zero grau. Não deve ser usado em atmosfera redutoras e sulfurosas. Seu uso no vácuo é por curto período de tempo.
S R	Platina-10% Ródio/Platina (+) / (-) Platina-13% Ródio/Platina	0 a 1600°C 0 a 1600°C	Recomendáveis em atmosferas oxidantes ou inertes. Não deve ser usado abaixo de zero grau, no vácuo em atmosferas redutoras ou com vapores metálicos. Apresenta boa precisão em temperaturas elevadas.
B	Platina-30% Ródio/ Platina-6% Ródio (+) / (-)	600 a 1700°C	Recomendáveis em atmosferas oxidantes ou inertes. Não deve ser usado abaixo de zero grau, no vácuo, em atmosferas 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 1200C, curva FEMxTemp, similar ao tipo K, porém possui menor potência termoeletrônica, apresenta maior estabilidade e menor drift tempo.

Tabela 2.4 – Tipos de termopares

Fonte: CR resistências (2019)

3. METODOLOGIA

Este capítulo tem como objetivo apresentar as etapas realizadas até o projeto do controlador, desenvolvido para seguir um determinado perfil de temperatura. Um circuito de controle de potência em corrente alternada foi desenvolvido para poder controlar a tensão eficaz aplicada no forno. A aquisição da temperatura do forno foi feita através de um termopar tipo K e um módulo sensor com o CI MAX6675 que traz a compensação da junta fria do termopar e digitaliza o sinal. O controlador foi implementado digitalmente através de um programa gravado em uma placa Arduíno Mega 2560. Na figura 3.1 é mostrado o esquema da bancada de trabalho.

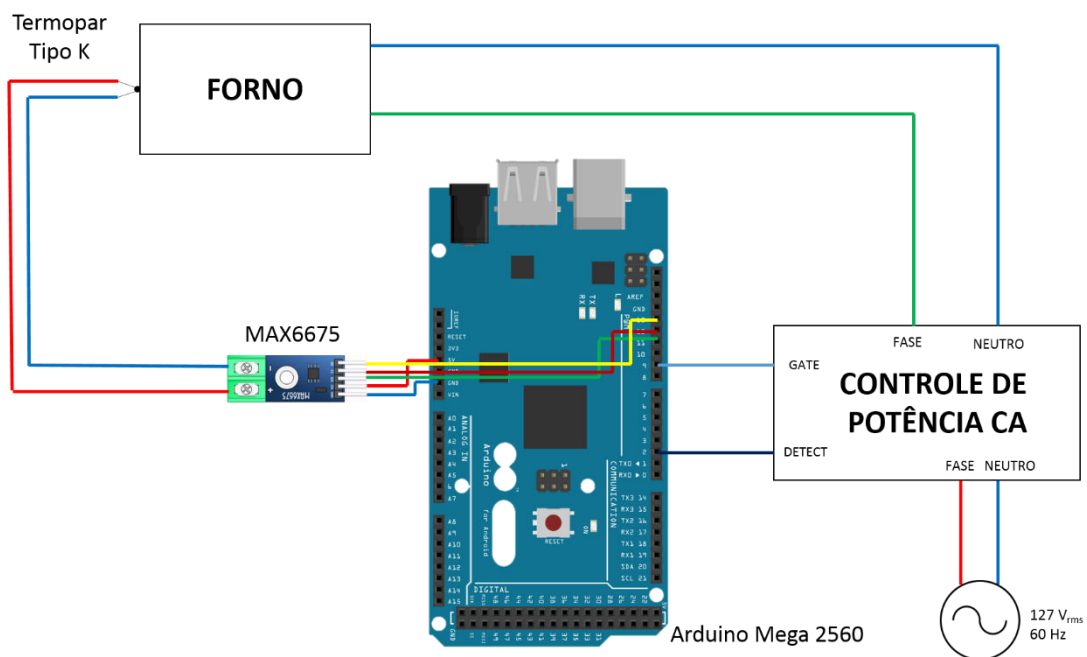


Figura 3.1 – Esquema da bancada de trabalho

Este trabalho foi desenvolvido no Laboratório de Eletrônica Analógica e Digital, pertencente ao Departamento de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto, em Ouro Preto, Minas Gerais.

3.1 Projeto do circuito de controle de potência em corrente alternada

Para controlar a tensão eficaz que é aplicada no forno projetou-se um circuito de controle de potência em corrente alternada, composto de um circuito detector de passagem por zero e um circuito de acionamento com o TRIAC BT138, capaz de suportar uma corrente de até 12 A_{rms}. Na figura 3.2 é mostrado o esquema elétrico do circuito detector de passagem por zero.

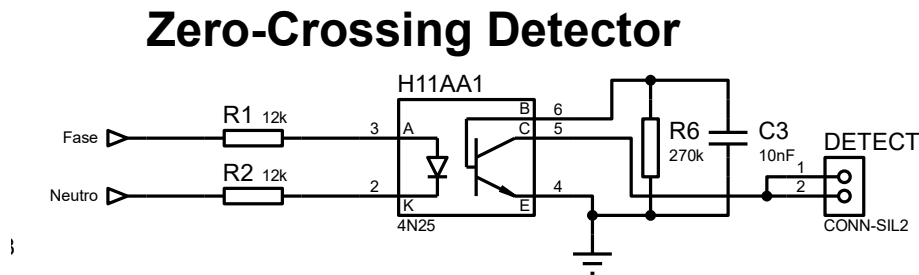


Figura 3.2 – Esquema elétrico detector de passagem por zero

Neste circuito utilizou-se o optoacoplador H11AA1. Quando a tensão da rede é diferente de zero, o fototransistor do H11AA1 funciona como uma chave fechada, permitindo a passagem de corrente, dessa forma a tensão na porta “DETECT”, definida como o pino 2 do Arduíno, é igual a zero. Sempre que a tensão da rede é igual a 0 V, o fototransistor do optoacoplador para de conduzir e, devido ao resistor de PULL-UP configurado via *firmware* no Arduíno, a tensão na porta “DETECT” passa a ser 5 V, dessa forma obtém-se um pulso de 5 V toda vez que a tensão da rede passa por zero. Quando esse pulso é recebido pelo Arduíno uma rotina de interrupção é executada, dentro dessa rotina uma variável contadora é incrementada, contando o número de passagens por zero da rede. Essa rotina de interrupção é responsável também por enviar o sinal que irá fazer com que o TRIAC, através da porta “GATE”, definida como o pino 9 do Arduíno, conduza ou deixe de conduzir, como é mostrado na figura 3.3.

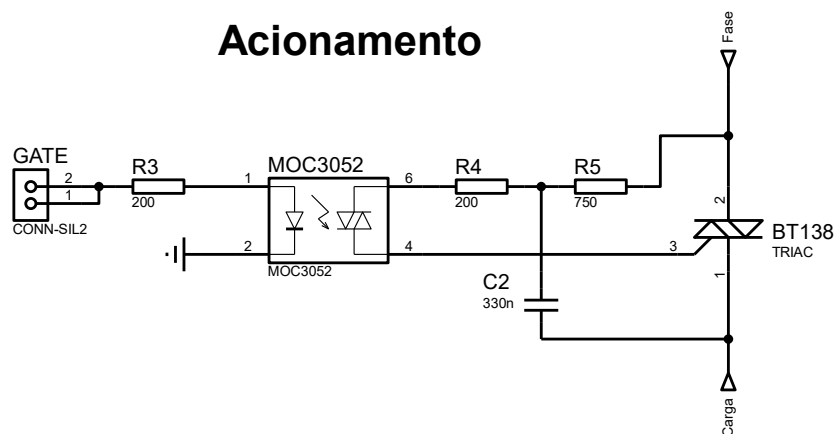


Figura 3.3 – Esquema elétrico acionamento do TRIAC

O circuito de acionamento consiste basicamente de um optoacoplador MOC3052, utilizado para isolar a parte de controle da parte de potência do circuito, e um TRIAC BT138. As resistências são utilizadas para limitar a passagem de corrente, tanto no LED do optoacoplador quanto no terminal do Gate do TRIAC, e o capacitor é utilizado para filtrar o sinal que chega no terminal do Gate, configuração para ambientes extremamente ruidosos descrita nas notas de aplicação do MOC3052 da Fairchild, impedindo que o TRIAC entre em regime de condução inapropriadamente. Quando um pulso é enviado através da porta “GATE” do Arduino, o LED interno do optoacoplador faz com que o foto diac passe a conduzir corrente, disparando assim o TRIAC e permitindo a passagem de corrente na carga.

O controle da tensão eficaz aplicada na carga é feito pelo método de trem de pulsos, ou controle por ciclos inteiros, uma vez que o controle é de temperatura e a constante de tempo nesse tipo de sistema é grande, então esse método é suficiente, sendo mais simples e mais robusto que o método por ângulo de disparo. Neste método para cada N ciclos completos da tensão da rede são aplicados K ciclos completos na carga, como é mostrado na figura 3.4. Dessa forma o valor eficaz da tensão aplicada na carga varia proporcionalmente à relação de N e K.

