



JEAN CARLOS TERRA

**CONTROLADOR DE TEMPERATURA PARA
PROCESSADORES UTILIZANDO PASTILHA DE
PELTIER**

**LAVRAS MG
2022**

JEAN CARLOS TERRA

**CONTROLADOR DE TEMPERATURA PARA PROCESSADORES
UTILIZANDO PASTILHA DE PELTIER**

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do Curso de Engenharia de Controle e Automação, para a obtenção do título de Bacharel.

Dr. Leonardo Silveira Paiva
Orientador

**LAVRAS MG
2022**

JEAN CARLOS TERRA

**CONTROLADOR DE TEMPERATURA PARA PROCESSADORES
UTILIZANDO PASTILHA DE PELTIER**

**TEMPERATURE CONTROLLER FOR PROCESSORS USING PELTIER
PELLET.**

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do Curso de Engenharia de controle e automação, para a obtenção do título de Bacharel.

Dr. Leonardo Silveira Paiva
Orientador

**LAVRAS MG
2022**

Aos meus pais, Carlos Roberto Terra e Emelice Fernanda Nascimento.

Sem vocês, nada seria possível.

Dedico

AGRADECIMENTOS

Agradeço a Deus por me amparar em todos os momentos, por iluminar meus caminhos, escolhas e pela sua presença em minha vida.

Aos meus pais e irmão, que me apoiaram em todos os momentos, especialmente meus estudos, pois sei que abdicaram de muitas coisas para tornarem meu sonho possível, e espero retribuir a todos vocês por seus esforços e sacrifícios.

A todos os meus amigos do curso de Engenharia de Controle e Automação, em especial ao Jurandir Alves Júnior e Gabriel Henrique, pelos grandiosos momentos de apoio, ensino e diversão.

Ao meu orientador, Leonardo Silveira Paiva, pela possibilidade de realização do presente trabalho.

À Universidade Federal de Lavras (UFLA) e ao departamento de automática, pela possibilidade de realização do curso de Engenharia de Controle e Automação.

MUITO OBRIGADO!

“Não basta conquistar a sabedoria, é preciso usá-la”

(Mundo das mensagens)

RESUMO

A utilização de supercomputadores têm se tornado cada vez mais necessária para processar o grande volume de informações presentes em softwares, seja ela uma melhoria no processamento de um vídeo, a capacidade de rodar um jogo ou até mesmo em pesquisas em que há a necessidade de processar grande número de cálculos matemáticos. Seus processadores estão cada vez menores em dimensão, suas frequências de *clock* estão cada vez maiores, e fatores como estes contribuem para o aumento da temperatura. A evolução dos dissipadores de temperatura utilizados nestes componentes não têm acompanhado no mesmo ritmo. Nesse contexto, objetivou-se com o trabalho o desenvolvimento de um protótipo que possa ser eficiente no controle de temperatura da *Central Processing Unit* (CPU), não emitindo ruídos e que possa receber uma temperatura desejada de operação, visto que cada processador possui a temperatura ideal para sua melhor operação. Para o desenvolvimento do protótipo foi necessário criar uma planta para coletar dados de entrada *Pulse Width Modulation* (PWM) e saída (Temperatura), e assim fazer a modelagem matemática da mesma. Após realização desta, foi necessário definir os parâmetros de desempenho e por fim, foi desenvolvido o controlador Proporcional, Integral Derivativo (PID) que controla a planta, seguindo os valores definidos no projeto. O sistema apresentou um desempenho superior quando comparado ao protótipo sem o perfil de controle, sendo possível manter a temperatura do processador na faixa definida (*setpoint* escolhido), e resultando assim em ganhos de desempenho para o mesmo.

Palavras-chave: Controlador de Temperatura. Dissipação térmica. Peltier.

ABSTRACT

The use of super-processors has become necessary, to process infinite information in software, be it, an improvement in the functioning of a video, the processing capacity of a video, or even in research the need for math calculations. Your processors are getting smaller and smaller in size, and their clock frequencies are getting bigger and bigger, as factors that contribute to the temperature increase each time. The evolution of the heat sinks used in these components does not follow the same pace. In this context, the objective of this work was the development of a prototype that can be efficient in controlling the temperature of the Central Processing Unit (CPU), does not emit noise and can receive the desired operating temperature since each processor has one temperature ideal for your best operation. For the development of the prototype, it was necessary to create a plant to collect input Pulse Width Modulation (PWM) and output (Temperature) data, and thus do the mathematical modeling of the same. After carrying out this, it was necessary to define the performance parameters and finally, the proportional, integral derivative (PID) controller was developed that controls the plant, following the values defined in the project. The system presented an excellent performance because from its use, it was possible to keep the processor temperature in the defined range (chosen set point), thus, generating performance gains for it.

Keywords: Temperature Controller. Thermal dissipation. Peltier.

SUMÁRIO

1	INTRODUÇÃO	10
2	OBJETIVO	12
2.1	Objetivo Geral	12
2.2	Objetivos Específicos	12
3	REFERENCIAL TEÓRICO	13
3.1	Calor	13
3.2	Leis da Termodinâmica	16
3.3	Efeitos Termoelétricos	18
3.4	Placas de Efeito Peltier	20
3.5	Definição de Sistemas	21
3.6	Identificação de Sistemas	22
3.7	Modelagem em Caixa Preta	23
3.8	Modelo por Função de Transferência	24
3.9	Teoria Clássica de Controle	24
3.10	Controladores	28
3.11	Ações básicas de controle	29
3.12	Métodos de Sintonia de Controle	33
3.13	Matlab	35
3.13.1	Toolbox Ident	36
3.13.2	PID Tuner	36
4	MATERIAL E MÉTODOS	38
4.1	Materiais utilizados	38
4.2	Montagem da planta para aquisição de dados	40
4.3	Identificação Do sistema	41
4.4	Definição dos parâmetros de desempenho	45
4.5	Desenvolvimento do controlador utilizando o PID Tuner	48
4.6	Simulações do controlador PID no Simulink	52
4.6.1	Simulações sem adição de ruídos	52
4.6.2	Simulações com adição de ruídos	54
4.6.3	Análise das simulações	57
4.7	Implementação do protótipo	57
4.7.1	Montagem do Hardware	57
4.7.2	Desenvolvimento do Software	58
5	RESULTADOS E DISCUSSÃO	59
5.1	Sistema com protótipo e processador nas condições nominais	59
5.2	Sistema sem protótipo e processador em condições de estresse	59

5.3 Sistema com protótipo e processador em condições de estresse	61
6 CONCLUSÃO	64
7 TRABALHOS FUTUROS	64
REFERÊNCIAS	65

1 INTRODUÇÃO

O processador é um dos principais componentes de um dispositivo eletrônico inteligente, seja ele um computador pessoal, smartphone ou até mesmo um *smartwatch*. Ele é o responsável por processar praticamente todos os comandos e operações, atuando no controle da memória *Random Access Memory* (RAM), abertura e execução de *softwares*, se comunicando com os demais componentes por meio da placa-mãe e seus vários barramentos. (MUNDO POSITIVO, 2021).

O processador aquece durante sua utilização. Se este calor for excessivo, pode ocorrer diversas manifestações de instabilidade no sistema, variando desde travamentos ocasionais a casos mais graves, como a queima do componente. O processador, por estar em uso contínuo, deve ser constantemente refrigerado, caso contrário atingirá temperaturas muito elevadas, podendo chegar facilmente a 100 [°C] (ADRIEL, 2009).

O dissipador de calor ativo é o mais utilizado, visto que ele possui um valor relativamente baixo e a função de resfriamento de maneira adequada quando o processador opera em suas condições nominais. Existem inúmeras opções disponíveis para manter a temperatura do processador nos limites estáveis de operação, cada uma oferecendo sua combinação de preço, nível de ruído e desempenho. Um dissipador de calor passivo consiste em um bloco de metal que deve ser mantido em contato com o processador e possui aletas finas. Tais dissipadores retiram calor de um processador, aumentando a quantidade de área superficial disponível para o calor se dissipar no ar. Os dissipadores passivos como solução de resfriamento não são mais utilizados devido a sua ineficiência em resfriar os processadores atuais, visto que os mesmos são cada vez menores em dimensão e este modelo de dissipador é mais eficiente quando há uma maior área de contato para realizar a troca de calor (BRITNEY, 2022).

O dissipador de calor ativo é a combinação do dissipador de calor passivo com um *cooler* acoplado e por esta razão possui maior capacidade de dissipação térmica quando comparado ao dissipador passivo, pois é capaz de afastar o ar quente do processador e mover o ar frio pelo dissipador, tornando-o mais eficiente.

Diferente dos *coolers* tradicionais, os *Water Coolers* (WC) oferecem uma proposta única de resfriamento. Por meio de um sistema de bombeamento de água, os WC são capazes de dissipar o calor do processador. Assim, uma bomba garante que o líquido fique em movimento, enquanto o radiador é responsável por manter o líquido resfriado (SALUTES, 2021).

Ao comparar os três principais modelos de resfriadores utilizados com a finalidade de proteger os processadores de sobretensões, é evidente que os WC se sobressaem em relação aos dissipadores mais comuns, contudo estes modelos possuem desvantagens, e por se tratar de uma tecnologia recente eles possuem custos mais elevados, demandam grande espaço para instalação além da geração de ruídos no ambiente. Os WCs ainda não são acessíveis financeiramente ou podem não ser adequados ao ambiente que serão instalados.

Analisando as opções oferecidas pelo mercado e comparando as características de cada modelo, o presente trabalho surge com a proposta de criar um dissipador utilizando a pastilha de Peltier, eficiente na troca de calor, não gerando ruídos sonoros, acessível e com a opção da escolha, pelo usuário, da temperatura a qual ele deseja manter seu processador.

2 OBJETIVO

2.1 Objetivo Geral

O objetivo deste trabalho é o desenvolvimento de um protótipo que permitirá o controle da temperatura de processadores computacionais. O controlador de temperatura será concebido seguindo a estrutura de Sistemas em Malha Fechada, para possibilitar ao usuário a opção de definir a temperatura a qual ele deseja para o processador.

Os objetivos do trabalho são interligados, pois a partir do momento em que o usuário pode definir a temperatura para seu processador, o mesmo poderá ser utilizado de maneira mais segura quando submetido a condições mais severas, sem o risco de queimar o componente por superaquecimento.

Por fim, é objetivado um protótipo eficiente e de baixo custo, para poder ser acessível e adaptável no controle de temperatura de processadores computacionais.

2.2 Objetivos Específicos

O presente trabalho apresenta os seguintes objetivos específicos:

- Criar um protótipo no qual utilizará um PWM para controlar a tensão aplicada na pastilha de Peltier, que estará fixada no processador juntamente com um sensor de temperatura. Coletar os dados de entrada (PWM) e saída (Temperatura do Processador) para a identificação do Sistema;
- Identificação do Sistema em Caixa Preta, utilizando a ferramenta Toolbox *Ident* do *software* Matlab;
- Definir os Parâmetros de desempenho, a serem utilizados no Controlador PID;
- Projetar o Controlador PID para cumprir os requisitos desejados;
- Simular o sistema com o Controlador PID;
- Desenvolver o protótipo;
- Apresentar os resultados alcançados pelo protótipo;

3 REFERENCIAL TEÓRICO

3.1 Calor

A física considera o calor como sendo a energia que passa de um corpo para outro ou de um sistema para outro, uma transferência associada ao movimento de átomos, moléculas e outras partículas. Neste sentido, o calor pode ser produzido a partir de reações nucleares (como as fusões que ocorrem no interior do Sol), reações químicas (a combustão) ou dissipações. Neste último caso, pode-se falar de dissipação eletromagnética (os fornos a microondas) ou dissipação mecânica (a fricção) (EQUIPE, 2021).

A equação fundamental da calorimetria define a energia necessária para um corpo variar da temperatura inicial T_i para a temperatura final T_f , ou seja, ΔT . Essa definição é representada matematicamente na equação 1.

$$Q = m * c * \Delta T \quad (1)$$

Onde:

Q = Calor [KJ]

m = Massa [kg]

c = Calor específico [KJ/kg·K]

ΔT = Variação da temperatura [K]

A transferência de calor pode ser definida como a transferência de energia de uma região para outra como resultado de uma diferença de temperatura entre elas. Os mecanismos de transferência de calor são: condução, convecção e radiação, como mostrado na Figura 1.

Figura 1 - Mecanismos de transferência de calor.



Fonte: Rafael Helerbrock (2019)

A condução pode ser definida como o processo pelo qual a energia é transferida de uma região de alta temperatura para outra de temperatura mais baixa dentro de um meio (sólido, líquido ou gasoso) ou entre meios diferentes em contato direto. Este mecanismo pode ser visualizado como a transferência de energia das partículas mais energéticas para partículas menos energéticas de uma substância devido a interações entre elas (GASPAR, 2012).

O fluxo de calor da condução térmica é definido pela equação 2 e é diretamente proporcional à condutividade térmica, K_t . A condutividade térmica é a potência térmica por metro devido a uma diferença de temperatura de 1 Kelvin.

$$\phi = - \frac{K_t \cdot A \cdot \Delta T}{\Delta L} \quad (2)$$

Onde:

ϕ = Fluxo de calor por condução [W]

K_t = Condutividade térmica [W/m·K]

A = Área da seção transversal [m²]

ΔT = Variação da temperatura [K]

ΔL = Comprimento do trajeto do calor [m]

A transferência de calor por condução ocorre prioritariamente em meios sólidos. Para os líquidos o processo é basicamente o mesmo, embora as moléculas estejam menos espaçadas e as interações sejam mais fortes e mais frequentes.

A convecção pode ser definida como o processo pelo qual a energia é transferida das porções quentes para as porções frias de um fluido através da ação combinada de: condução de calor, armazenamento de energia e movimento de mistura. Esse tipo de transferência de energia ocorre prioritariamente em líquidos ou gases.

A convecção se dá pelo deslocamento de massas de fluido por diferença de densidades. Quando uma porção de fluido é aquecido, seus átomos e moléculas começam a se mover mais rapidamente, fazendo com que eles ocupem um espaço intermolecular maior. O fluido se expande e aumenta o seu volume por unidade de massa (diminui a densidade). Logo uma porção aquecida é “mais leve” e tende a subir, sendo substituída por uma quantidade de volume com menor temperatura e maior densidade. Esse processo se repete inúmeras vezes e daí surgem as correntes de convecção que mantêm o fluido em circulação. (GASPAR, 2012).

O fluxo de calor da convecção térmica é definido pela equação 3 e é independente do tipo da convecção.

$$\phi = h * A * \Delta T \quad (3)$$

Onde:

ϕ = Fluxo de calor por convecção [W]

h = Coeficiente de calor por convecção [W/m²·K]

A = Área de transferência de calor [m²]

ΔT = Variação de temperatura [K]

A radiação pode ser definida como o processo pelo qual calor é transferido de uma superfície em alta temperatura para uma superfície em temperatura mais baixa, quando tais superfícies estão separadas no espaço, ainda que exista vácuo entre elas. A energia assim transferida é chamada de radiação térmica feita sob a forma de ondas eletromagnéticas.