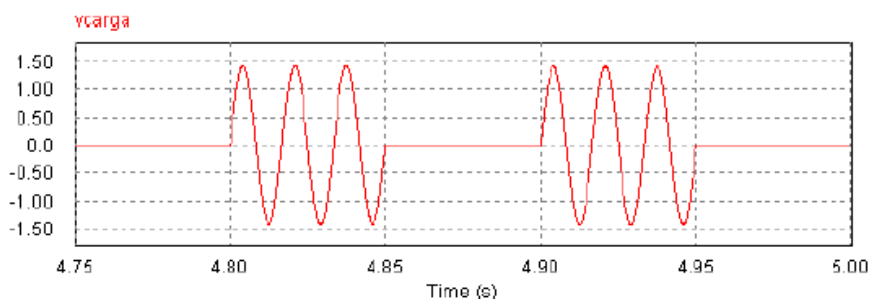


Figura 3.4 – Forma de onda da tensão na carga

Fonte: Apostila Eletrônica de Potência I PEA – 3487 (2017)

Para este trabalho fixou-se o valor de N em 512, dessa forma a tensão aplicada na carga depende somente do valor K, como é mostrado a seguir:

$$V_c = V_r * \sqrt{\frac{K}{N}} = 127 * \sqrt{\frac{K}{512}} \quad (3.1)$$

É importante ressaltar que nesse caso não é utilizada a definição usual de valor eficaz, que integra o valor em um ciclo da rede CA, a integração é feita em N ciclos da rede CA. O projeto da placa de circuito impresso (PCI) foi desenvolvido por meio do ARES, uma aplicação que faz parte do *software* Proteus Design Suite, programa da Labcenter Electronics que auxilia no desenvolvimento de PCIs. O projeto da placa de circuito impresso em escala real se encontra no Anexo 02, juntamente com a lista de componentes utilizados.

A alimentação do circuito de controle é realizada pela fonte HLK-PM01, que transforma os 127 V_{rms} em 5 V_{cc}. O esquema elétrico da fonte de alimentação é mostrado na figura 3.5.

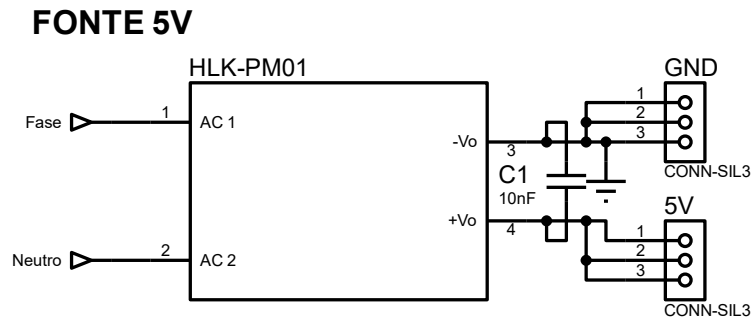


Figura 3.5 – Esquema elétrico alimentação do circuito de controle

3.2 Sistema de aquisição de temperatura

A aferição da temperatura do forno foi feita utilizando um termopar tipo K e um módulo sensor com o CI MAX6675, mostrados na figura 3.6, que traz a compensação da junta fria do termopar e realiza a conversão A/D do valor de tensão proporcionado pelo termopar. A conversão do valor de tensão fornecido pelo módulo sensor para °C foi feita usando a função “ktc.readCelsius()” da biblioteca “max6675.h” do Arduino.

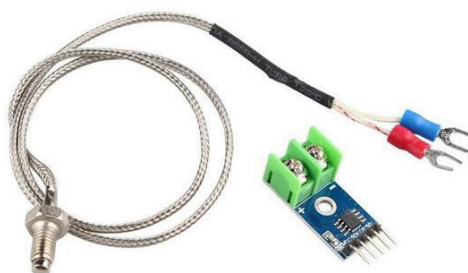


Figura 3.6 – Termopar tipo K e módulo sensor MAX6675

Fonte: Eletrogate (2019)

3.3 Identificação dos parâmetros do forno

Segundo AGUIRRE (2007) existem diversos aspectos importantes envolvidos em problemas reais de identificação, no entanto as principais etapas de um problema de identificação são:

- I. Testes dinâmicos e coleta de dados.
- II. Escolha da representação matemática a ser usada.
- III. Determinação da estrutura do modelo.
- IV. Estimação de parâmetros.
- V. Validação do modelo.

As seções a seguir apresentam os resultados obtidos nestas etapas para o forno estudado.

3.3.1 Testes dinâmicos e coletas de dados

Uma vez que a identificação de sistemas tem como objetivo obter modelos matemáticos a partir de dados de entrada e saída do sistema, é necessário gerar esses dados. Para incluir no modelo as características dinâmicas do forno excitou-se a planta com um sinal binário pseudo-aleatório de sequência m . Foram realizados testes em 3 patamares de temperatura, patamares estes, próximos aos pontos de operação para o perfil de temperatura ao qual o forno precisará seguir. Os dados referentes ao sinal de entrada para cada um dos 3 patamares são dados a seguir:

Para $T = 120^{\circ}\text{C}$:

$V_+ = 88.03 \text{ V};$ $V_- = 62.5 \text{ V};$ $m = 2;$ $t_s = 15 \text{ s}$ $N = 1024;$

Para $T = 175^{\circ}\text{C}$:

$V_+ = 109.99 \text{ V};$ $V_- = 84.38 \text{ V};$ $m = 2;$ $t_s = 15 \text{ s}$ $N = 1024;$

Para $T = 220^{\circ}\text{C}$:

$V_+ = 125 \text{ V};$ $V_- = 99.61 \text{ V};$ $m = 2;$ $t_s = 15 \text{ s}$ $N = 1024;$

Na figura 3.7 é mostrado o sinal PRBS aplicado no forno para $T = 120\text{ }^{\circ}\text{C}$. É importante ressaltar que para os 3 patamares de temperatura a mesma sequência de bits foi utilizada, variando então somente os valores de $V+$ e $V-$.

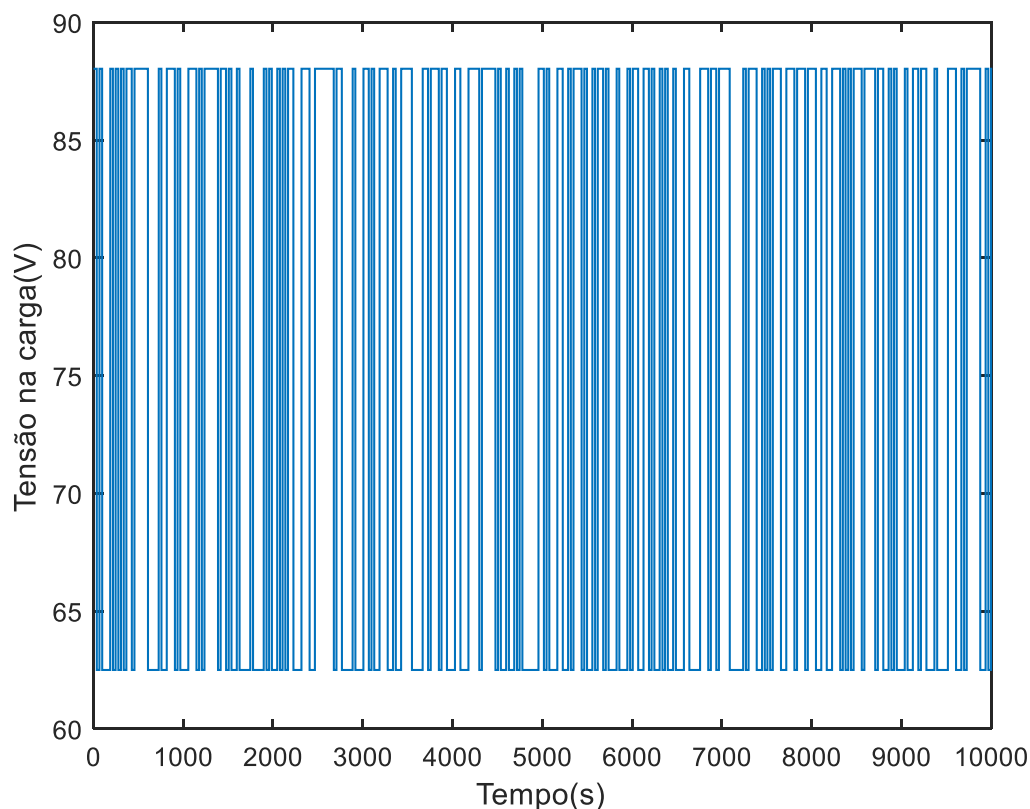


Figura 3.7 – Sinal PRBS

3.3.2 Representação matemática e estrutura do modelo

A escolha da representação matemática foi feita de forma empírica. Concluiu-se que o modelo ARX não identificou satisfatoriamente os parâmetros da planta, com isso optou-se em utilizar o modelo ARMAX, no qual o erro na equação é modelado como um processo de média móvel.

Assim como a representação matemática, a estrutura do modelo também foi obtida de forma empírica, testando diversas estruturas e escolhendo aquela que melhor conseguiu representar o modelo do forno. Sendo assim, o modelo matemático que melhor se aproximou do modelo real do forno foi o seguinte:

$$y(k) = a_1y(k-1) + a_2y(k-2) + b_0u(k-1) + b_1u(k-2) + c_1e(k) + c_2e(k-1) \quad (3.2)$$

Onde $y(k)$, $u(k)$ e $e(k)$ são a temperatura do forno, a tensão do forno e o erro do modelo, respectivamente, no instante k e os parâmetros a_1 , a_2 , b_0 , b_1 , c_1 e c_2 são os parâmetros a serem encontrados pelo estimador de mínimos quadrados estendido.

3.3.3 Estimação de parâmetros e validação do modelo

Uma vez que foram amostrados os dados de entrada e saída do sistema, a representação matemática e a estrutura do modelo foram definidas, estimou-se os parâmetros do modelo por meio de um algoritmo estimador de mínimos quadrados estendido. Os parâmetros encontrados para cada um dos patamares de temperatura são dados a seguir:

Para T = 120°C:

$$a_1 = 1.7715 \quad a_2 = -0.7783 \quad b_0 = 0.0074 \quad b_1 = 0.0038 \quad c_1 = -0.9268 \quad c_2 = 0.2940$$

Para T = 175°C:

$$a_1 = 1.8065 \quad a_2 = -0.8145 \quad b_0 = 0.0056 \quad b_1 = 0.0097 \quad c_1 = -0.9376 \quad c_2 = 0.1128$$

Para T = 220°C:

$$a_1 = 1.7031 \quad a_2 = -0.7162 \quad b_0 = 0.0072 \quad b_1 = 0.0168 \quad c_1 = -0.9276 \quad c_2 = 0.2148$$

Vale ressaltar que para a estimação dos parâmetros somente metade dos dados de entrada e saída foram utilizados, o restante dos dados foi utilizado para a validação do modelo estimado, como é mostrado nas figuras 3.8, 3.9 e 3.10.

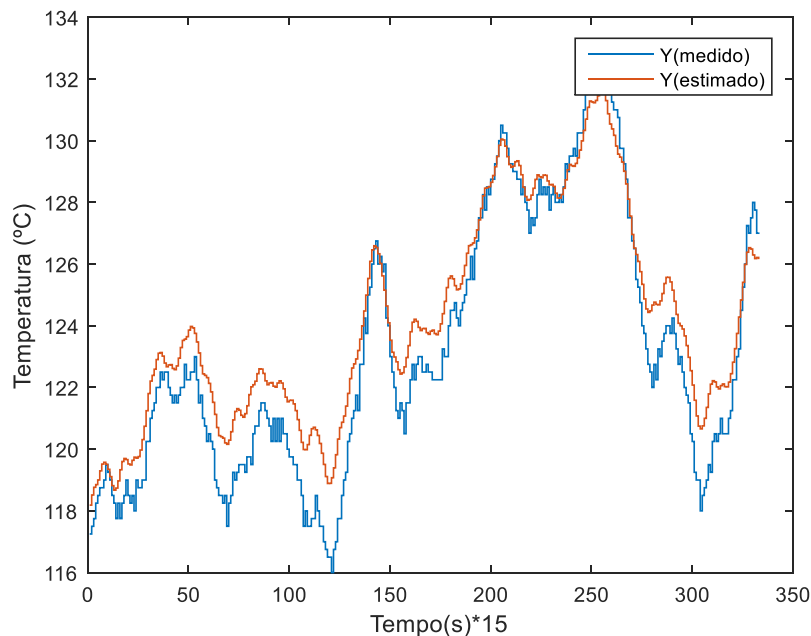


Figura 3.8 – Resposta temporal da temperatura do modelo escolhido (laranja) e da temperatura real medida (azul), para o patamar T = 120°C, utilizando dados de validação.

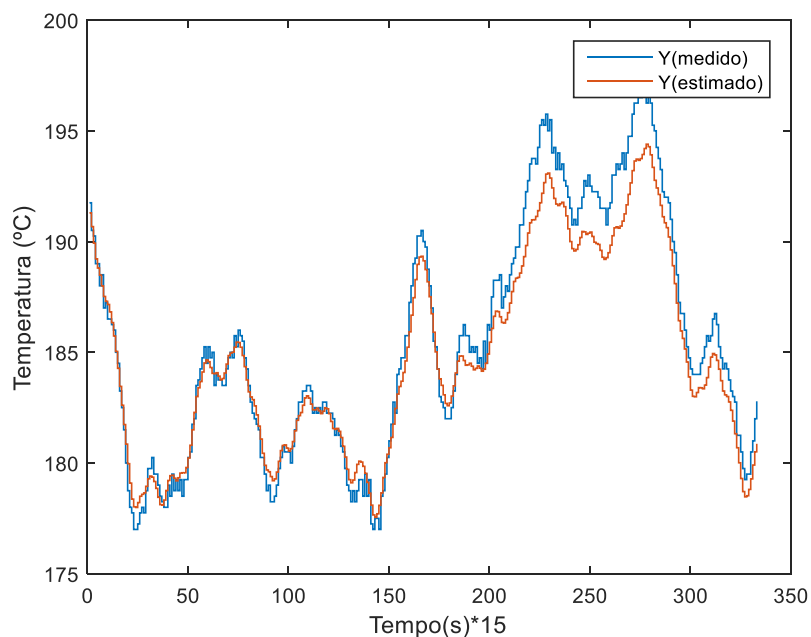


Figura 3.9 – Resposta temporal da temperatura do modelo escolhido (laranja) e da temperatura real medida (azul), para o patamar $T = 175^{\circ}\text{C}$, utilizando dados de validação.

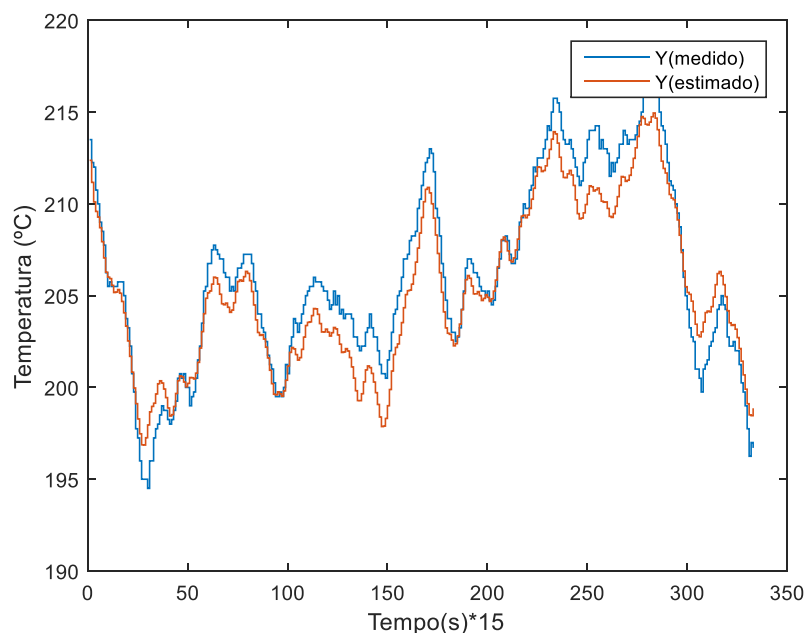


Figura 3.10 – Resposta temporal da temperatura do modelo escolhido (laranja) e da temperatura real medida (azul), para o patamar $T = 220^{\circ}\text{C}$, utilizando dados de validação.

3.4 Projeto do regulador auto ajustável indireto

Como foi visto no capítulo 2, um regulador auto ajustável pode ser obtido combinando um estimador recursivo estendido de mínimos quadrados com um método de alocação de polos com grau mínimo. O *firmware* do regulador foi implementado por meio da IDE do Arduino. O fluxograma do controle da tensão aplicada no forno é mostrado na figura 3.11.