As emissões de ondas eletromagnéticas podem ser atribuídas a variações das configurações eletrônicas dos constituintes de átomos e moléculas, e ocorrem devido a vários fenômenos, porém, para a transferência de calor interessa apenas as ondas eletromagnéticas resultantes de uma diferença de temperatura (radiações térmicas).

Todos corpos em temperatura acima do zero absoluto emitem continuamente radiação térmica. As intensidades das emissões dependem somente da temperatura e da natureza da superfície emissora. A radiação térmica viaja na velocidade da luz (300.000 km/s).

A equação 4 define o fluxo de transferência de calor por radiação, dependente da emissividade (ϵ).

$$\phi = - \epsilon * \sigma * A (T_{sup}^4 - T_{viz}^4) \quad (4)$$

Onde:

ϕ = Fluxo de calor por irradiação [W]

ϵ = Emissividade

σ = Constante de Stefan-Boltzmann [W/(m² * K⁴)]

A = Área da seção transversal [m²]

T_{sup} = Temperatura da superfície [K]

T_{viz} = Temperatura da vizinhança [K]

3.2 Leis da Termodinâmica

Termodinâmica, de acordo com Atkins (2006) é o estudo das relações entre trabalho, energia térmica e calor através da observação da pressão, do volume e da temperatura, que são variáveis mensuráveis.

A primeira lei da termodinâmica, descoberta por Rudolf Clausius em 1850, e conhecida como o princípio da conservação de energia, a soma da energia mecânica e da energia térmica de um sistema isolado é constante. A variação de energia interna é igual à troca de calor e trabalho do sistema com o ambiente ou do sistema com outro sistema. De acordo com Goupil et al (2011) essa definição é representada matematicamente na equação 5.

$$\Delta U = Q - W \quad (5)$$

Onde:

ΔU = Variação de energia interna de um sistema.

Q = Transferência de calor resultante para dentro e para fora do sistema.

W = Trabalho resultante realizado no sistema.

A segunda lei da termodinâmica enunciada por Kelvin e Planck diz que não existe um dispositivo que trabalhe em um ciclo termodinâmico capaz de transformar todo o calor absorvido de uma fonte, sem que se altere a temperatura, em trabalho. A segunda lei define que é impossível uma máquina apenas tomar calor de um sistema e produzir trabalho; alguma energia é perdida no processo; isso se dá pelo princípio da irreversibilidade. O princípio da irreversibilidade é causado pelo atrito mecânico, pelo efeito Joule e outros processos que causam dissipação de energia. De acordo com Goupil et al (2011) essa definição é representada matematicamente na equação 6.

$$\eta T = \frac{Q_h - Q_l}{Q_h} \quad (6)$$

Onde:

ηT = Rendimento térmico.

Q_h = Calor fornecido por aquecimento.

Q_l = Calor não convertido em calor.

É impossível construir um dispositivo que opere segundo um ciclo e que não produza outros efeitos, além da transferência de calor de um corpo quente para um corpo frio (VAN WYLEN, 1998).

A terceira lei da termodinâmica, feita por Walther Nernst, define que um sistema quando chega à temperatura do zero absoluto, temperatura igual a 0 K, tem sua entropia igual a zero e todos os processos cessam. De acordo com Goupil et al (2011), a terceira lei é descrita pela equação 7.

$$\lim_{T \rightarrow 0} \Delta S = 0 \quad (7)$$

Onde:

ΔS = Variação de entropia.

T = Temperatura.

É impossível através de qualquer procedimento, não importa o quão idealizado, reduzir a temperatura de qualquer sistema à temperatura zero em um finito número de finitas operações (Guggenheim, 1976).

3.3 Efeitos Termoelétricos

Efeitos termoelétricos são fenômenos que associam fluxo de calor com corrente elétrica (calor diretamente convertido em eletricidade) (CALLEN, 1985); esses efeitos podem ser aplicados na construção de conversores de energia usando-se uma classe de materiais conhecidos como materiais termoelétricos (MINNICH et al., 2009).

Existem três tipos de efeitos termoelétricos: efeito Seebeck, efeito Peltier e efeito Thomson.

O efeito Seebeck, na sua essência, refere-se à conversão de um fluxo de calor em um fluxo de elétrons. Este fenômeno físico só é perceptível quando se junta os extremos de dois materiais dissimilares (“materiais diferentes”, condutores ou semicondutores) formando uma configuração conhecida como “par termoelétrico” ou “termopar” (PROJETO, 2022).

De acordo com Goupil et al (2011) a diferença de potencial gerada pelo efeito Seebeck é dada pela equação 8. A polaridade da tensão depende de qual junção é a fonte de calor e qual é a fonte fria. No semicondutor tipo-n a corrente elétrica flui do lado frio para o lado quente. No semicondutor tipo-p, a corrente elétrica flui do lado quente para o lado frio.

$$V = \alpha * \Delta T \quad (8)$$

Onde:

V = Diferença de potencial [V]

α = Coeficiente Seebeck [V/K]

ΔT = Diferença de temperatura das juntas [K]

Em 1834, Peltier descobriu o efeito inverso ao efeito Seebeck, ou seja, uma corrente elétrica direta passa em um circuito formado por semicondutores diferentes, o que fez surgir em suas junções uma diferença de temperatura (SANTOS, 2016).

A aplicação de uma tensão elétrica gera uma corrente elétrica. De acordo com Goupil et al (2011), a equação 9 define a taxa de transferência de calor em função do fluxo de corrente elétrica.

$$Q = - \pi * I \quad (9)$$

Onde:

Q = Taxa de transferência de calor pelo efeito Peltier [W]

π = Coeficiente de Peltier do termopar [V]

I = Corrente elétrica [A]

Para obter-se um efeito contrário ao efeito Peltier, basta o aquecimento ou resfriamento externo das junções. Quando uma junção é mantida à temperatura constante e a outra aquecida por uma fonte de energia externa, ainda obtém-se uma corrente elétrica em uma direção, sem que estejam ocorrendo outros efeitos termoelétricos.

Willian Thomson descobriu a relação entre o efeito Seebeck e o efeito Peltier, onde os coeficientes dos dois efeitos estão relacionados através da termodinâmica. No efeito Thomson, a quantidade de transferência de calor em um condutor tem uma relação de proporcionalidade da transferência de calor com a diferença de temperatura dos terminais e a corrente elétrica (FERNANDE, 2016).

Segundo o efeito Thomson, quando uma corrente elétrica passa por um condutor homogêneo, este tendo um gradiente ou diferença de temperatura nas extremidades, o condutor absorve ou emite calor. Dependendo dos sentidos da corrente elétrica e do gradiente de temperatura.

De acordo com Goupil et al (2011) a equação 10 define a taxa de transferência de calor pelo efeito Thomson. A taxa de transferência de calor total é uma parcela devido ao efeito Thomson e uma parcela devido ao efeito Joule.

$$QT = u * I * \Delta T \quad (10)$$

Onde:

QT = Taxa de transferência de calor pelo efeito Thomson [W];

u = Coeficiente de Thomson [V/K];

I = Corrente elétrica [A];

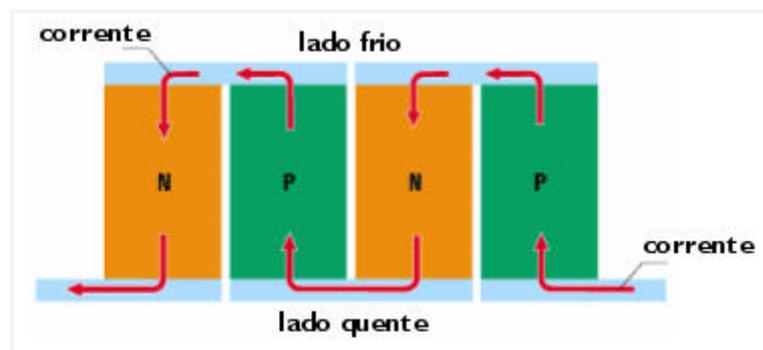
ΔT = Diferença de temperatura [K]

3.4 Placas de Efeito Peltier

Módulos Peltier, também conhecidos como pastilhas termoelétricas, são pequenas unidades de que utilizam tecnologia de matéria condensada para operarem como bombas de calor. Esses módulos são essencialmente junções de placas cerâmicas com pequenos cubos de telureto de bismuto. Sua operação é baseada no Efeito Peltier. Quando uma corrente é aplicada, o calor move-se de um lado ao outro, onde ele deve ser removido com um dissipador. Se os pólos elétricos forem revertidos, a pastilha se tornará um excelente aquecedor. É importante salientar que os módulos não “consomem” calor, por isso que se torna necessário o uso do dissipador (WELTY et al., 2008).

Pastilhas termoelétricas operam utilizando o efeito Peltier. A tensão elétrica aplicada aos pólos de dois materiais distintos cria uma diferença de temperatura. Graças a essa diferença, o resfriamento Peltier fará o calor mover-se de um lado ao outro. Uma típica pastilha de Peltier conterà uma série de elementos semicondutores do tipo-p e tipo-n, agrupados como pares (ver Figura 2), que agirão como condutores dissimilares (DANVIC, 2016).

Figura 2 - Elementos de formação da pastilha.

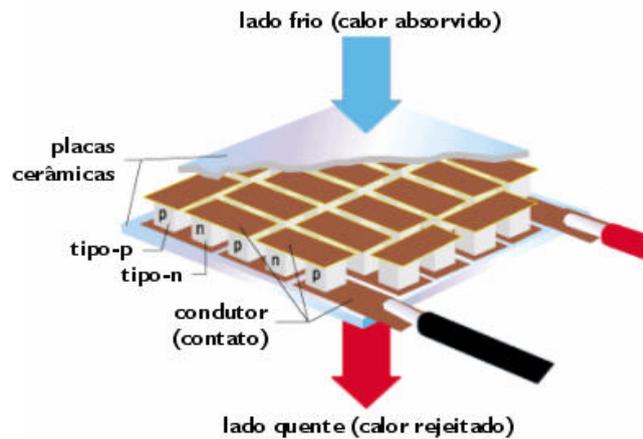


Fonte: Mecatrônica Hoje (2022)

Essa série de elementos é soldada entre duas placas cerâmicas, eletricamente em série e termicamente em paralelo. Quando uma corrente *Direct Current* (DC) passa por um ou mais pares de elementos de tipo-n e tipo-p, há uma redução na temperatura da junta do lado frio,

resultando em uma absorção do calor do ambiente. Este calor é transferido pela pastilha por transporte de elétrons emitido no lado quente, via elétrons que movem de um estado alto para um estado baixo (ver Figura 3). A capacidade de bombeamento de calor de um resfriador é proporcional à corrente e ao número de pares de elementos tipo-n e tipo-p (DANVIC, 2016).

Figura 3 - Operação da pastilha de Peltier.



Fonte: Mecatrônica Hoje (2022)

A grande vantagem de pastilhas do tipo Peltier são a ausência de peças móveis, gás freon, barulho e vibração; além do tamanho reduzido, alta durabilidade e precisão. Elas são utilizadas hoje em inúmeros setores, principalmente os de bens de consumo, automotivo, industrial e militar (DANVIC, 2016).

3.5 Definição de Sistemas

De acordo com Mohapatra et al (1994), as principais características de um sistema são:

- Um sistema é um agrupamento complexo de humanos e de máquinas;
- Um sistema pode estar formado de subsistemas, a quantidade de detalhes dos subsistemas depende do problema que está sendo estudado. Os diagramas de fluxo dão a descrição de um caminho para o real entendimento desses subsistemas;
- As saídas de um dado subsistema proporcionam a entrada de outros subsistemas. Assim, um subsistema interage com outro subsistema; portanto, não podem ser estudados isoladamente;

- O sistema que está sendo estudado, usualmente, formará parte de uma hierarquia de tais sistemas. O sistema superior é muito importante e exerce considerável influência no sistema abaixo dele;
- Para funcionar, o sistema deve ter um objetivo, mas este objetivo é também influenciado pelos demais sistemas do qual ele forma parte. Normalmente, os sistemas possuem múltiplos objetivos que estão em conflito um com o outro; assim, é requerido um objetivo geral que afete os compromissos entre esses objetivos conflitantes;
- Para funcionar com a máxima eficiência, um sistema deve ser projetado de tal forma que seja capaz de alcançar seu objetivo geral da melhor forma possível

3.6 Identificação de Sistemas

Na literatura de identificação de sistemas há duas abordagens para a modelagem matemática de sistemas dinâmicos: a modelagem fundamentada pela física do processo; e a modelagem matemática via teoria de Identificação de Sistemas. A modelagem fundamentada na física do processo, também conhecida como modelagem caixa-branca, necessita do conhecimento dos fenômenos físicos e químicos envolvidos, pois ocorre através das leis e princípios que representam estes fenômenos. A utilização desta abordagem permite derivar modelos que descrevem a dinâmica interna do sistema, além da relação entrada-saída. Assim os parâmetros do modelo são oriundos do sentido físico aos quais pertencem, sendo essa sua principal vantagem. Sua maior desvantagem está no fato de que a maioria dos sistemas a serem modelados são grandes e complexos, podendo implicar em equações desconhecidas, análises complexas e demoradas (AGUIRRE, 2007).

A maneira de superar a desvantagem apresentada acima é a construção de modelos matemáticos de sistemas dinâmicos a partir de dados obtidos de um sistema real, ou planta experimental. Estes modelos representam uma determinada característica do sistema em estudo. Há duas formas para a construção de modelos matemáticos: modelagem caixa-preta, na qual não se tem conhecimento prévio do sistema a ser modelado, neste caso apenas os dados de entrada e saída do processo são usados durante a identificação, é a forma escolhida para a identificação do sistema neste trabalho. A grande vantagem desta técnica está na obtenção do modelo e da dinâmica de resposta. A modelagem caixa-cinza, considera tanto as variáveis de entrada e a saída do sistema, como a física do sistema e gera um modelo parcial baseado em ambas as técnicas (AGUIRRE, 2007).

3.7 Modelagem em Caixa Preta

A modelagem matemática utiliza, então, os objetos-modelo e os modelos matemáticos, que podem ser baseados em teoremas, em leis e em princípios que regem um determinado sistema. Portanto, essa é a área do conhecimento que estuda maneiras de desenvolver e de implementar modelos matemáticos de sistemas reais (AGUIRRE, 2007).

Em teoria de sistemas, a denominação “caixa preta” se refere aos métodos de modelagem onde a única fonte de informação a partir da qual um modelo é construído é a relação causa-efeito apresentada pelos sinais de entrada e saída do sistema modelado (AGUIRRE, 2007). A modelagem em caixa preta possibilita a escolha de modelos de menor complexidade para representar um sistema, permitindo a aplicação de algoritmos de identificação de sistemas quase que diretamente. Como não há preocupação com o funcionamento interno do sistema, nesses modelos os parâmetros envolvidos na modelagem não têm relação direta com as características do sistema real.