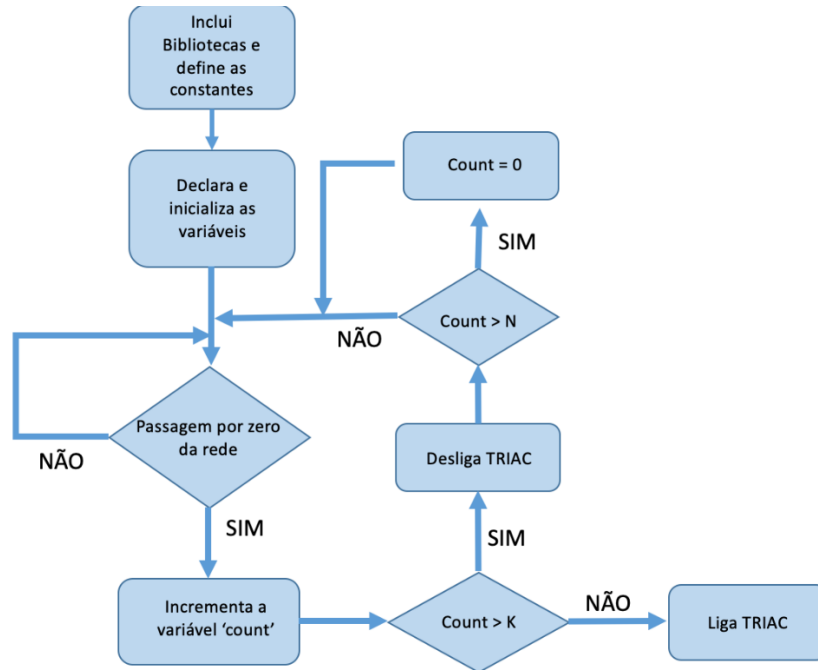


Figura 3.11 – Fluxograma Controle da tensão média

Inicialmente são incluídas as bibliotecas utilizadas e definidas as constantes. Em seguida são declaradas e inicializadas as variáveis que serão utilizadas ao longo do programa. O disparo do TRIAC ocorre dentro da interrupção ligada à porta “DETECT”, que é chamada a cada passagem por zero da rede. Isso é feito para que o TRIAC dispare somente quando $\alpha = 0$. Uma variável contadora é incrementada toda vez que essa rotina de interrupção é chamada, assim enquanto o valor dessa variável for menor que K o TRIAC é mantido acionado. Quando o valor da contagem é maior que K o TRIAC é desligado e, quando o valor da contagem é maior que N a variável contadora recebe zero e a contagem é então reiniciada.

O valor de K é calculado a partir da saída do controlador através da seguinte equação:

$$K = \frac{u^2}{V_r^2} * N \quad (3.3)$$

Onde u é a saída do controlador e V_r é a tensão eficaz da rede.

A programação completa do *firmware* pode ser visualizada no Anexo 01.

4. RESULTADOS

Este capítulo tem como objetivo apresentar os resultados obtidos quanto aos controladores utilizados. Na figura 4.1 é mostrada a bancada de trabalho desenvolvida no Laboratório de Eletrônica Analógica e Digital da Universidade Federal de Ouro Preto. O forno utilizado neste trabalho foi um forno elétrico 127 V com 700 W de potência

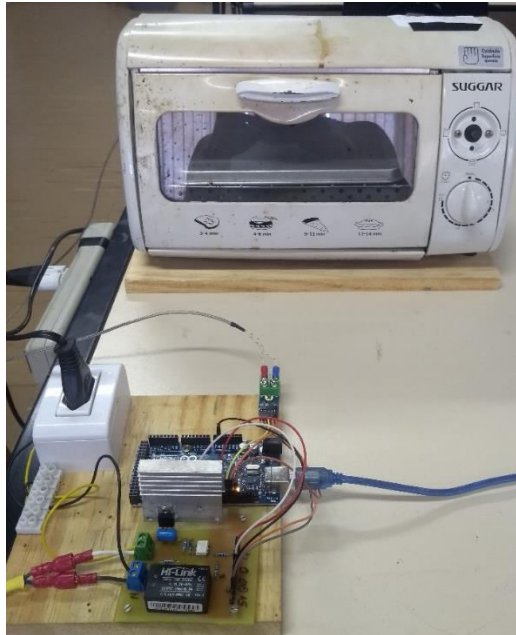


Figura 4.1 – Bancada de trabalho

4.1 Placa de circuito impresso para controle de potência CA

Nas figuras 4.2 e 4.3 são mostradas as faces inferior e superior da placa de circuito impressa desenvolvida para o controle de potência em corrente alternada.

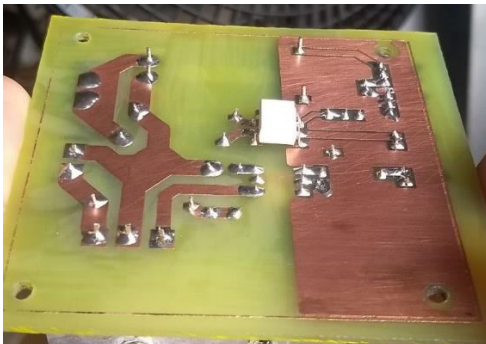


Figura 4.2 – Face inferior da PCI



Figura 4.3 – Face superior da PCI

4.2 Teste dos Controladores

Foram analisados 2 métodos adaptativos por alocação de polos, um com cancelamento de zeros e outro sem cancelamento de zeros do processo, para um setpoint de 100°C. Na figura 4.4 é mostrado uma comparação entre os dois controladores quando aplicados ao forno.

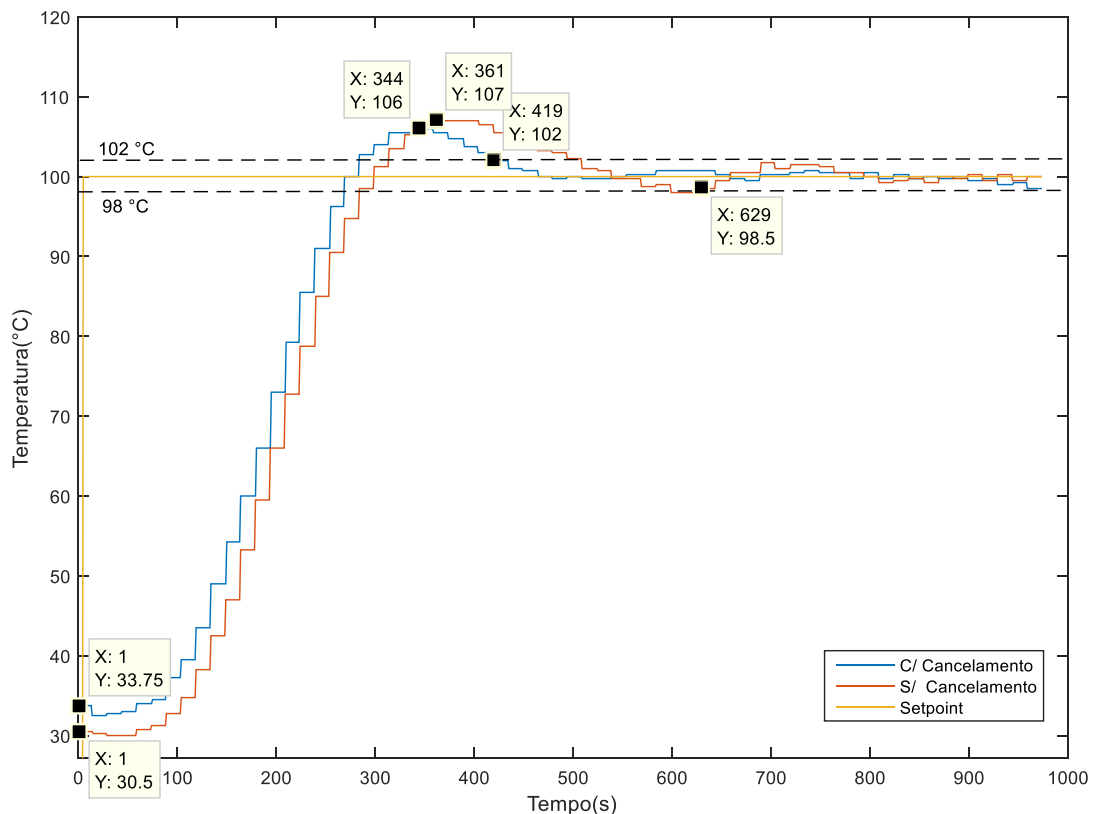


Figura 4.4 – Resposta da temperatura medida do forno real a uma referência em degrau unitário de 100°C.

Os polos de ambos os controladores foram alocados de forma que o sistema obtivesse um sobressalto de 10% do degrau aplicado e um tempo de acomodação ($\pm 2\%$) de aproximadamente 333 segundos, cerca de 5 minutos. É possível observar que o controlador com cancelamento de zeros do processo possui um menor tempo de acomodação, entrando em regime permanente por volta dos 419 segundos, se aproximando melhor das características desejadas no projeto do controlador. Portanto é possível dizer que o controlador com cancelamento de zeros do processo possui uma resposta melhor se comparado ao controlador sem cancelamentos de zeros. As saídas dos

controladores com e sem cancelamento são mostradas nas figuras 4.5 e 4.6 respectivamente.

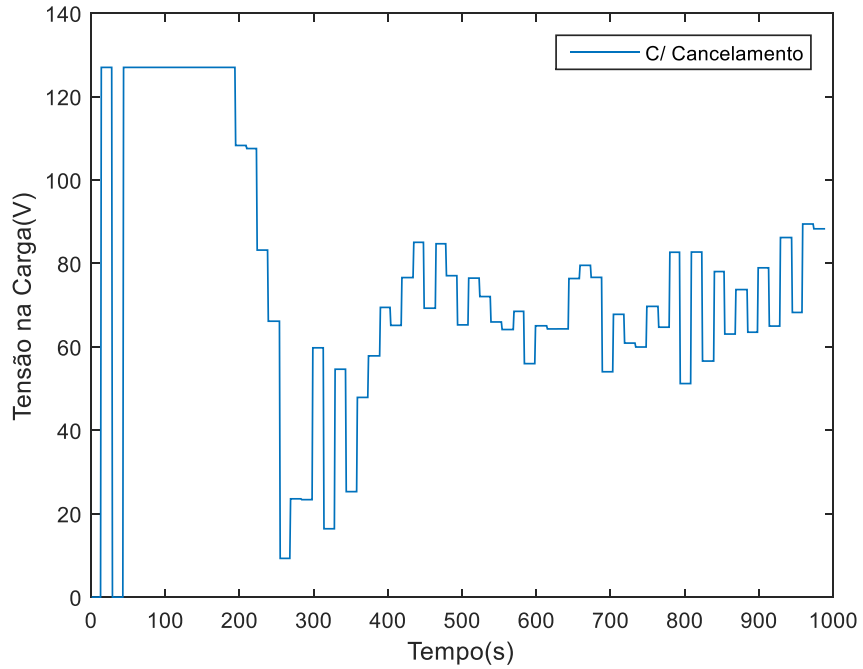


Figura 4.5 – Saída do controlador com cancelamento de zeros

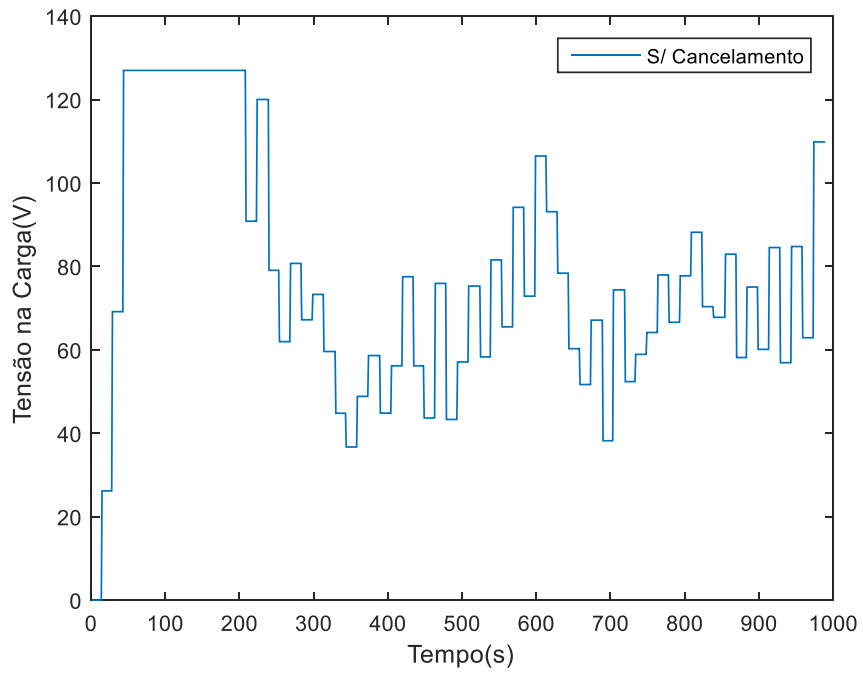


Figura 4.6 – Saída do controlador sem cancelamento de zeros

4.3 Resultados da identificação dos parâmetros do forno

Nas figuras 4.7 e 4.8 são mostrados os gráficos de convergência dos parâmetros do sistema para os controladores com e sem cancelamento de zeros do processo respectivamente.

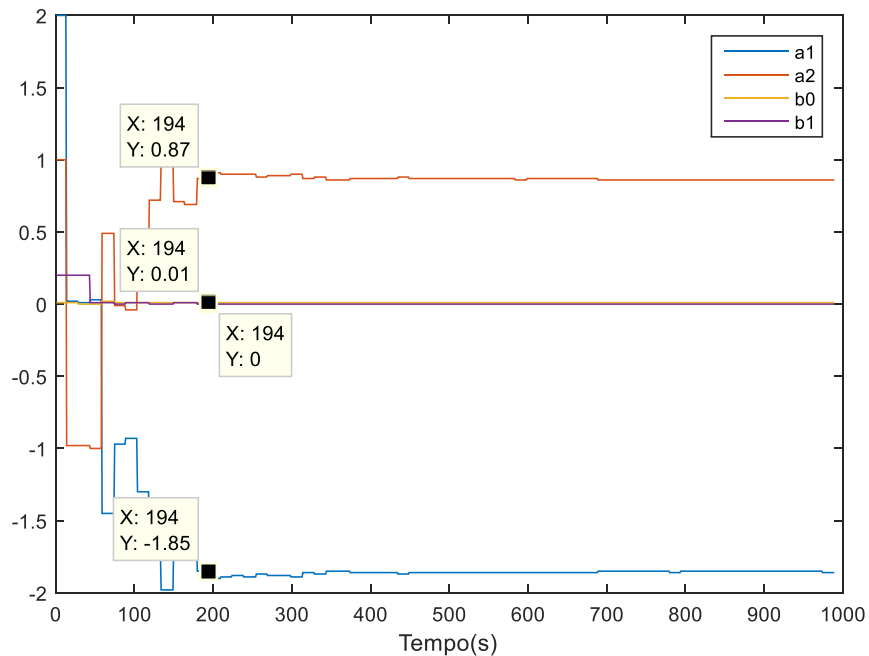


Figura 4.7 – Convergência dos parâmetros para o controlador com cancelamento

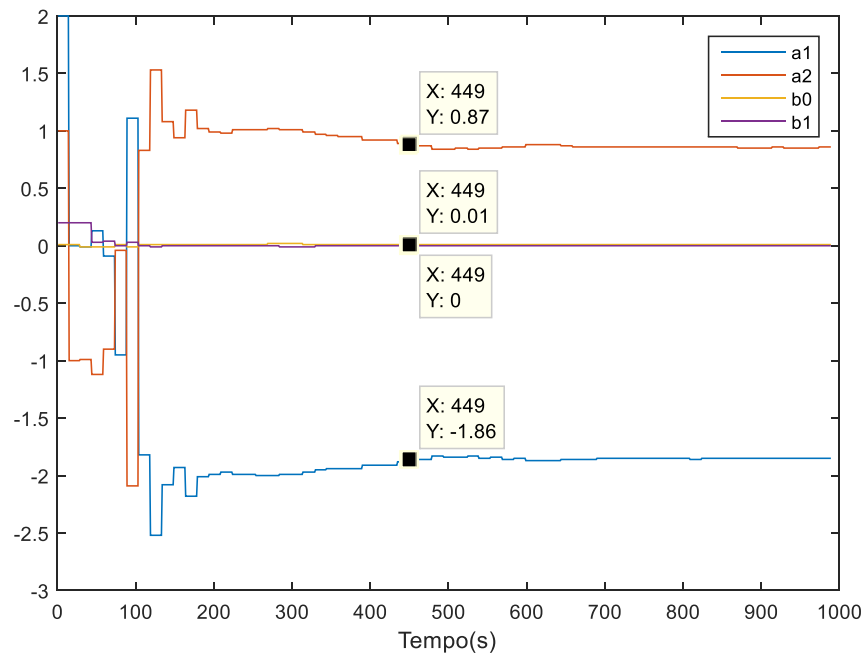


Figura 4.8 – Convergência dos parâmetros para o controlador sem cancelamento

A partir dos gráficos é possível observar que os parâmetros do sistema convergem mais rápido no controlador com cancelamentos de zeros. Segundo Astrom (2008) essa diferença acontece porque a saída do controlador com cancelamentos de zeros possui uma oscilação maior em seus valores, isso melhora a excitação do sistema e facilita a identificação de seus parâmetros. Essa diferença no tempo de acomodação dos parâmetros estimados explica a diferença nos tempos de acomodação da saída do sistema, pois uma vez que esses parâmetros convergem mais rápido para o valor real do sistema é natural que o controlador obtenha uma resposta melhor.