A modelagem caixa-preta, tem como vantagem a flexibilidade dos modelos e possibilidade de ajustar-se aos dados, mas não apresenta relação conceitual com o fenômeno estudado. A modelagem caixa-preta surgiu como alternativa às dificuldades enfrentadas ao realizar a modelagem caixa-branca, já que essa última exige bastante conhecimento sobre o sistema, apesar de o modelo resultante ser muito útil para a compreensão e análise do processo. O processo de modelagem em Caixa Preta é apresentado na Figura 4.

Figura 4 - Modelagem em caixa preta.



Fonte: Wikipedia (2022)

3.8 Modelo por Função de Transferência

Uma função de transferência (FT) relaciona duas variáveis em um processo físico: uma delas é a causa, função perturbação ou variável de entrada, e a outra é o efeito, resposta ou variável de saída (COUGHANOWR; KOPPEL, 1986). Esta é definida como a relação entre a transformada de Laplace da saída pela transformada de Laplace da entrada, admitindo condições iniciais nulas.

Denotando a transformada de Laplace da entrada e da saída respectivamente por $Y(s)$ e $U(s)$ temos que a função de transferência $G(s)$ é dada pela equação 11.

$$G(s) = \frac{L(y(t))}{L(u(t))} = \frac{Y(s)}{U(s)} \quad (11)$$

Onde $Y(s)$ e $U(s)$ são polinômios em s . A partir desta representação e supondo que as raízes de $Y(s)$ são todas diferentes das raízes de $U(s)$, define-se o seguinte:

- Pólos de $G(s)$: raízes de $U(s)$.
- Zeros de $G(s)$: raízes de $Y(s)$.
- Ordem do sistema: grau de $U(s)$.
- Tipo do sistema: número de pólos da $G(s)$ em $s=0$.

Se a função de transferência de um sistema é conhecida, pode-se estudar o comportamento do sistema para diferentes funções de entrada.

3.9 Teoria Clássica de Controle

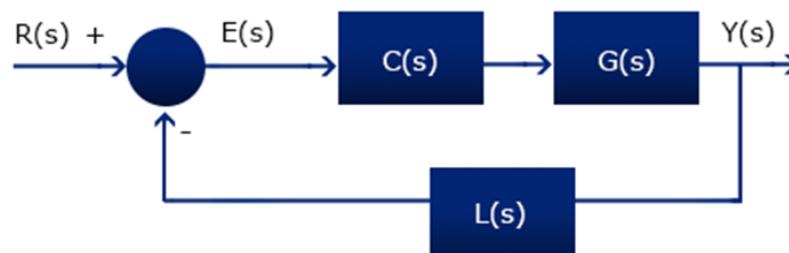
Um sistema de controle é definido como uma interligação de componentes formando uma configuração de sistema que fornecerá uma saída de sistema desejada. Como a resposta desejada de um sistema é conhecida, é gerado um sinal proporcional ao erro entre a resposta desejada e a resposta real (DORF, 2001).

O controle em malha aberta consiste em aplicar um sinal de controle pré-determinado, esperando-se que ao final de um determinado tempo, a variável controlada atinja um determinado valor, ou apresente um determinado comportamento. Neste tipo de sistema de controle não são utilizadas informações sobre evolução do processo para determinar o sinal de controle a ser aplicado em um determinado instante. Mais especificamente, o sinal de controle

não é calculado a partir de uma medição do sinal de saída (AGUIRRE, 2007).

No controle em malha fechada, informações sobre como a saída de controle está evoluindo são utilizadas para determinar o sinal de controle que deve ser aplicado ao processo em um instante específico. Isto é feito a partir de uma realimentação da saída para a entrada. Em geral, a fim de tornar o sistema mais preciso e de fazer com que ele reaja a perturbações externas, o sinal de saída é comparado com um sinal de referência (*setpoint*) e o desvio (erro) entre estes dois sinais é utilizado para determinar o sinal de controle que deve efetivamente ser aplicado ao processo. Assim, o sinal de controle é determinado para corrigir este desvio entre a saída e o sinal de referência. O dispositivo que utiliza o sinal de erro para determinar ou calcular o sinal de controle a ser aplicado à planta é chamado de controlador, ou compensador (AGUIRRE, 2007). O diagrama básico de um sistema de controle em malha-fechada é mostrado na Figura 5.

Figura 5 - Diagrama básico de um sistema em malha-fechada.

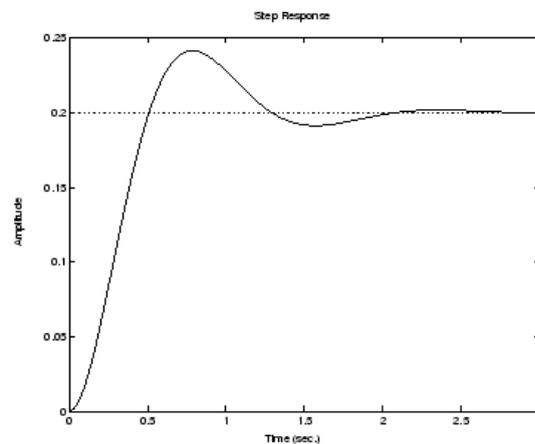


Fonte: Maxwell (2022)

A estabilidade de um sistema pode ser definida de diversas maneiras e segundo vários pontos de vista. Segundo o conceito *Bounded Input-Bounded Output* (BIBO) um sistema é dito ser estável se, para todo sinal de amplitude limitada aplicado em sua entrada, o sinal de saída é também limitado. Por outro lado, se o sistema é instável, ao aplicarmos um sinal de amplitude limitada em sua entrada, sua saída divergirá com o passar do tempo, ou seja, a amplitude do sinal de saída tenderá a crescer indefinidamente (DORF, 2001).

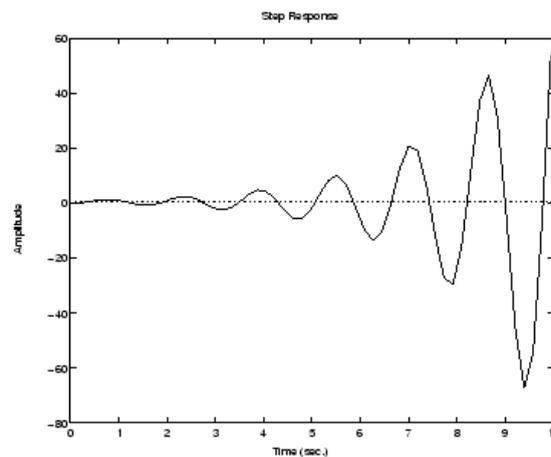
As Figuras 6 e 7 mostram a resposta de dois sistemas a uma entrada do tipo degrau unitário, sinal de amplitude limitada. Note que a resposta do sistema da Figura 6 é limitada em amplitude, portanto, este sistema é estável. Já o sistema da Figura 7 apresenta um comportamento divergente, sendo, portanto, instável.

Figura 6 - Exemplo de sistema estável.



Fonte: João Manoel (2000)

Figura 7 - Exemplo de sistema instável.



Fonte: João Manoel (2000)

A BIBO estabilidade é determinada pelos pólos da função de transferência em sistemas de malha aberta $G(s)$ e para sistemas em malha fechada pelos zeros da equação característica $(1+G(s))$ que são equivalentes aos pólos (AGUIRRE, 2007).

Conforme o posicionamento dos pólos no semi-plano esquerdo (SPE), semi-plano direito (SPD) ou eixo imaginário temos os seguintes comportamentos:

- Estabilidade: todos os pólos do sistema devem estar no SPE.
- Marginalmente Estável: se o sistema possui ao menos 1 par de pólos complexos conjugados sobre o eixo imaginário. Neste caso, a resposta em regime permanente do sistema a uma entrada de amplitude limitada será uma senóide.
- Instabilidade: se o sistema possuir ao menos 1 pólo no SPD.

O desempenho em regime transitório de um sistema é avaliado, em geral, pela resposta temporal do sistema a uma entrada do tipo degrau unitário. O desempenho do sistema é medido pelo valor das seguintes grandezas:

- *Overshoot*: supondo que o valor da saída ultrapasse o valor da referência, o máximo sobrepasso é definido como a máxima diferença entre a saída e a entrada durante o período transitório, ou seja, o valor de pico máximo atingido pela resposta. Se a saída não ultrapassa o valor da entrada, o sobrepasso máximo igual a zero.
- Tempo de subida: É definido como o tempo transcorrido para a resposta ir de 10% a 90% do seu valor final. O tempo de subida é um indicativo de quão rápido o sistema responde à aplicação de uma entrada degrau. Muitas vezes a redução excessiva do tempo de subida de um sistema a partir da sintonia dos parâmetros de um controlador pode provocar o aparecimento de um alto *overshoot*.
- Tempo de acomodação: é o tempo necessário para que a resposta entre e permaneça dentro de uma faixa percentual de 2% em torno do valor de regime permanente.

O desempenho em regime permanente é medido a partir da capacidade de um sistema seguir referências padrões (salto, rampa, parábola) e a rejeitar assintoticamente sinais de perturbação. A ferramenta matemática utilizada na análise do regime permanente é o Teorema do Valor Final dado pela equação 12.

$$y(\infty) = \lim_{s \rightarrow 0} s * u(s) * T(s) \quad (12)$$

Onde:

$y(s)$ = Valor em regime permanente.

$u(s)$ = Entrada aplicada ao sistema.

$T(s)$ = Função de transferência entre y e u .

Em sistemas de controle em malha fechada, é desejável que a saída do sistema em regime permanente seja igualada ao sinal de referência (*setpoint*). Entretanto, em alguns casos esta igualdade não é atingida, ocorrendo o erro em regime permanente, dado pela equação 13.

$$e(\infty) = u(\infty) - y(\infty) \quad (13)$$

Onde:

$e(\infty)$ = Erro em regime permanente

$u(\infty)$ = *Setpoint*

$y(\infty)$ = Valor em regime permanente

3.10 Controladores

Os controladores analógicos operam com sinais contínuos no tempo. A variável de processo é medida através de um sensor que converte a grandeza física em um sinal elétrico, pneumático ou mecânico, este sinal é comparado com uma referência de mesma grandeza. A diferença entre os dois sinais é então aplicado a um controlador que nada mais é que um sistema elétrico, pneumático ou mecânico que irá gerar um sinal contínuo a ser aplicado na entrada da planta a ser controlada. Um controlador analógico pode ser visto também como um filtro analógico. Sua implementação se dá então a partir de uma função de transferência. Assim, por exemplo, este filtro pode ser realizado sob a forma de um circuito eletrônico cuja característica entrada-saída corresponde à função de transferência que o controlador deve apresentar a fim de satisfazer certas especificações de controle (MAIA, 1993).

Um controlador digital trabalha com sinais numéricos digitais. Um controlador digital é fisicamente implementado como uma rotina ou programa a ser executado sobre um microprocessador, ou microcontrolador. O controle digital de um processo envolve então o que chamamos de processo de amostragem. O sinal de saída (ou de erro) é amostrado periodicamente com um período T . O sinal amostrado (analógico) passa então por um conversor analógico/digital (A/D) onde é quantizado e transformado em um sinal numérico (palavra de “ n ” *bits*). Este sinal digital é lido por um microcontrolador que vai então realizar operações numéricas com este sinal e gerar outra palavra de “ n ” *bits* correspondente à ação de controle que deverá ser aplicada sobre a planta no próximo instante de amostragem. Este sinal numérico é então convertido novamente em um sinal analógico por um conversor digital-analógico (D/A) que disponibilizará, no próximo *clock* de amostragem, um sinal constante de tensão. Desta forma, entre dois instantes de amostragem, o sinal efetivamente aplicado pela planta é um sinal contínuo de amplitude fixa (MAIA, 1993).

As vantagens da utilização de um controlador digital são:

- Robustez para implementar leis de controle.

- Facilidade para mudar as leis de controle .
- Menor sensibilidade a ruídos.
- Menor sensibilidade a desgaste de componentes e mudanças de ambiente.
- Custo cada vez mais baixo.

As desvantagens são:

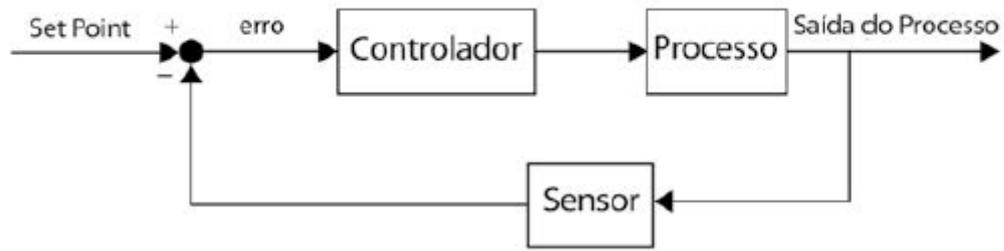
- Banda limitada (taxa de amostragem).
- Resolução (comprimento da palavra).
- Velocidade de computação.

3.11 Ações básicas de controle

O controlador pode ser entendido como um dispositivo que realiza determinadas operações matemáticas sobre o sinal de erro $e(t)$ a fim de produzir um sinal $u(t)$ a ser aplicado à planta com o intuito de satisfazer um determinado objetivo. Estas operações matemáticas constituem o que chamamos de ações de controle (MAIA, 1993).

No controle em malha fechada, informações sobre como a saída de controle está evoluindo são utilizadas para determinar o sinal de controle que deve ser aplicado ao processo em um instante específico. Isto é feito a partir de uma realimentação da saída para a entrada. Em geral, a fim de tornar o sistema mais preciso e de fazer com que ele reaja a perturbações externas, o sinal de saída é comparado com um sinal de referência (*setpoint*) e o desvio (erro) entre estes dois sinais é utilizado para determinar o sinal de controle que deve efetivamente ser aplicado ao processo. Assim, o sinal de controle é determinado para corrigir este desvio entre a saída e o sinal de referência. O dispositivo que utiliza o sinal de erro para determinar ou calcular o sinal de controle a ser aplicado à planta é chamado de controlador, ou compensador. (BOLTON, 1995). O diagrama básico de um sistema de controle em malha-fechada é mostrado na Figura 8.

Figura 8 - Diagrama de um controlador em malha fechada.



Fonte: Instrumentação e Controle (2022)

As ações básicas de controle são divididas em 4 modelos de controle.

Ação liga-desliga (*on-off*): neste tipo de ação o sinal de controle pode assumir apenas dois valores, dependendo do valor do erro. Erro positivo implica em um valor “x” e erro negativo valor “y” para o sinal de controle. Este tipo de função pode ser entendida como um simples comparador. A ação de controle liga-desliga pode assim ser considerada a ação de controle mais simples e mais econômica. Entretanto, este tipo de ação possui limitações no que diz respeito ao comportamento dinâmico e em regime permanente do sistema em malha fechada.

Ação proporcional: neste tipo de ação o sinal de controle aplicado a cada instante à planta é proporcional à amplitude do valor do sinal de erro, como mostra a equação 14:

$$U = Kp * E \quad (14)$$

Onde:

U = Ação proporcional

Kp = Ganho proporcional

E = Erro

Dessa forma, se em um dado instante, o valor da saída do processo for menor ou maior que o valor da referência, o controle a ser aplicado será positivo ou negativo e proporcional ao módulo de $e(t)$. Quanto maior o ganho Kp , menor será o erro em regime permanente, isto é, melhor a precisão do sistema em malha fechada, entretanto nunca será possível anular completamente o erro. Por outro lado, quanto maior o ganho, mais oscilatório tende a ficar o comportamento transitório do sistema em malha fechada. Na maioria dos processos físicos, o aumento excessivo do ganho proporcional pode levar o sistema a instabilidade (NISE, 2017).

A ação de controle integral consiste em aplicar um sinal de controle proporcional à

integral do erro, como mostra a equação 15.

$$U = Ki * \int_0^t E dt \quad (15)$$

Onde:

U = Ação integrativa

Ki = Ganho integral

E = Erro

A ação integral possui uma função “armazenadora de energia”, pois se em um determinado tempo o erro é igual a zero, o sinal de controle será mantido em um valor constante proporcional a “energia armazenada” até o instante t . Este fato permitirá, no sistema em malha fechada, obter-se o seguimento de uma referência com erro nulo em regime permanente, pois a ação integral garantirá a aplicação ao processo de um sinal de controle constante. A ação integral está diretamente ligada à melhoria da precisão do sistema. Sob um ponto de vista matemático, a ação integral permite aumentar o tipo do sistema adicionando um pólo à origem sistema, entretanto, a introdução do mesmo tende a piorar a estabilidade relativa do sistema em malha fechada ou mesmo torná-lo instável. Por este motivo, esta ação de controle em geral não é aplicada de maneira isolada (NISE, 2017).

A ação derivativa corresponde a aplicação de um sinal de controle proporcional à derivada do sinal de erro, como mostra a equação 16.

$$U = Kd * \frac{dE}{dt} \quad (16)$$

Onde:

U = Ação derivativa

Kd = Ganho derivativo

E = Erro

A ação derivativa fornece ao sistema uma ação antecipativa, evitando previamente que o desvio se torne maior quando o processo se caracteriza por ter uma correção lenta comparada à velocidade do desvio. A ação derivativa tem sua resposta proporcional a taxa de variação da variável do processo, aumentando a velocidade de resposta do sistema caso a

presença do erro seja detectada. Logo, em sistemas de resposta lenta, como controle de temperatura, a ação derivativa permite antecipar o aumento do erro e aumentar a velocidade de resposta do sistema. Quando o sistema a ser controlado possui maior velocidade de resposta, como exemplo o controle de rotação de motores e controle de vazão de fluidos, a ação derivativa pode ser desativada, pois não há necessidade de antecipar a resposta ao erro, pois o sistema pode corrigir rapidamente seu valor. Para desativar a ação derivativa basta tornar seu valor igual a zero (AGUIRRE, 2007).

O controle Proporcional Integral Derivativo (PID) combina as vantagens do controlador PI e PD. A ação integral está diretamente ligada à precisão do sistema, sendo responsável pelo erro nulo em regime permanente. O efeito desestabilizador do controlador PI é contrabalançado pela ação derivativa que tende a aumentar a estabilidade relativa do sistema ao mesmo tempo que torna a resposta do sistema mais rápida devido ao seu efeito antecipatório.

Controladores PID podem ser descritos pela equação 17:

$$MV = Kp * E + Ki * \int_0^t E dt + Kp * Kd \frac{dE}{dt} \quad (17)$$

Onde:

MV = Variável manipulada.

Kp = Ganho proporcional.

Ki = Ganho integral.

Kd = Ganho derivativo.

E = Erro ou desvio.

O controle PID consiste em um algoritmo matemático, que tem por função o controle preciso de uma variável em um sistema, permitindo ao sistema operar de forma estável no ponto de ajuste desejado, mesmo que ocorram variações ou distúrbios que afetam sua estabilidade. A implementação do controlador PID pode ser feita fazendo-se aproximações numéricas das derivadas e da integral que aparecem na lei de controle. Desta forma, é possível descrever cada uma das ações por uma equação de recorrência. As equações de recorrência descrevem as operações matemáticas a serem programadas no microcontrolador ou no microcomputador onde será implementado o PID digital. (AGUIRRE, 2007).

3.12 Métodos de Sintonia de Controle

Uma vez que um sistema de controle esteja instalado, os ajustes dos parâmetros do controlador devem ser realizados até que o desempenho do sistema seja considerado satisfatório, esta atividade é chamada de sintonia do controlador. Ao ter um modelo matemático representativo do sistema, a escolha dos parâmetros do controlador recai no desenvolvimento de um projeto, que pode ser feito com base no método do lugar geométrico das raízes, dentre outros. Como nem sempre é possível se obter um modelo que represente adequadamente a dinâmica que se deseja controlar, se fez necessário o surgimento de técnicas, que não dependessem de modelo, para sintonia do controlador, como os métodos de Ziegler e Nichols.

Os métodos de projeto de Ziegler-Nichols (ZN) são os mais utilizados no controle de processo para determinar os parâmetros de um controlador PID. Embora esses métodos foram apresentados na década de 1940, eles ainda são amplamente utilizados. O método de resposta ao degrau é baseado em um teste de resposta ao degrau em malha aberta do processo. Portanto, exigindo que o processo seja estável, a resposta ao degrau unitário do processo é caracterizada por dois parâmetros, L e T. Estes são determinados traçando uma linha tangente no ponto de inflexão, onde a inclinação da resposta ao degrau tem seu valor máximo (NAGRATH e GOPAL, 2002).

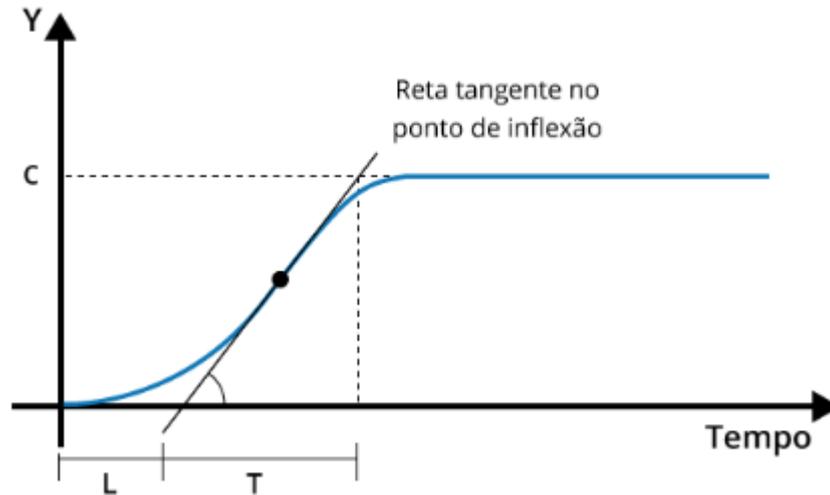
As interseções da tangente e dos eixos coordenados dão os parâmetros do processo conforme mostrado na Figura 9, e estes são usados no cálculo dos parâmetros do controlador. Os ganhos dos controladores PID podem ser obtidos a partir da Tabela 1.

Tabela 1 - Parâmetros do controlador ZN.

Parâmetro do Controlador	Kp	Ki	Kd
Controlador PID	$1.2 \cdot T/L$	$2 \cdot L$	$0.5 \cdot L$

Fonte: do autor (2022)

Figura 9 - Curva de resposta para o método de Ziegler e Nichols.



Fonte: Araújo (2007)

O método de Cohen e Coon foi proposto em 1953 pelo engenheiro G. H. Cohen e pelo matemático G. A. Coon. Eles desenvolveram regras de ajuste de controladores visando obter respostas em malha fechada com uma razão de decaimento de $\frac{1}{4}$ (aproximadamente 20% de *overshoot*). Esse tipo de resposta apresenta melhor desempenho para rejeitar distúrbios na carga, que é o principal objetivo a ser alcançado para processos contínuos que trabalham em um mesmo ponto de operação. Para esses sistemas, o principal papel desempenhado pelo controlador é a rejeição de distúrbios (COHEN, 1953).

Os parâmetros de controle do método são dados pelas equações da Tabela 2, onde $\tau = L/(T + L)$.

Tabela 2 - Parâmetros do controlador CC.

Parâmetro do Controlador	K_p	K_i	K_d
Controlador PID	$\frac{1}{k} * \frac{\tau}{L} \left(\frac{16\tau+30}{12\tau} \right)$	$\frac{L(32+\frac{6L}{\tau})}{13+8\frac{L}{\tau}}$	$4 \frac{L}{11+2\frac{L}{\tau}}$

Fonte: do autor (2022)

O método *Chien, Hrones and Reswick method* (CHR) foi desenvolvido em 1952, como uma alternativa para resolver problemas com resposta ao degrau. Apesar dos ganhos serem menores do que os obtidos por *Ziegler-Nichols*, o sistema tem mais estabilidade e fornece uma melhor maneira de selecionar um compensador para aplicações de controle de

processos. Na indústria de processos, os parâmetros do controlador geralmente são ajustados conforme a recomendação CHR (NEIL, 2002).

Os parâmetros do controlador do método de resposta do ponto de ajuste de Chien, Hrones e Reswick estão resumidos na Tabela 3 para um *overshoot* de 0% e na Tabela 4 para um *overshoot* de 20%, onde $a = CL/T$.

Tabela 3 - Parâmetros do controlador CHR, overshoot = 0%.

Parâmetro do Controlador	Kp	Ki	Kd
Controlador PID	0.6/a	T	0.5*L

Fonte: do autor (2022)

Tabela 4 - Parâmetros do controlador CHR, overshoot = 20%.

Parâmetro do Controlador	Kp	Ki	Kd
Controlador PID	0.95/a	1.4*T	0.47*L

Fonte: do autor (2022)

3.13 Matlab

O *Matrix Laboratory* (MATLAB) é um *software* de computação numérica de análise e visualização de dados. Embora seu nome signifique Laboratório de Matrizes, seus propósitos atualmente são mais amplos. Desenvolvido como um programa para operações matemáticas sobre matrizes, ao longo dos anos transformou-se em um sistema computacional bastante útil e flexível. Seu ambiente de trabalho é funcional, pois os problemas e soluções são escritos em linguagem matemática e não na linguagem de programação tradicional, como muitos outros *softwares* utilizam. Assim, o MATLAB é uma ferramenta e uma linguagem de programação de alto nível, e tem como principais funções: construção de gráficos e compilação de funções, manipulação de funções específicas de cálculo e variáveis simbólicas (CHAPMAN, 2006).

O MATLAB possui uma infinidade de bibliotecas auxiliares, denominadas *Toolboxes* que otimizam o tempo gasto para realizar tarefas, uma vez que o usuário poderá utilizar muitas funções já definidas, poupando o tempo de criá-las. Por outro lado, infelizmente, os

programas feitos são difíceis de serem executados num ambiente fora do MATLAB.

3.13.1 Toolbox Ident

O *System Identification Toolbox* (SITB) foi lançado em sua primeira versão em 1987. O SITB foi desenvolvido para construir modelos precisos e simplificados de sistemas complexos a partir de dados de séries temporais ruidosas. Fornece ferramentas para criar modelos matemáticos de sistemas dinâmicos com base em dados de entrada/saída (E/S) observados. A caixa de ferramentas apresenta uma interface gráfica de usuário flexível que auxilia na organização de dados e modelos. As técnicas de identificação fornecidas com esta caixa de ferramentas são úteis para aplicações que vão desde o projeto do sistema de controle e processamento de sinal até a análise de séries temporais e análise de vibração (LJUNG, 1995).

No SITB existem diversos modelos para estimação do sistema, tais como função de transferência, auto regressivo com entradas exógenas (ARX), auto regressivo com média móvel e entradas exógenas (ARMAX), modelo por espaço de estados, entre outros (SILVA, 2015).

Na identificação do sistema, alguns parâmetros devem ser determinados pelo usuário, incluindo a estrutura do modelo, a ordem do modelo e a frequência de amostragem dos sinais de E/S. As estruturas modelo são determinadas a partir do sistema em estudo, a frequência de amostragem para os sinais de E/S dependem do sistema de medição e da região de interesse, determinados na etapa de projeto experimental. A ordem do modelo deve ser determinada com base na estrutura do sistema. A ordem do modelo geralmente é muito alta e não considera os efeitos da perturbação, então os usuários podem precisar definir uma ordem de modelo mais alta para obter um resultado de identificação preciso (LJUNG, 1995).

3.13.2 PID Tuner

A *Toolbox PID Tuner* ajusta automaticamente os ganhos de um controlador PID para uma planta SISO para alcançar um equilíbrio entre desempenho e robustez. É possível especificar o tipo de controlador, como PI, PID com filtro derivado ou controladores PID de dois graus de liberdade. Os gráficos de análise permitem examinar o desempenho do controlador nos domínios de tempo e frequência. O desempenho do controlador é refinado iterativamente para ajustar a largura de banda do loop e a margem de fase ou para favorecer o

rastreamento do ponto de ajuste ou a rejeição de distúrbios. O PID *Tuner* representa uma planta (sistema) por um modelo numérico, como uma função de transferência ou um modelo de espaço de estados (LAGHARI, 2022).

Os objetivos típicos de ajuste do PID incluem: estabilidade de malha fechada, desempenho adequado, robustez adequada. O algoritmo do PID *Tuner* para ajuste de controladores PID atende a esses objetivos ajustando os ganhos PID para alcançar um bom equilíbrio entre desempenho e robustez. Por padrão, o algoritmo escolhe uma frequência de cruzamento (largura de banda do *loop*) com base na dinâmica da planta e projeta para uma margem de fase alvo de 60° . Quando é alterado interativamente o tempo de resposta, a largura de banda, a resposta transitória ou a margem de fase usando a interface do sintonizador PID, o algoritmo calcula os novos ganhos PID. Para uma dada robustez (margem de fase mínima), o algoritmo de ajuste escolhe um projeto de controlador que equilibre as duas medidas de desempenho, rastreamento de referência e rejeição de perturbação. É possível alterar o foco do design para favorecer uma dessas medidas de desempenho através da caixa de diálogo *Options* no PID Tuner. Quando se altera o foco do projeto, o algoritmo tenta ajustar os ganhos para favorecer o rastreamento de referência ou a rejeição de distúrbios, enquanto atinge a mesma margem de fase mínima. Quanto mais parâmetros ajustáveis houver no sistema, mais provável é que o algoritmo PID possa atingir o foco de projeto desejado sem sacrificar a robustez (MATHWORKS, 2022).

4 MATERIAL E MÉTODOS

O desenvolvimento do trabalho foi realizado utilizando os conceitos de transferência de calor, modelagem de sistemas, controladores PID, juntamente com as técnicas e conhecimentos de eletrônica para a montagem do protótipo e de desenvolvimento de software para programar o Arduino Mega de maneira adequada para cada etapa de desenvolvimento do projeto.