4.4 Perfil *reflow* projetado

Na figura 4.9 é mostrado o perfil de temperatura '*reflow*', este perfil foi obtido através de uma simulação do sistema no Arduino, utilizando os parâmetros do sistema estimados na seção (3.3.4) para $T = 220^{\circ}\text{C}$.

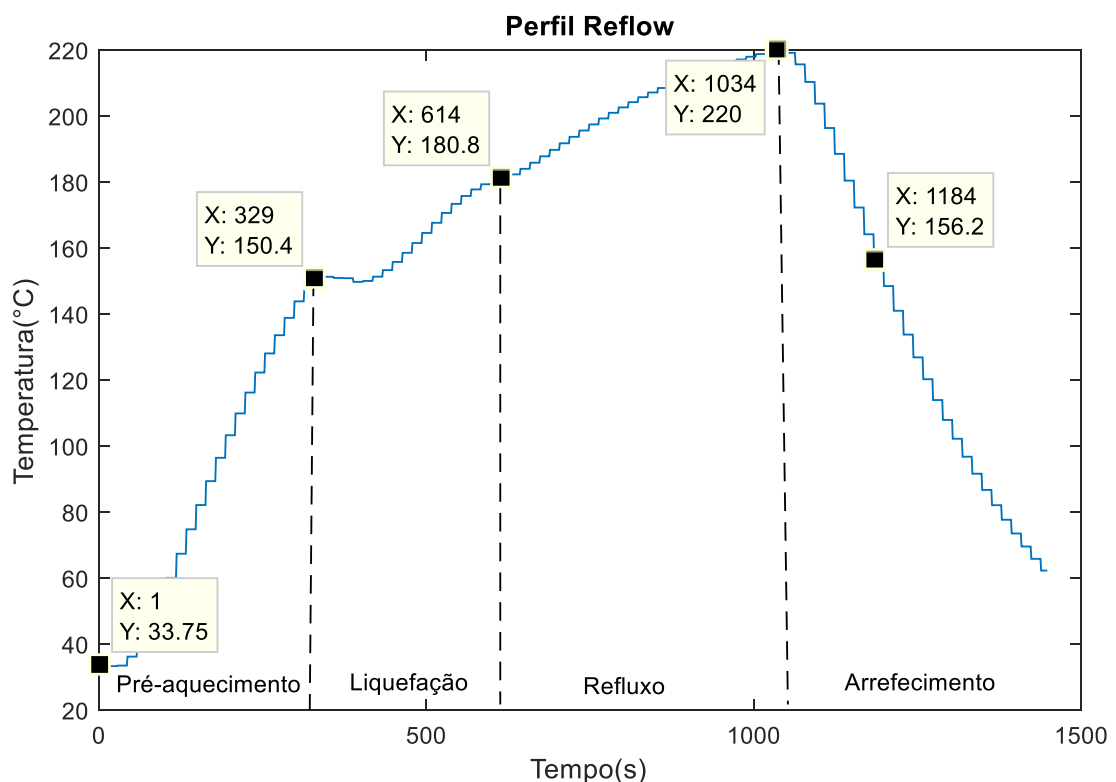


Figura 4.9 – Perfil de temperatura '*reflow*'

Segundo Altera (2011) o tempo para o forno ir de 25°C até a temperatura de pico deve ser de no máximo 360 segundos, 6 minutos. É possível observar na figura 4.9 que o forno utilizado nesse trabalho gastou 1034 segundos, cerca de 17 minutos, para ir da temperatura ambiente até a temperatura de pico. Dessa forma é possível dizer que o forno

utilizado não é adequado para este tipo de aplicação, pois possui um tempo de acomodação grande, como é possível perceber na figura 4.4, onde o melhor tempo de acomodação foi de 419 segundos.

5. CONCLUSÕES E RECOMENDAÇÕES

Este capítulo apresenta as conclusões obtidas com este trabalho e algumas sugestões para trabalhos a serem realizados no futuro.

5.1 Conclusões

O forno utilizado não se mostrou adequado para esse tipo de aplicação, pois possui um tempo de acomodação e não conseguiu cumprir com os requisitos de tempo do perfil de temperatura do tipo refluxo. O controlador foi capaz de identificar os parâmetros da planta de forma recursiva, sendo o controlador com cancelamento de zeros do processo o que apresentou uma convergência mais rápida dos parâmetros e, conseqüentemente, obteve um tempo de acomodação melhor.

É possível dizer que o controlador projetado neste trabalho se caracteriza como um controlador adaptativo, pois demonstrou ser capaz de identificar os parâmetros do sistema a ser controlado e de ajustar seus parâmetros de acordo com as variações do sistema. Apesar do forno utilizado neste trabalho não permitir que os requisitos de referência para a temperatura fossem alcançados, os resultados foram satisfatórios dentro dessas limitações. Tudo indica que o controlador deve ser capaz de alcançar estes requisitos ao utilizar-se um forno de potência mais elevada.

5.2 Recomendações para trabalhos futuros

Para trabalhos futuros sugere-se a utilização de um forno com maior potência e que conseqüentemente apresente um tempo de acomodação menor, a fim de se obter o perfil de temperatura adequado. Sugere-se também o desenvolvimento de um sistema supervisor, para acompanhar e analisar o perfil de temperatura obtido além de permitir a seleção do tipo de solda a ser realizado, a fim de mudar o perfil de temperatura ao qual o forno deverá seguir.

REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à identificação de sistemas – Técnicas lineares e não-lineares aplicadas a sistemas reais**. 3. Ed. Belo Horizonte: Editora UFMG, 2007.
- ALTERA CORPORATION. **SMT Board Assembly Process Recommendations**. Outubro de 2011.
- ASTROM, K. J.; WITTENMARK, B. **Adaptive control**. 2. Ed. Nova York: Dover publications, 2008.
- CORDEIRO, D. A. S. **Uso do compensador PID no Controle de Temperatura de um forno elétrico a resistência para soldagem de componentes SMD**. Universidade Federal de Ouro Preto, 2016. 88p.
- 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.
- GOOSENS, P. **Controlador para forno de componentes SMD**. Elektor, setembro de 2006.
- INSTITUTO NEWTON C. BRAGA. **Componentes SMD do PC**. Disponível em: <<http://www.newtonbraga.com.br/index.php/novos-componentes/52-artigos-tecnicos/artigos-diversos/7714-componentes-smd-do-pc-art935>> Acesso em: janeiro de 2019.
- KOMATSU, W.; JUNIOR, L. M.; KAISER, W. **Eletrônica de Potência I**. 01 ago. 2017, 15 dez. 2017. 156p. Notas de Aula.
- K. W. WAIT. **Adaptive temperature control of a class of home refrigerators**. 2012 American Control Conference (ACC), Montreal, QC, 2012, pp. 380-385. doi: 10.1109/ACC.2012.63147
- LU, C. H.; TSAI, C. C. **Adaptive Predictive Control With Recurrent Neural Network for Industrial Processes: An Application to Temperature Control of a Variable-Frequency Oil-Cooling Machine**. IEEE Transactions on Industrial Electronics, vol. 55, no. 3, pp. 1366-1375, March 2008. doi: 10.1109/TIE.2007.896492
- MARTINS, N. A. **Sistemas Microcontrolados**. 1. Ed. São Paulo: Novatec Editora, 2005.
- PAWLOWSKI, P.; DABROWSKI, A. **Reflow oven for heating and soldering SMD and BGA components**. IEEE 25th International Conference Mixed Design of Integrated Circuits and Systems, 21-23 junho 2018. p.324-329. Polônia, 2018.

PUHLMANN, H. F. **Trazendo o mundo real para dentro do processador – Conversor A/D**. Setembro de 2015. Disponível em: <<https://www.embarcados.com.br/conversor-a-d/>> Acesso em: janeiro de 2019

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

SOUZA, D. R.; SOUSA, D. J. **Desbravando o PIC18: Ensino didático**. 1 Ed. São Paulo: Editora Érica, 2012.

THOMSEN, A. **O que é Arduíno**. Setembro de 2014. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-arduino/>> Acesso em: janeiro de 2019.

TSAI, T. -N. **Thermal parameters optimization of a reflow soldering in printed circuit board assembly: A comparative study**. Applied Soft Computing. n. 12, p. 2601-2613. Taiwan, 2012.