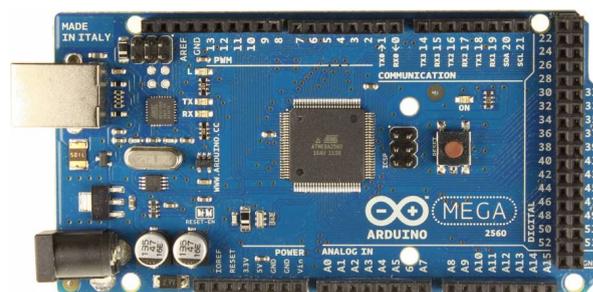
A utilização do Matlab e seus recursos e *Toolboxes* foram importantes no desenvolvimento do projeto, proporcionando facilidades na Identificação do Sistema e também no projeto do Controlador PID a ser utilizado. As técnicas a serem apresentadas podem ser utilizadas em um vasto número de projetos.

4.1 Materiais utilizados

O experimento foi conduzido em duas etapas: montagem da planta para aquisição de dados e montagem do protótipo juntamente com o controlador desenvolvido. Para isso, foram utilizados os seguintes componentes:

- Arduino Mega 2560 (Figura 10).

Figura 10 - Arduino Mega.



Fonte: Baú da Eletrônica (2022)

- Sensor de Temperatura LM35 (Figura 11).

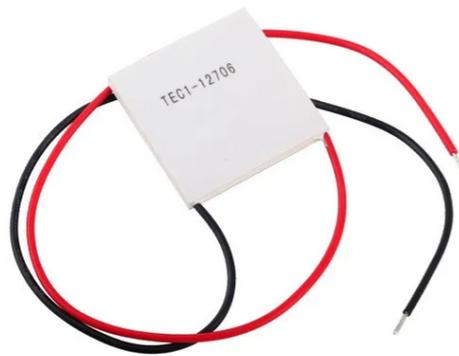
Figura 11 - Sensor de temperatura LM35.



Fonte: Baú da eletrônica (2022)

- Pastilha Peltier Tec1-12706 (Figura 12).

Figura 12 - Pastilha Peltier Tec1-12706.



Fonte: Baú da Eletrônica (2022)

- Módulo *Driver* PWM 5 A 36V 15A (Figura 13).

Figura 13 - Módulo *Driver* PWM 5 A 36V 15A.



Fonte: Baú da Eletrônica (2022)

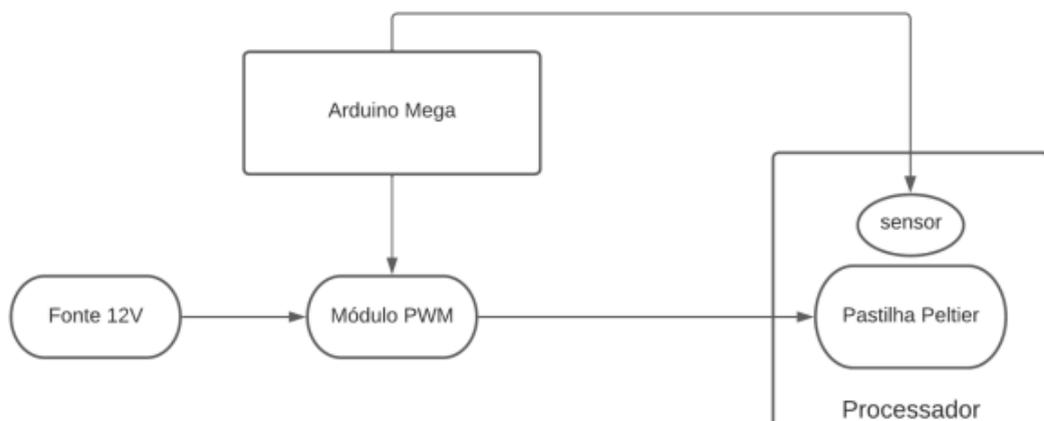
- Fonte 12V, 4 A.
- Dissipador de alumínio.
- Processador Intel Pentium E2220 - 65 nm.

4.2 Montagem da planta para aquisição de dados

Inicialmente a pastilha de Peltier foi fixada sobre o processador Intel Pentium E2220 juntamente com o sensor de temperatura LM35, este responsável pelo sensoriamento da temperatura disponibilizando-a para o Arduino Mega 2560 em forma de sinal analógico. A fonte de 12 [V] alimenta o módulo PWM controlado pelo Arduino Mega e por meio deste controle é possível variar a tensão de alimentação aplicada na pastilha de Peltier. Quanto maior o valor do PWM maior é a tensão na pastilha e consequentemente maior é seu poder de resfriamento do processador. Um valor de 0 para o PWM é equivalente a 0 [V] aplicado na pastilha de Peltier e um valor de 255 para o PWM é equivalente à aplicação de 12 [V] na pastilha de Peltier.

A planta representativa do sistema foi montada na configuração apresentada na Figura 14.

Figura 14 - Diagrama da planta.



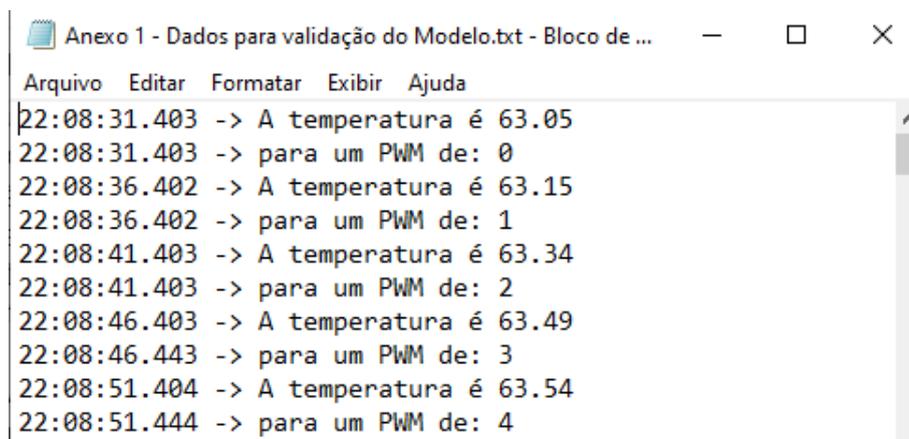
Fonte: o autor (2022)

Após a montagem da planta foi desenvolvido um software no Arduino Mega para realizar o controle da tensão aplicada na pastilha (variando os valores de PWM), leitura e armazenamento dos dados da temperatura no processador.

Para a implementação do software, foram definidas as pinagens a serem utilizadas no Arduino Mega e também foi desenvolvida a rotina para a leitura dos dados de temperatura. A rotina implica na variação dos valores de PWM (0 até 255), leituras e armazenamento das temperaturas do processador, que operava em condições nominais. Por exemplo, para um valor PWM de 50, foram feitas 10 medições de temperatura, uma no instante igual a 1 [s], outra no instante 2 [s] até o instante 10 [s], por fim é calculada a média destes valores e armazenada, tal estratégia foi adotada para reduzir os erros de leituras. O código implementado encontra-se no APÊNDICE A.

Os resultados das leituras foram extraídos para um arquivo contendo as informações de hora, valor de PWM e valor da temperatura no processador, como mostrado na figura 15.

Figura 15 - Exemplo de coleta de dados.



```

Arquivo Editar Formatar Exibir Ajuda
22:08:31.403 -> A temperatura é 63.05
22:08:31.403 -> para um PWM de: 0
22:08:36.402 -> A temperatura é 63.15
22:08:36.402 -> para um PWM de: 1
22:08:41.403 -> A temperatura é 63.34
22:08:41.403 -> para um PWM de: 2
22:08:46.403 -> A temperatura é 63.49
22:08:46.443 -> para um PWM de: 3
22:08:51.404 -> A temperatura é 63.54
22:08:51.444 -> para um PWM de: 4

```

Fonte: o autor (2022)

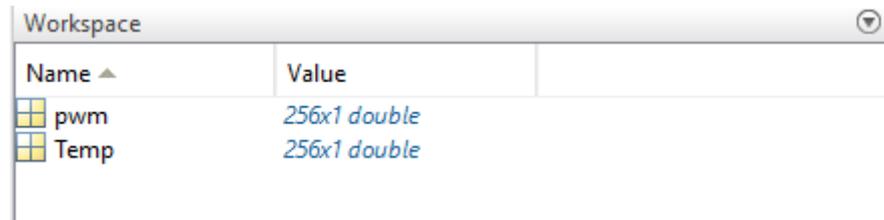
4.3 Identificação Do sistema

O modelo matemático do sistema em estudo foi conduzido seguindo o princípio de Modelagem em Caixa Preta, com o intuito de obter o modelo matemático que representa o sistema em estudo. Os resultados provenientes do tópico anterior são utilizados para a modelagem, sendo a entrada do sistema representada pelos valores de PWM, a planta é o sistema propriamente dito e a saída é a temperatura do processador.

Para obtenção do Modelo Matemático do Sistema, foi utilizado o *SITB*.

Antes de utilizar a *toolbox* foi necessário realizar o tratamento dos dados e importá-los para a *workspace* do Matlab. A Figura 16 apresenta os dados que serão utilizados na modelagem do sistema.

Figura 16 - *Workspace* do Matlab com os dados importados.



Fonte: o autor (2022)

Para utilizar a *ToolBox System Identification*, é necessário digitar o comando *ident* no *Command Window* do Matlab, como mostra a Figura 17.

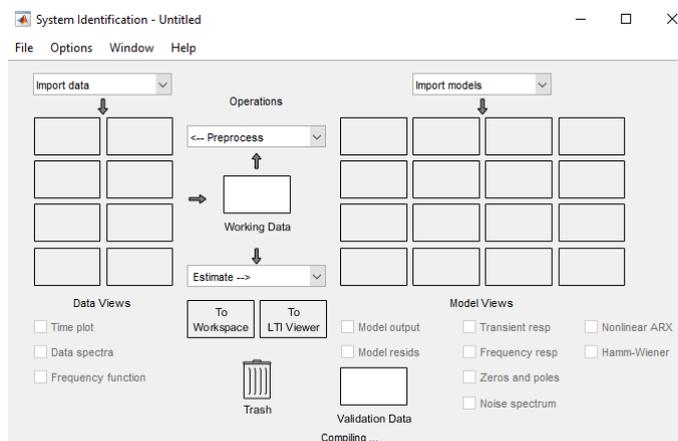
Figura 17 - *Command Window*.



Fonte: o autor (2022)

A *Graphical User Interface* (GUI) referente à SITB é representada na Figura 18.

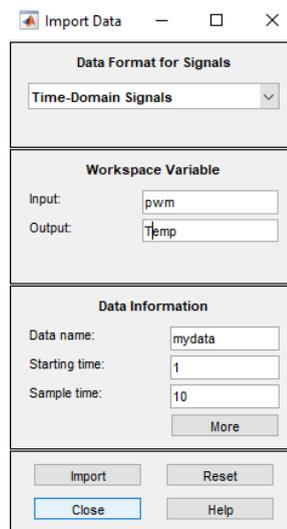
Figura 18 - GUI do SITB.



Fonte: o autor (2022)

No GUI é preciso importar os dados da *Workspace* e para realizar esta tarefa é necessário selecionar o campo *Import Data* e em seguida a opção *Time domain data*, assim será disponibilizada uma nova janela como a mostrada na Figura 19. Nos campos *input* e *output* foram alocados os dados de PWM e Temperatura respectivamente, por fim é necessário definir o tempo de amostragem e realizar o *import* dos dados.

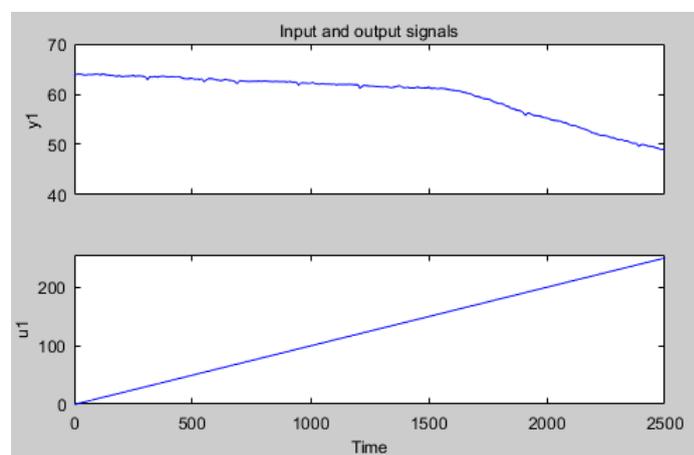
Figura 19 - Janela *Import Data*.



Fonte: o autor (2022)

Neste nível é possível analisar o comportamento referente aos dados da planta, escolhendo a opção *Time Plot*. Para a planta em estudo foi obtido o resultado da Figura 20, no qual é possível observar que a temperatura cai de acordo com o aumento do valor do PWM.

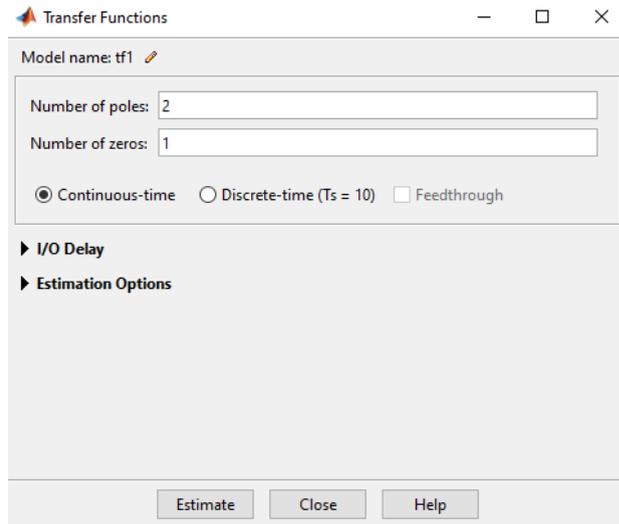
Figura 20 - Gráfico de entrada u_1 (PWM) e saída y_1 (Temperatura) da planta.



Fonte: O autor (2022)

Para estimar a função de transferência do sistema, objetivo deste tópico, deve-se escolher a opção *Transfer Function Model* no campo denominado *Estimate*, que fará com que abra uma nova janela (Figura 21). A Função de Transferência que representará o sistema depende do número de pólos e zeros definidos pelo usuário.

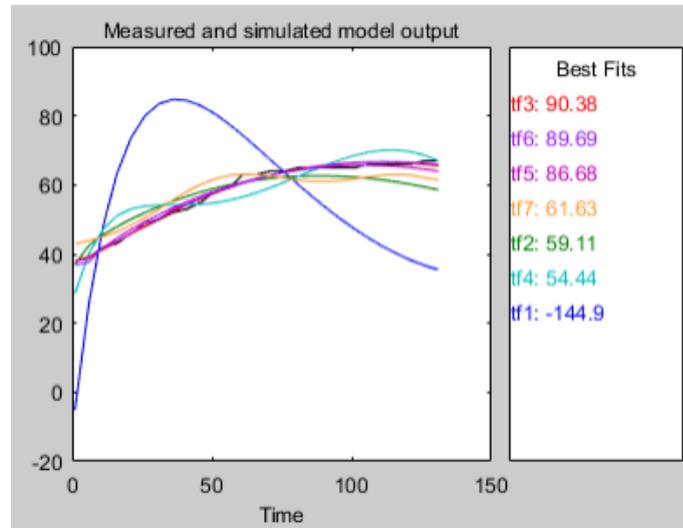
Figura 21 - Janela do *Transfer Function*.



Fonte: o autor (2022)

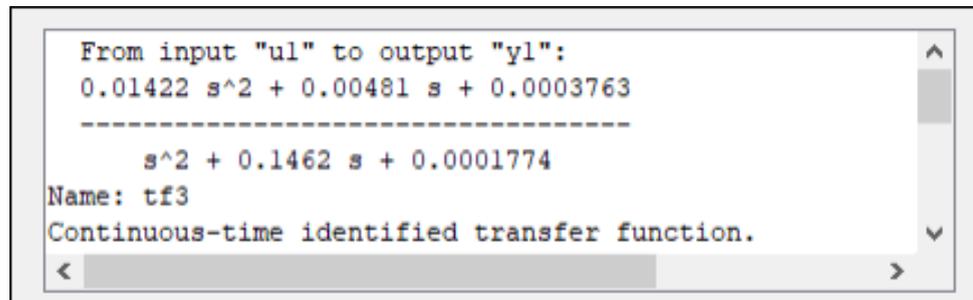
No presente trabalho, foram realizadas 7 estimativas variando os números de pólos e zeros da FT. A Figura 22 apresenta todas as estimativas dos modelos matemáticos conforme as escolhas de pólos e zeros. A função de transferência *tf3* corresponde ao maior *Fit*, ou seja, maior aproximação do Sistema em estudo e por isso será definida como a FT que representará o sistema e será a base para estudo ao longo deste trabalho. Possuindo 2 polos e 2 zeros, seu resultado numérico é mostrado na Figura 23.

Figura 22 - Estimativas das FTs.



Fonte: o autor (2022)

Figura 23 - Função de Transferência de melhor aproximação.



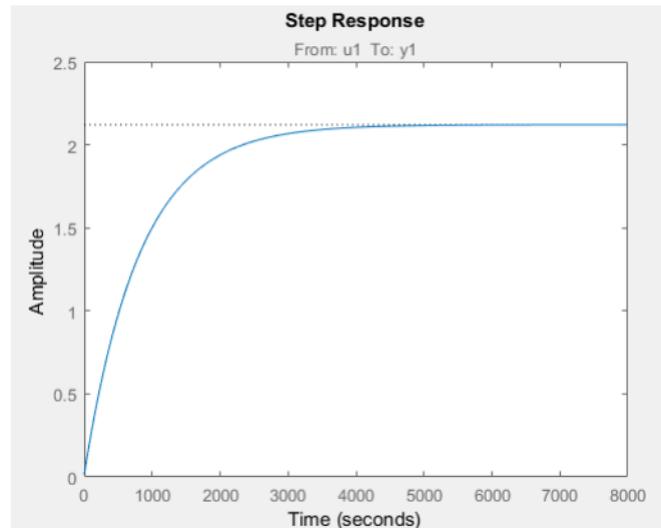
Fonte: o autor (2022)

4.4 Definição dos parâmetros de desempenho

Após identificar o sistema, com a posse da Função de Transferência que melhor o representa, é necessário verificar o comportamento do sistema para a entrada ao degrau unitário, juntamente com o tempo de subida, tempo de assentamento, *overshoot* e erro em regime permanente.

A entrada ao degrau unitário é utilizada como função de teste desde que seu salto inicial instantâneo revela a velocidade que um sistema responde a entradas com mudanças abruptas. A resposta do sistema a este estímulo é mostrada na Figura 24.

Figura 24 - Resposta à entrada degrau.



Fonte: o autor (2022)

As informações pertinentes ao modelo encontrado são apresentadas na Figura 25.

Figura 25 - Informações do Sistema

```

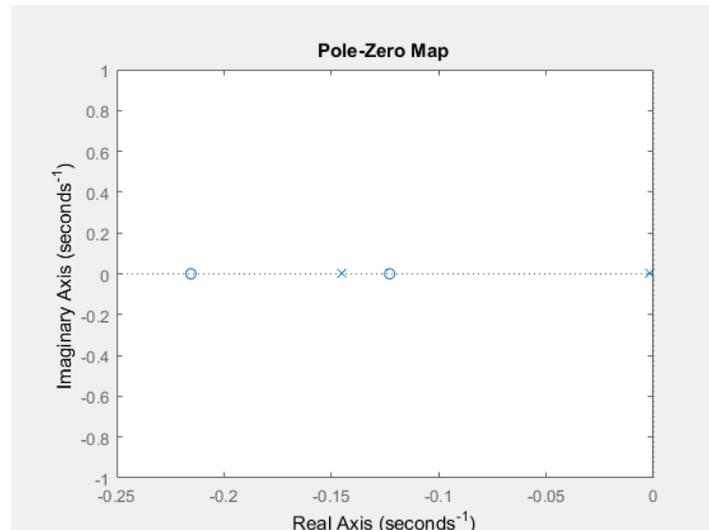
RiseTime: 1.7955e+03
SettlingTime: 3.1967e+03
SettlingMin: 1.9106
SettlingMax: 2.1211
Overshoot: 0
Undershoot: 0
Peak: 2.1211
PeakTime: 8.6184e+03

```

Fonte: o autor (2022)

A Figura 26, representa o mapa de pólos e zeros da FT, e ao analisá-lo é possível inferir que o sistema é estável, visto que todos os pólos se encontram no SPE.

Figura 26 - Mapa de pólos e zeros.



Fonte: o autor (2022)

Ao pensar em um protótipo funcional para ser utilizado no controle de temperatura de processadores reais e em posse das informações apresentadas na Figura 19 os requisitos de desempenho para o controlador de temperatura a ser desenvolvido terá um desempenho na qual o tempo de acomodação será 5 vezes menor que o tempo atual, o valor de 640 [s] foi escolhido com base em simulações feitas, no processador utilizado na modelagem, e considerando os tempos de inicialização do computador como um todo juntamente com o período necessário para que o mesmo alcance altas temperaturas. O erro em regime permanente deverá ser próximo de 0, para que o processador mantenha sempre próximo do valor desejado. Com as possíveis transformações que a inserção de um controlador na planta poderá trazer será objetivado um *overshoot* menor ou igual a 10%. Os valores definidos são apresentados na Tabela 5:

Tabela 5 - Valores dos parâmetros de desempenho.

Parâmetro	Valor
Tempo de acomodação	640 [s]
Erro em Regime Permanente	± 0
<i>Overshoot</i>	$\leq 10\%$

Fonte: o autor (2022)

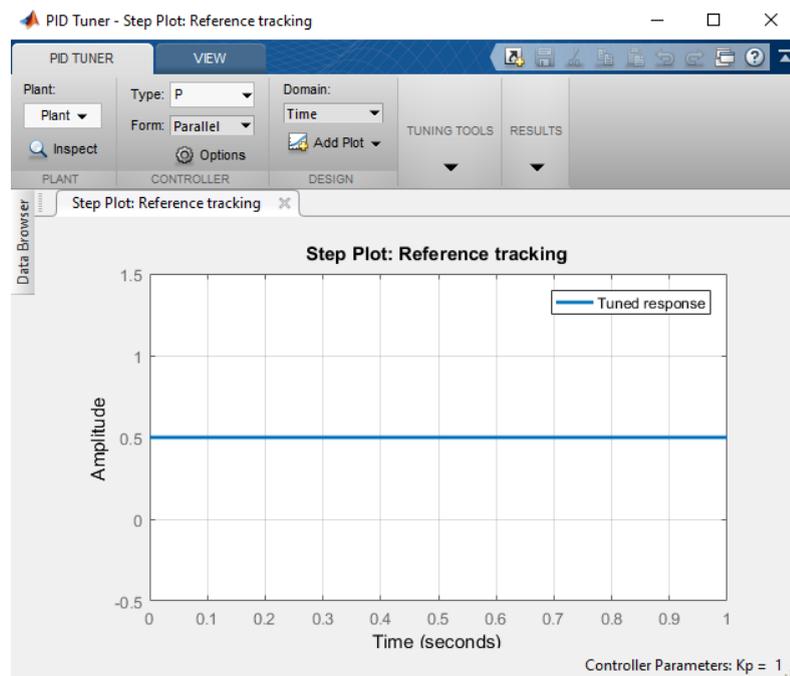
4.5 Desenvolvimento do controlador utilizando o PID Tuner

Para o desenvolvimento do Controlador PID buscou-se os ganhos proporcional K_p , integrativo K_i e o derivativo K_d , que farão parte do algoritmo matemático que controlará a temperatura do processador.

Para obter os ganhos necessários para que o Sistema obedeça aos requisitos de desempenhos definidos na Tabela 5, foi utilizado a ferramenta PID *Tuner* do Matlab.

Como apresentado no referencial teórico, a ferramenta oferece ajuste dinâmico para a obtenção dos ganhos, equilibra desempenho e robustez, além de oferecer um GUI intuitivo, como mostrado na Figura 27.

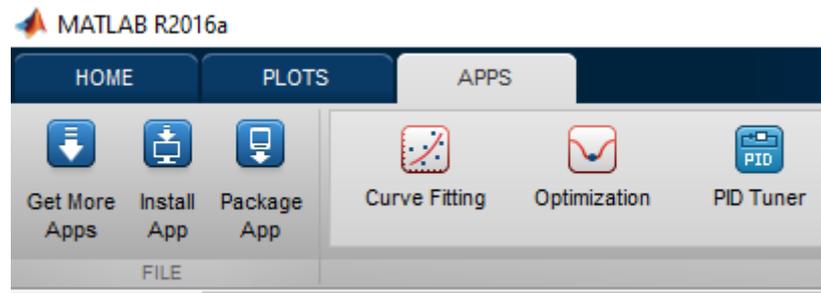
Figura 27 - GUI referente ao PID Tuner.



Fonte: o autor (2022)

Para iniciar a *Toolbox* do PID *Tuner* é necessário acessar a aba APPS na página principal do Matlab (veja Figura 28) e clicar em sua opção.

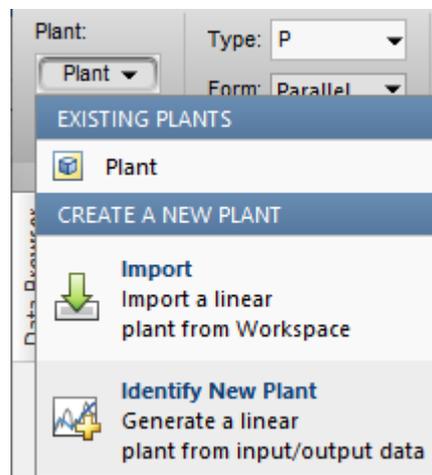
Figura 28 - Página de APPS do Matlab.



Fonte: o autor (2022)

No Guia do PID Tuner é necessário importar a Planta a qual deseja-se estudar, acessando o campo *Plant* e em seguida *Import*, como mostra a Figura 29. No presente trabalho, a FT obtida no tópico de identificação do sistema representa matematicamente o sistema e conseqüentemente a planta, assim é a FT quem deve ser importada da *workspace* do Matlab.

Figura 29 - Campo para importar a planta.

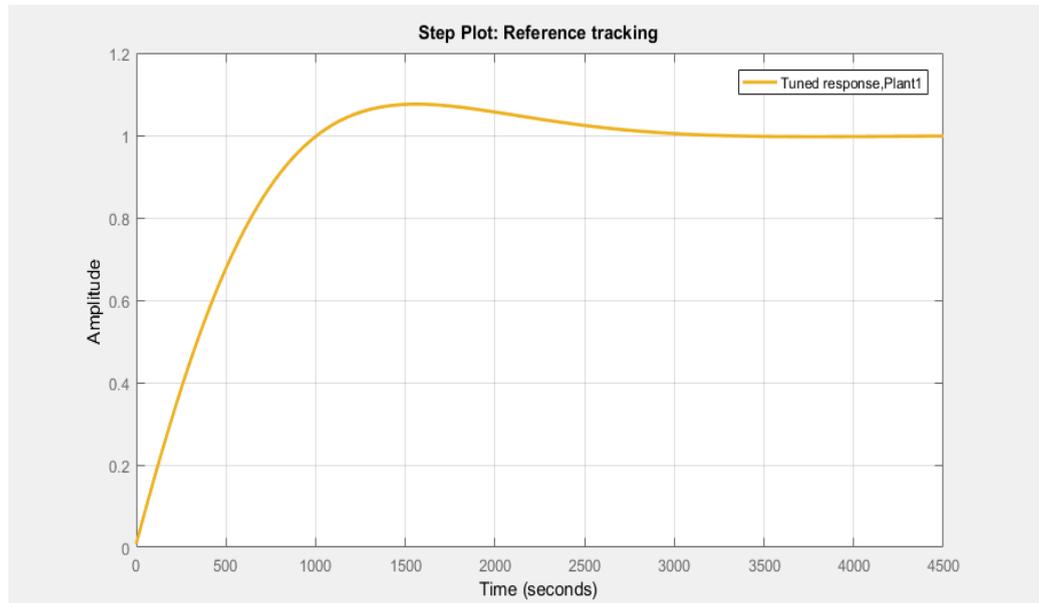


Fonte - o autor (2022)

Com a planta disponível na ferramenta, é preciso definir o modelo de Controlador a ser utilizado no campo *Type*. Na Figura 29 é possível observar o campo *Type*, e ao clicar em sua opção, abrirá vários modelos de controladores, e seguindo o objetivo deste trabalho, faz-se a escolha do modelo PID.

Neste momento é possível visualizar a resposta da Planta para uma entrada degrau unitário, mas sem a compensação almejada (Figura 30).

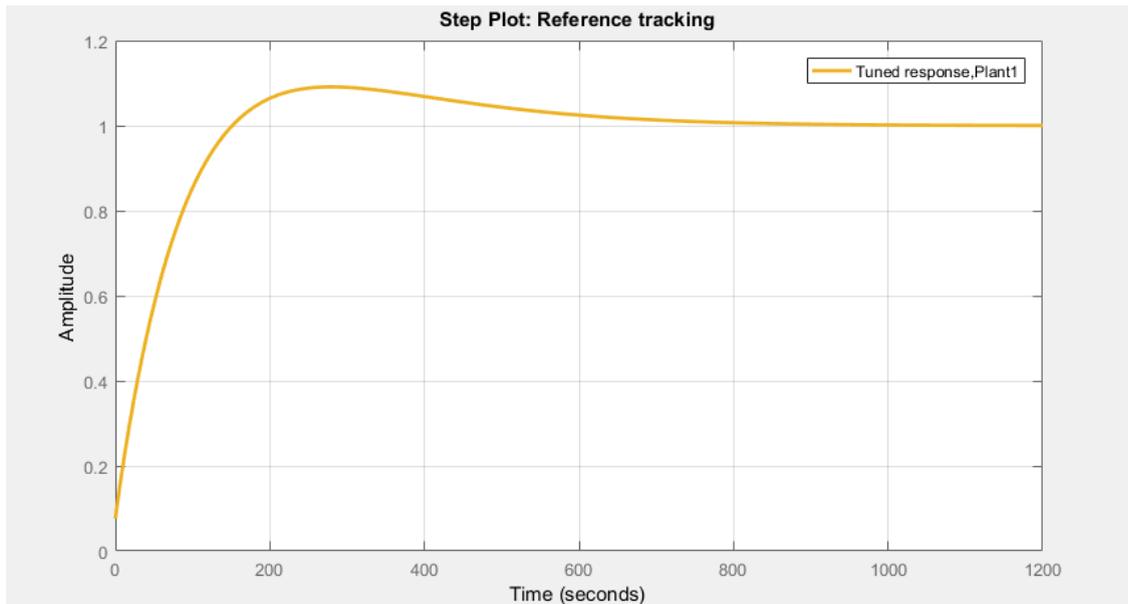
Figura 30 - Resposta do sistema não compensado.