ANEXO 01
(FIRMWARE DO MICROCONTROLADOR)

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////  ALGORITMO REGULADOR SELF-TUNING  \\\\\\\\\\\\\\\\\\\
// RECURSIVE EXTENDED LEAST SQUARES (RELS) E MINIMUM
//////////          DEGREE POLE PLACEMENT (MDPP)          //////////
//   Rafael O. Quintão 01/2019 v1.2                      //////////
//   Modelo  $H(q) = (b_0*q + b_1)/(q^2+a_1*q+a_2)$           //////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// inclui bibliotecas
#include <avr/io.h>
#include <math.h>
#include <avr/interrupt.h>
#include "max6675.h" // Biblioteca Modulo de Aquisição de temperatura

#define DETECT 2 //Pino detecta a passagem por zero da rede
#define GATE 9 //Pino que aciona TRIAC
#define ktcSO 13 // Saída serial do modulo do sensor de temperatura
#define ktcCS 12 // Pino CS do modulo do sensor de temperatura
#define ktcSCK 11 // Pino de de clock para comunicação serial
#define runEvery(t) for (static uint16_t _lasttime;
(uint16_t)((uint16_t)millis() - _lasttime) >= (t); _lasttime += (t))
#define N 512 // numero de ciclos totais, cada ciclo corresponde a 2
passagens por 0 da rede
#define Vref 127 //Tensão eficaz da rede

//Variaveis Globais
// e = resíduo, n= inovação, w = frecuencia natural, z = fator de
amortecimento
// y = y(k), y1 = y(k-1), u = u(k), u1 = u(k-1), lambda = fator de esquecimento

```

```

// ysp = SetPoint(k), ysp1 = SetPoint(k-1) , eps = inovação,
float y, u, y1, u1, w, z, e, h, lambda, ysp, ysp1, n, eps, denom;
float x1, x2, x3, x4, x5, x6; // Vetor de regressores x = [y(k-1) y(k-2) u(k-1)
u(k-2) e(k-1) e(k-2)]
float k1, k2, k3, k4, k5, k6; // Vetor de pesos K
float th1, th2, th3, th4, th5, th6; // Vetor de parametros Theta = [a1, a1, b1,
b2, c1, c2]
float p11, p12, p13, p14, p15, p16; // Matriz de Covariância
float    p22, p23, p24, p25, p26;
float    p33, p34, p35, p36;
float    p44, p45, p46;
float    p55, p56;
float    p66;
// v= v(k), v1 = v(k-1), [a, am1, am2, aop, ao] são os parametros para se obter
os polos desejados
float v, v1, a, am1, am2, aop, ao;
// Resultado da operação matricial PxX
float px1, px2, px3, px4, px5, px6;
// [umax, umin] limite dos valores de entrada, [a1, a2, b0, b1] são os
parametros do modelo
float umax, umin, a1, a2, b0, b1;
float r10, w1, s00, w2, s10, as, bs, bm0; // parametros do controlador s/
cancelamento de zeros
float r1, s0, s1, t0, t1; // Parametros do controlador c/ cancelamento de zeros
bool cancel = 1; // Se 1 Controlador c/ cancelamento de zeros, se 0
controlador s/ cancelamento de zeros
unsigned int tempo = 0; // Tempo para controlar o perfil de temperatura
'reflow'

```

```

// Variáveis de acionamento do TRIAC
float Tamb; // Temperatura ambiente
unsigned int count = 0; // Conta o número de ciclos decorridos; cada ciclo
corresponde a duas passagens por zero
unsigned int K = 0; // K = numero de ciclos com TRIAC conduzindo
bool r = 1, b = 0, c = 0; // Se r=0 desaciona TRIAC, se r=1 aciona TRIAC;
se b = 0 sistema de controle de potencia CA desligado, se 1 sistema ligado

//Rotinas de interrupção
void zeroCrossingInterrupt() { //Detecção de passagem por 0 na rede
  if (c) {
    count++; // conta o numero de ciclos decorridos
    if (r) {
      digitalWrite(GATE, HIGH); // Aciona o TRIAC
    } else {
      digitalWrite(GATE, LOW); // Desliga o TRIAC
    }
  }
}

// Decalara Função do termopar
MAX6675 ktc(ktcSCK, ktcCS, ktcSO);

void setup() {
  // Comunicação Serial

```

```

Serial.begin (9600); // Define o baud rate em 9600
// Inicializa pinos de entrada e saída
pinMode(DETECT, INPUT_PULLUP); //Detecção da passagem por
zero da rede
pinMode(GATE, OUTPUT); //Disparo do TRIAC
// Inicializa interrupção no pino 2
attachInterrupt(digitalPinToInterrupt(DETECT), zeroCrossingInterrupt,
RISING);
// Chama zeroCrossingInterrupt na borda de subida
//Inicializa as variáveis
Tamb = ktc.readCelsius(); // Lê Temperatura ambiente do Forno
z = 0.6; // Constante de amortecimento OS = 10%
w = 0.02; // Frequencia natural
h = 15; // período de amostragem
lambda = 0.990; // Fator de esquecimento
y = Tamb; //Lê temperatura do forno
y1 = y;
ysp = 0;
u = 0;
u1 = 0;
x1 = -y;
x2 = -y;
x3 = x4 = x5 = x6 = 0; // Inicializa vetor de regressores
k1 = k2 = k3 = k4 = k5 = k6 = 0; // Inicializa vetor de ganhos
th1 = a1 = 2; // Inicializa vetor de parametros com valores obtidos
anteriormente
th2 = a2 = 1; // para facilitar a convergência
th3 = b0 = 0.01;

```

```

th4 = b1 = 0.2;
th5 = 0;
th6 = 0;
p11 = p22 = 100; // Inicializa matriz de covariancia
p33 = p44 = 1;
p55 = p66 = 10;
p12 = p13 = p14 = p15 = p16 = 0;
p23 = p24 = p25 = p26 = 0;
p34 = p35 = p36 = 0;
p45 = p46 = 0;
p56 = 0;
//Limite do sinal de controle
v1 = 0;
v = 0;
ysp1 = 0;
umax = Vref;
umin = 0;
//Calcula Am e Ao discretos
a = exp(-z * w * h);
am1 = -2 * a * cos(w * h * sqrt(1 - z * z));
am2 = a * a;
}

void loop() {
    //////////////////////////////////////
    //////////////////////////////////////Estimador REMQ////////////////////////////////////
    //////////////////////////////////////
    if (b) {

```

```

// Perfil reflow
if (tempo < 360) { // Fase de pré-aquecimento
    ysp = 150;
    w = 0.05;
    z = 0.6;
    //Recalcula Am e Ao discretos
    a = exp(-z * w * h);
    am1 = -2 * a * cos(w * h * sqrt(1 - z * z));
    am2 = a * a;
} else if (tempo < 584) { // Fase de liquefação
    ysp = 180;
    w = 0.018;
    z = 0.6;
    //Recalcula Am e Ao discretos
    a = exp(-z * w * h);
    am1 = -2 * a * cos(w * h * sqrt(1 - z * z));
    am2 = a * a;
} else if (tempo < 1033) { // Fase de refluxo
    ysp = 220;
    w = 0.4;
    z = 0.5;
    //Recalcula Am e Ao discretos
    a = exp(-z * w * h);
    am1 = -2 * a * cos(w * h * sqrt(1 - z * z));
    am2 = a * a;
} else { // Fase de arrefecimento
    ysp = Tamb;
    w = 0.01;

```



```

z = 0.6;
//Recalcula Am e Ao discretos
a = exp(-z * w * h);
am1 = -2 * a * cos(w * h * sqrt(1 - z * z));
am2 = a * a;
}
runEvery(15000) { // tempo de amostragem = 15s
// y = -1.7031 * x1 + 0.7162 * x2 + 0.0072 * x3 + 0.0168 * x4 - 0.9276 *
x5 + 0.2148 * x6; // Modelo para simulação do forno
y = ktc.readCelsius(); //Le valor da temperatura y(k)
//Calculo do vetor de ganhos K
px1 = p11 * x1 + p12 * x2 + p13 * x3 + p14 * x4 + p15 * x5 + p16 * x6;
px2 = p12 * x1 + p22 * x2 + p23 * x3 + p24 * x4 + p25 * x5 + p26 * x6;
px3 = p13 * x1 + p23 * x2 + p33 * x3 + p34 * x4 + p35 * x5 + p36 * x6;
px4 = p14 * x1 + p24 * x2 + p34 * x3 + p44 * x4 + p45 * x5 + p46 * x6;
px5 = p15 * x1 + p25 * x2 + p35 * x3 + p45 * x4 + p55 * x5 + p56 * x6;
px6 = p16 * x1 + p26 * x2 + p36 * x3 + p46 * x4 + p56 * x5 + p66 * x6;
denom = lambda + x1 * px1 + x2 * px2 + x3 * px3 + x4 * px4 + x5 *
px5 + x6 * px6;
k1 = px1 / denom;
k2 = px2 / denom;
k3 = px3 / denom;
k4 = px4 / denom;
k5 = px5 / denom;
k6 = px6 / denom;
// Calcula inovacao
eps = y - x1 * th1 - x2 * th2 - x3 * th3 - x4 * th4 - x5 * th5 - x6 * th6;
// Atualiza parametros Theta

```