Fonte: o autor (2022)

Para obter ajuste da resposta, aproximando-a dos valores objetivados na tabela 5, o PID *tuner* oferece recursos que facilitam na descoberta dos ganhos para o comportamento desejado. No presente trabalho foi necessário ajustar os valores de *Response Time* e *Transient Behavior* para alcançar uma resposta próxima dos valores objetivados. O novo comportamento pode ser visualizado na Figura 31 e os resultados na Figura 32.

Figura 31 - Resposta do sistema compensado.



Fonte: o autor (2022)

Figura 32 - Ganhos e resultados obtidos pelo PID *Tuner*.

Controller Parameters	
	Tuned
Kp	5.8148
Ki	0.026366
Kd	0
Tf	n/a

Performance and Robustness	
	Tuned
Rise time	107 seconds
Settling time	644 seconds
Overshoot	9.26 %
Peak	1.09
Gain margin	Inf dB @ NaN rad/s
Phase margin	83.7 deg @ 0.0157 rad/s
Closed-loop stability	Stable

Fonte: o autor (2022)

A Figura 32 apresenta o desempenho resultante dos ganhos (K_p , K_i e K_d) definidos pela Toolbox e que a performance obtida por eles concordam com os requisitos de desempenho almejados, ou seja, tempo de acomodação próximo de 640 [s], overshoot menor

que 10% e pela Figura 21 pode-se verificar que o erro em regime permanente é aproximadamente 0.

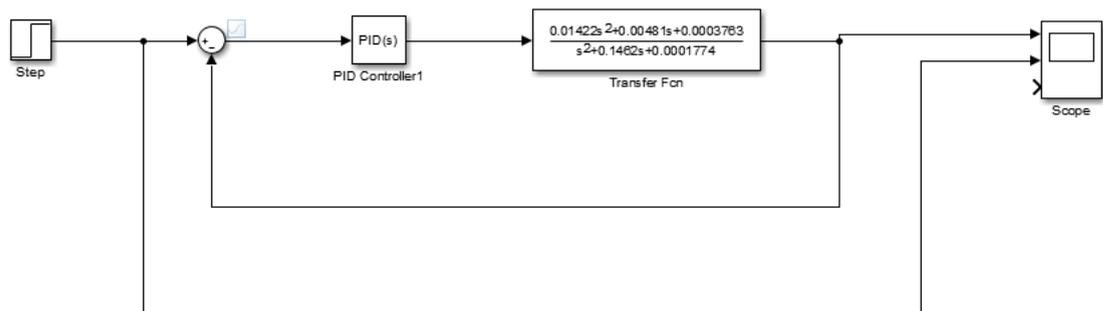
4.6 Simulações do controlador PID no Simulink

Antes da implementação do protótipo, faz-se necessária a execução de simulações, visando a verificação do comportamento do dispositivo a ser implementado.

O Simulink (ferramenta de simulação integrada no *software* MATLAB, que permite modelar, simular e analisar sistemas dinâmicos) simulará o comportamento geral do protótipo a ser desenvolvido. A Planta será representada por sua FT e será alimentada pelo Controlador PID, que por sua vez receberá o sinal de Erro do Sistema, composto pelo valor desejado subtraído do valor real (medido).

A Figura 33, apresenta a construção desenvolvida no Simulink. O primeiro bloco denominado *Step*, é referente ao valor desejado, ou seja, o valor do *setpoint*. O bloco PID foi implementado utilizando os valores encontrados no *PID Tuner*, apresentados anteriormente neste trabalho. A planta é representada pela FT encontrada no tópico de Identificação do Sistema. O *Scope* é utilizado apenas para leitura e visualização dos comportamentos.

Figura 33 - Implementação no Simulink.



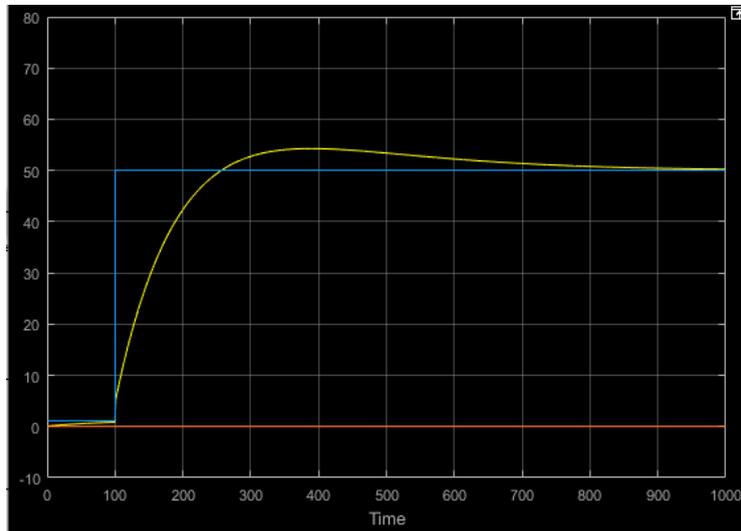
Fonte: O autor (2022)

4.6.1 Simulações sem adição de ruídos

Os valores de *Setpoint* na simulação são equivalentes ao valor da temperatura escolhida para o processador permanecer, dadas em [°C]. Nas simulações os gráficos em azul representam o valor do *setpoint* e os amarelos a resposta do sistema.

- **Simulação 1:** *Setpoint* definido para 50 [°C] a partir do instante 100 [s], resultado na Figura 34.

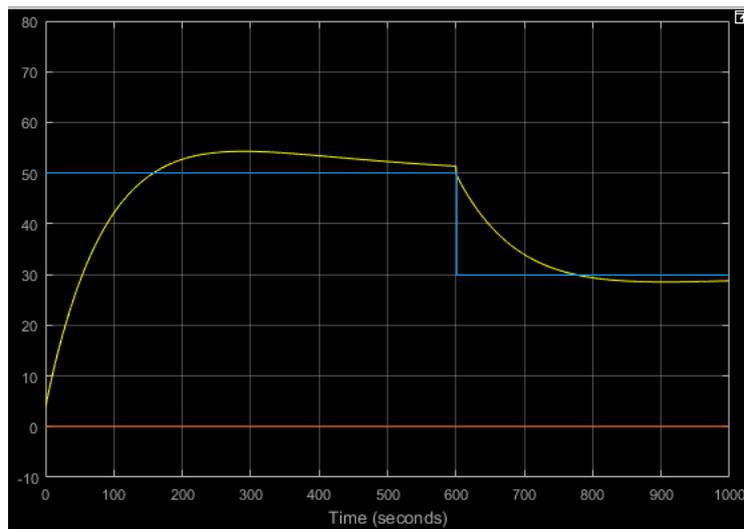
Figura 34 - Resultado da simulação 1.



Fonte: O autor (2022)

- **Simulação 2:** *Setpoint* definido para 50[°C] a partir do instante 0 [s] e alterado para 30[°C] no instante 600 [s], resultado na Figura 35.

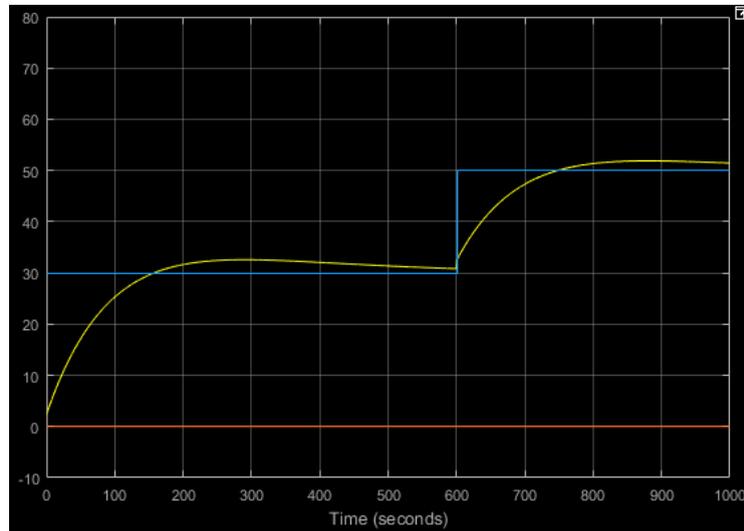
Figura 35 - Resultado da simulação 2.



Fonte: O autor (2022)

- **Simulação 3:** *Setpoint* definido para 30 [°C] a partir do instante 0 [s] e alterado para 50 [°C] no instante 600 [s], resultado na Figura 36.

Figura 36 - Resultado da simulação 3.

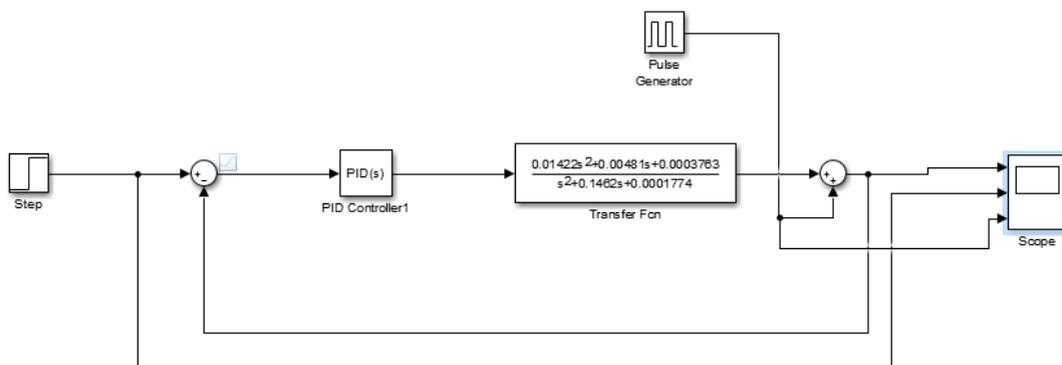


Fonte: O autor (2022)

4.6.2 Simulações com adição de ruídos

Para a realizar as simulações com a adição de ruídos foi necessário a adição de outros blocos no Simulink, como o *Pulse Generator*, que simulará a adição de ruído ao sistema, como mostrado na Figura 37. Este bloco simulará as variações de temperatura no ambiente, para podermos analisar a resposta do Sistema ante à essas alterações.

Figura 37 - Implementação com adição de ruídos no Simulink.

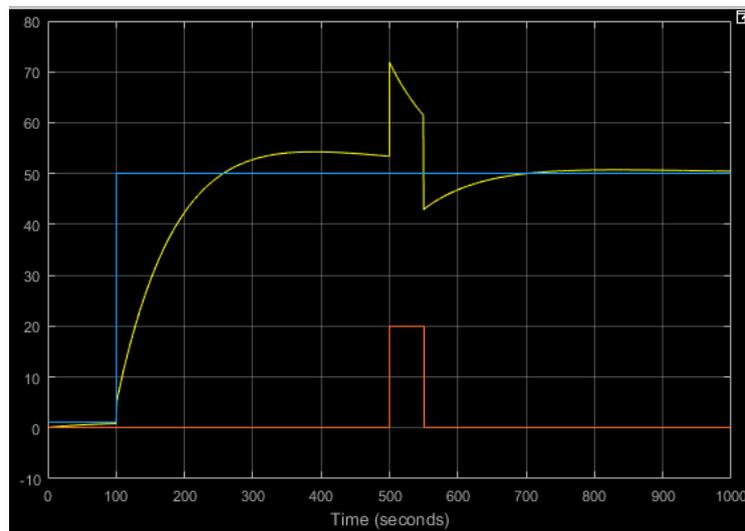


Fonte: O autor (2022)

Nas simulações os gráficos em azul representam o valor do *setpoint*, os amarelos a resposta do sistema e em vermelho o ruído acrescentado.

- **Simulação 1:** *Setpoint* definido para 50[°C] a partir do instante 100 [s] com adição de ruído com valor de 20[°C] no instante entre 500 e 550[s], resultado na Figura 38.

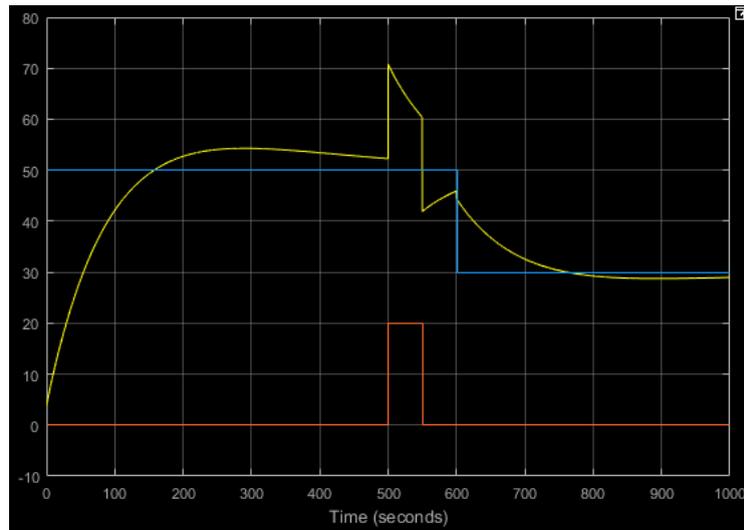
Figura 38 - Resultado da simulação 1.



Fonte: O autor (2022)

- **Simulação 2:** *Setpoint* definido para 50 [°C] a partir do instante 0 [s], alterado para 30 [°C] no instante 600 [s] e adição de ruído com valor de 20 [°C] no instante entre 500 e 550[s], resultado na Figura 39.

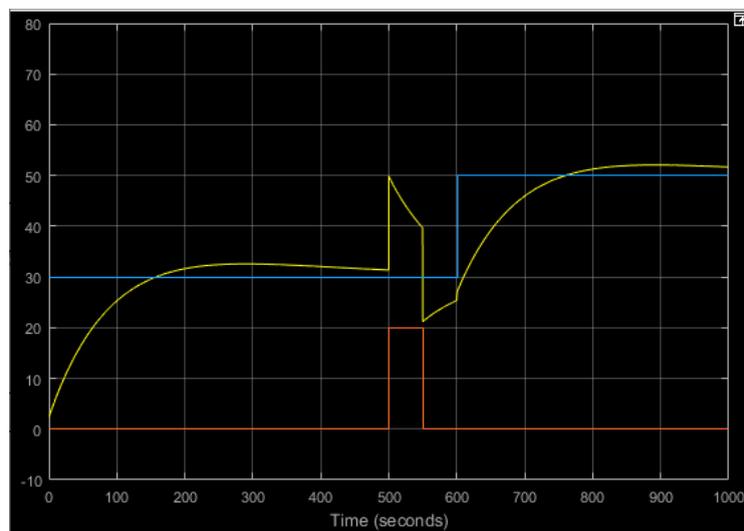
Figura 39 - Resultado da simulação 2.