```

th1 = th1 + k1 * eps;
th2 = th2 + k2 * eps;
th3 = th3 + k3 * eps;
th4 = th4 + k4 * eps;
th5 = th5 + k5 * eps;
th6 = th6 + k6 * eps;
// Atualiza Matriz de Covariancia
p11 = (p11 - px1 * k1) / lambda;
p12 = (p12 - px2 * k1) / lambda;
p13 = (p13 - px3 * k1) / lambda;
p14 = (p14 - px4 * k1) / lambda;
p15 = (p15 - px5 * k1) / lambda;
p16 = (p16 - px6 * k1) / lambda;
p22 = (p22 - px2 * k2) / lambda;
p23 = (p23 - px3 * k2) / lambda;
p24 = (p24 - px4 * k2) / lambda;
p25 = (p25 - px5 * k2) / lambda;
p26 = (p26 - px6 * k2) / lambda;
p33 = (p33 - px3 * k3) / lambda;
p34 = (p34 - px4 * k3) / lambda;
p35 = (p35 - px5 * k3) / lambda;
p36 = (p36 - px6 * k3) / lambda;
p44 = (p44 - px4 * k4) / lambda;
p45 = (p45 - px5 * k4) / lambda;
p46 = (p46 - px6 * k4) / lambda;
p55 = (p55 - px5 * k5) / lambda;
p56 = (p56 - px6 * k5) / lambda;
p66 = (p66 - px6 * k6) / lambda;

```

```

//Calcula Resíduo
e = y - x1 * th1 - x2 * th2 - x3 * th3 - x4 * th4 - x5 * th5 - x6 * th6 ;
// Fim do RELS (RECURSIVE EXTENDED LEAST SQUARES)
/////////////////////////////////////////////////////////////////
//////////////////// PROJETO DO CONTROLADOR MDPP \\\\\\\\\\\\\\\\\\\\\\\
/////////////////////////////////////////////////////////////////
// Renomeia parametros
a1 = th1;
a2 = th2;
b0 = th3;
b1 = th4;
// Resolve a equacao Diophantine AR + BS = AoAm
n = b1 * b1 - a1 * b0 * b1 + a2 * b0 * b0;
r10 = (ao * am2 * (b0 * b0) + (a2 - am2 - ao * am1) * b0 * b1 + (ao +
am1 - a1) * (b1 * b1)) / n;
w1 = (a2 * am1 + a2 * ao - a1 * a2 - am2 * ao) * b0;
s00 = (w1 + (-a1 * am1 - a1 * ao - a2 + (a1 * a1) + am2 + am1 * ao) *
b1) / n;
w2 = (-a1 * am2 * ao + a2 * am2 + a2 * am1 * ao - (a2 * a2)) * b0;
s10 = (w2 + (-a2 * am1 - a2 * ao + a1 * a2 + am2 * ao) * b1) / n;
// Calcula o polinomio T+Ao*Am(1)/B(1)
bs = b0 + b1;
as = 1 + am1 + am2;
bm0 = as / bs;
// Escolhe algoritmo de controle
if (cancel) { // parametros do controlador c/ cancelamento de zeros
r1 = b1 / b0;
s0 = (am1 - a1) / b0;

```



```

x2 = x1;
x1 = -y;
x4 = x3;
x3 = u;
x6 = x5;
x5 = e;
// Atualiza Estado do Controlador
y1 = y;
u1 = u;
v1 = v;
y1 = ysp;
}
// Imprime a tensão(u) e a temperatura do forno(y) e a temperatura
desejada(ysp)
runEvery(1000) {
  Serial.print(a1);
  Serial.print(",");
  Serial.print(a2);
  Serial.print(",");
  Serial.print(b0);
  Serial.print(",");
  Serial.print(b1);
  Serial.print(",");
  Serial.print(u);
  Serial.print(",");
  Serial.print(ysp);
  Serial.print(",");
  Serial.println(y);
}

```

```

    tempo++; //Incrementa tempo
  }
}
//Liga ou desliga TRIAC
runEvery(2) {
  if (count > N) { //Compara se o valor da contagem é igual ao numero de
ciclos totais
    count = 0; // -> Reinicia contagem
    r = 1;
  }
  if (count > K) { // Compara se a contagem é igual ao valor de K
    r = 0; // -> Desliga TRIAC
  } else {
    r = 1; //-> Liga TRIAC
  }
}

// Altera valor do setpoint pela porta serial
runEvery(50) {
  if ( Serial.available() > 0)
  {
    char ch = Serial.read();
    switch (ch) {
      case 't':
        ysp = lenumero(ch);
      case 'b':
        b = 1;
        break;
    }
  }
}

```

```
    }  
  }  
}  
}
```

```
long int lenumero(char caracter)
```

```
{
```

```
  /*
```

Função responsável por interpretar o que é dígito enviado pela comunicação serial ao arduino e concatená-los

maneira a formar um número inteiro.

```
  */
```

```
  long int valor = 0;
```

```
  int sinal = 1;
```

```
  while (caracter != 10)           // verifica se eh um digito entre 0 e 9
```

```
  {
```

```
    caracter = Serial.read();
```

```
    if (caracter >= '0' && caracter <= '9') // verifica se eh um digito entre 0 e
```

```
9
```

```
    valor = (valor * 10) + (caracter - '0'); // se sim, acumula valor
```

```
    else if ( caracter == '-')
```

```
      sinal = -1;
```

```
    else           // em caso de nao ser um numero ou simbolo
```

```
de menos termina o valor
```

```
  {
```

```
    valor = valor * sinal ;           // seta a variavel valor com o valor
```

```
acumulado
```

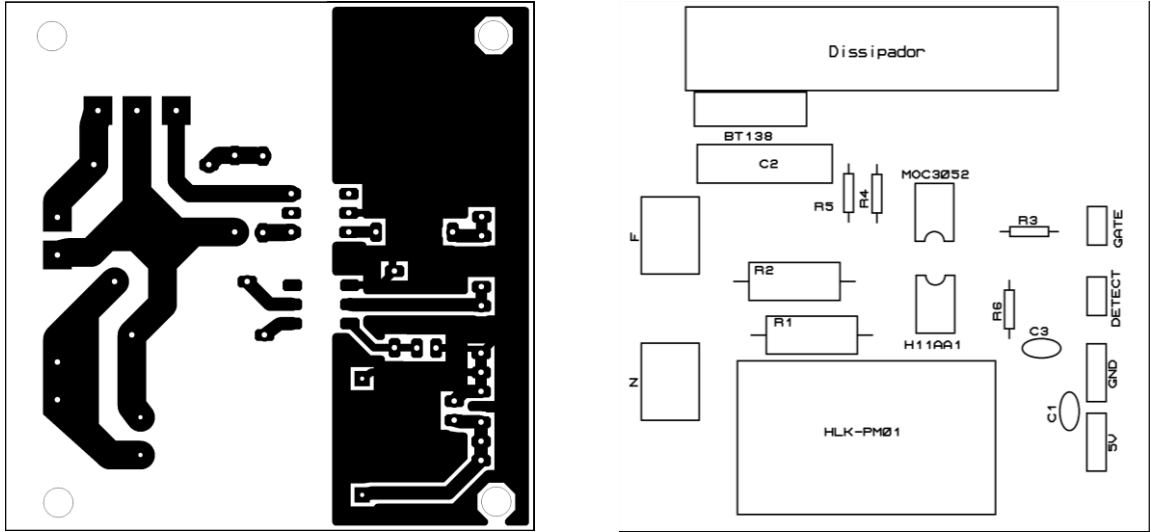
```
    return (valor);
```

```
// valor = 0;
sequencia de digitos
    sinal = 1;
}
}
}
```

// reseta valor para 0 para a proxima

ANEXO 02
(PROJETO DA PLACA DE CIRCUITO IMPRESSO)

Projeto da placa de circuito impresso



Lista de componentes

Componente	Valor	Quantidade
Capacitor	10nF	2
Capacitor	330nF	1
Resistor	12K	2
Resistor	200	2
Resistor	750	1
Resistor	270K	1
Conector Borne 2 pinos	-	2
Barra de pinos 1x3	-	2
Barra de pinos 1x2	-	2
H11AA1	-	1
Moc3052	-	1
HLK-PM01	-	1

Certifico que o aluno **RAFAEL DE OLIVEIRA QUINTÃO**, autor do trabalho de conclusão de curso intitulado “**DESENVOLVIMENTO DE UM CONTROLADOR ADAPTATIVO DE TEMPERATURA PARA FORNOS ELÉTRICOS A RESISTÊNCIA APLICADOS NA SOLDAGEM DE COMPONENTES SMD**”, efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

A handwritten signature in black ink, appearing to read 'J. Castro', is written over a horizontal line. The signature is stylized and cursive.

Prof. M. Sc. João Carlos Vilela de Castro - Orientador

Ouro Preto, 7 de março de 2019.