Fonte: O autor (2022)

- **Simulação 3:** *Setpoint* definido para 30 [°C] a partir do instante 0 [s], alterado para 50 [°C] no instante 600 [s] e adição de ruído com valor de 20 [°C] no instante entre 500 e 550 [s], resultado na Figura 40.

Figura 40 - Resultado da simulação 3.



Fonte: O autor (2022)

4.6.3 Análise das simulações

Após as simulações, é possível observar que o sistema responde de maneira satisfatória às condições testadas. Para os testes onde não há o incremento de ruído a resposta foi suave e sempre permanecendo próxima aos valores definidos anteriormente.

Quando há o incremento de ruído, é possível verificar que a resposta é eficiente, visto que foi inserido um ruído com grande amplitude, e a saída do sistema tende a permanecer próxima dos valores definidos. As várias condições testadas reforçam a confiança para a implementação do protótipo, pois os resultados obtidos são satisfatórios.

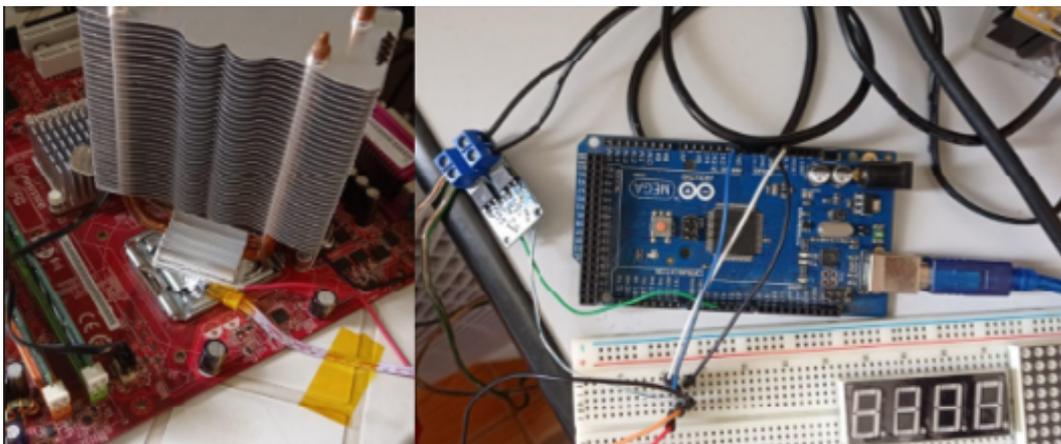
4.7 Implementação do protótipo

A implementação do protótipo é dividida em duas etapas: montagem do Hardware e Desenvolvimento do Software (SW).

4.7.1 Montagem do Hardware

Para a montagem do hardware manteve-se a mesma disposição apresentada no tópico referente à coleta de dados. A Figura 41 contém as imagens referentes à planta (lado esquerdo) e ao controlador (lado direito).

Figura 41 - Protótipo desenvolvido.



Fonte: o autor (2021)

4.7.2 Desenvolvimento do Software

O *Software* desenvolvido é o responsável por realizar o controle do sistema, as leituras da temperatura no processador, comparação com a temperatura desejada e cálculo do valor de PWM para que o erro (diferença entre Setpoint e Valor Medido) seja o menor possível.

No presente SW, foram inicialmente criadas variáveis para representar as medidas: temperatura atual, valor do *setpoint*, controle do PWM, erro, erro anterior, integral do erro, derivada do erro e os ganhos Kp, Ki e Kd que foram calculados anteriormente.

O loop de processamento do SW recebe o valor definido no *setpoint* (definido pelo usuário), realiza a leitura do sensor de temperatura correspondente à temperatura atual e realiza a subtração de ambos calculando o valor do erro.

A integral do erro é a soma de todos os erros calculados, a derivada do erro recebe o valor do erro atual subtraído do erro anterior. Com estes valores calculados, é necessário calcular o controle do PWM, este recebendo a soma das multiplicações entre Kp e o erro, Ki e a integral do erro, Kd e a derivada do erro, como mostrado na equação 18.

$$PWM = (Kp * Erro) + (Ki * Integral Erro) + (Kd * Derivada Erro) \quad (18)$$

O resultado se trata do valor do PWM, ou seja, este valor deve permanecer entre 0 e 255. Para que essa condição seja atendida, há duas condições que forçam um resultado menor que 0 para 0 e resultado maior que 255 para 255. Este sinal é o responsável por alimentar a pastilha de Peltier, variando a tensão aplicada na mesma.

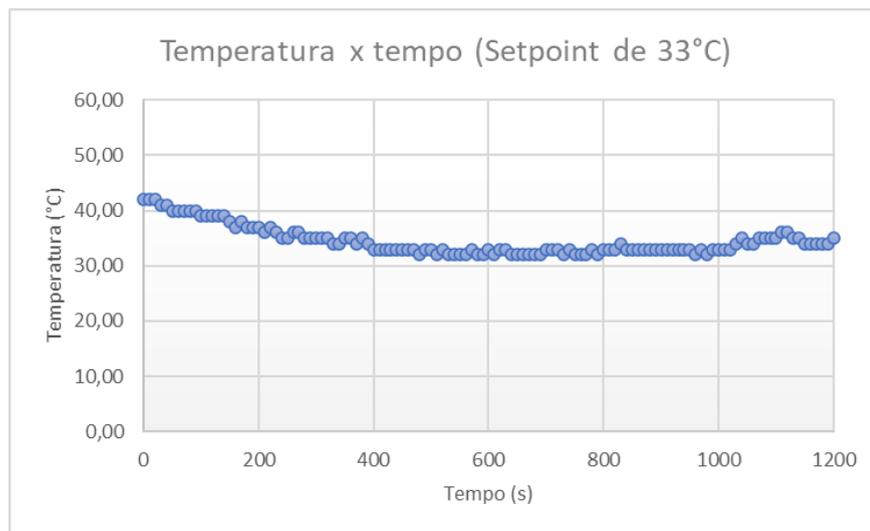
O código completo encontra-se no APÊNDICE B.

5 RESULTADOS E DISCUSSÃO

5.1 Sistema com protótipo e processador nas condições nominais

A primeira análise de resultados foi realizada, para verificar o comportamento do protótipo no controle da temperatura do processador, quando o mesmo está operando nas condições nominais. A Figura 42 representa o resultado das temperaturas lidas no processador, iniciando-se de uma temperatura de operação normal do processador. Foi definido um *setpoint* de 33 [°C] e feita a coleta dos resultados, a fim de verificar a capacidade do protótipo de manter a temperatura próxima ao valor desejado. É válido lembrar que a temperatura de 33 [°C] para o processador é considerada uma temperatura baixa, pois os mesmos têm uma faixa de operação normal, de 40 a 65 [°C].

Figura 42- Gráfico das leituras da temperatura para uma condição normal de operação.



Fonte: o autor (2022)

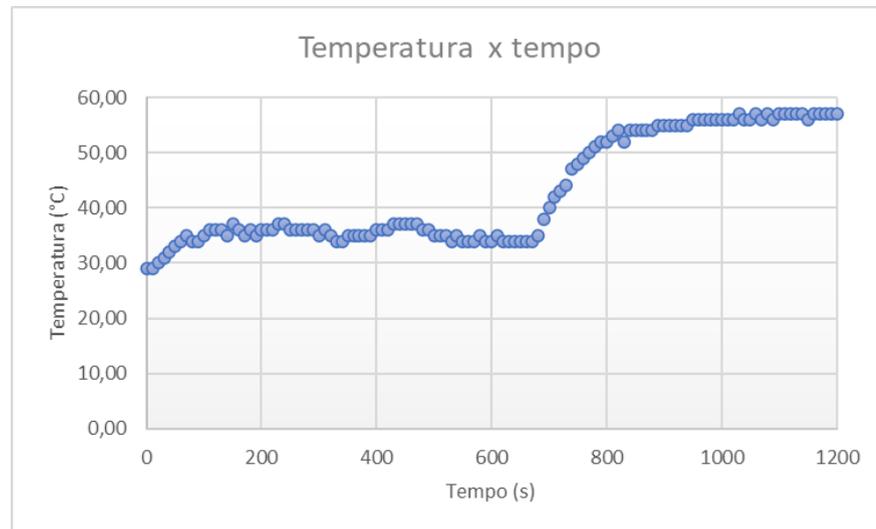
5.2 Sistema sem protótipo e processador em condições de estresse

Para obtenção dos parâmetros das condições usuais de temperatura no processador, *score* de pontuação e certificar a eficiência do protótipo foi realizado o teste de estresse sem o protótipo, e o resultado é apresentado na Figura 43.

Um teste de estresse analisa a capacidade de desempenho do computador quando este está em seu uso máximo por um período prolongado, e ao performar essa condição consequentemente a temperatura aumenta durante esses testes, indo próxima e atingindo o

limite máximo suportado pelo processador.

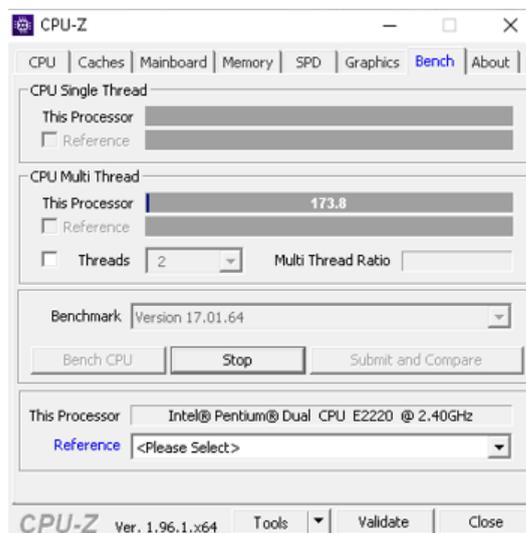
Figura 43 - Gráfico das leituras de temperatura para a condição de estresse sem o protótipo.



Fonte: o autor (2022)

O resultado de performance na condição acima é demonstrado na Figura 44.

Figura 44 - Resultado de performance na condição de estresse sem o protótipo.

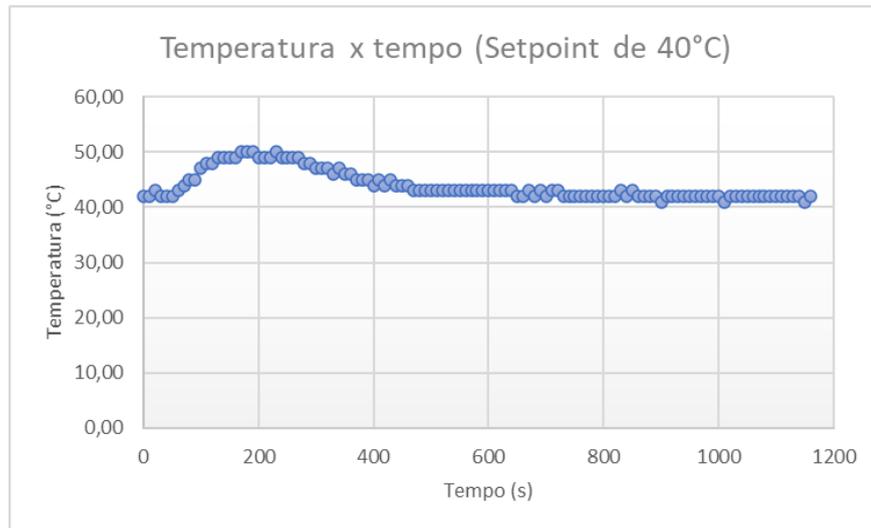


Fonte: o autor (2022)

5.3 Sistema com protótipo e processador em condições de estresse

Para as demais análises, os testes foram continuados com a aplicação de estresse no processador. Os testes apresentados a seguir foram realizados para diferentes valores de *setpoint* e ambos na condição de estresse.

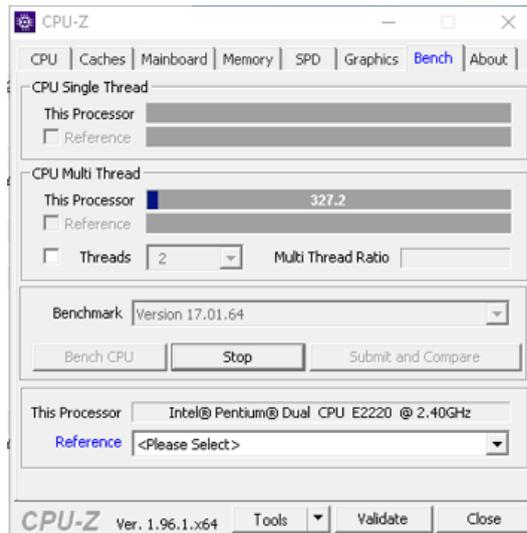
Figura 45 - Gráfico das leituras de temperatura para a condição de estresse e *setpoint* de 40 [°C].



Fonte: O autor (2022)

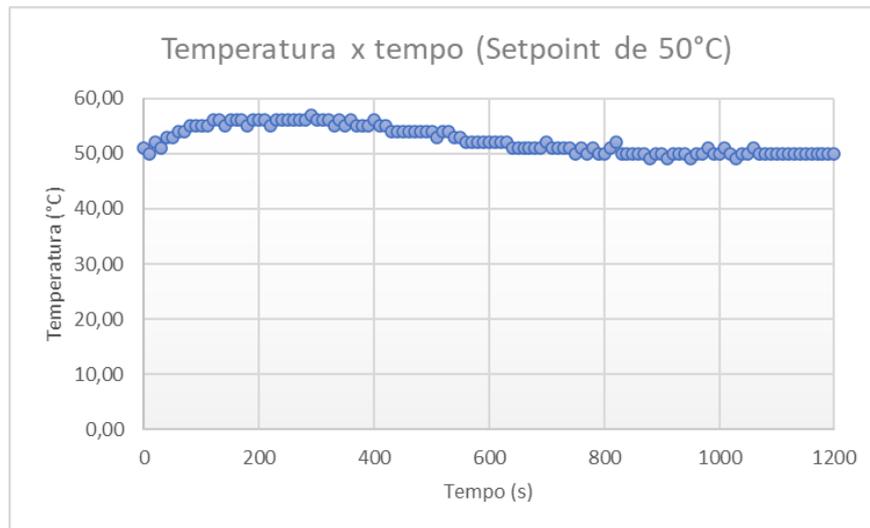
Para o teste representado na Figura 45, foi verificado o comportamento de performance do processador durante sua condição de estresse, a verificação é feita por um software que pontua a condição atual através de algumas variáveis, sendo a temperatura a variável de maior impacto. Para a condição onde o *setpoint* foi de 40 [°C] o resultado da performance é mostrado na Figura 46.

Figura 46 - Resultado de performance na condição de estresse para um *setpoint* de 40 [°C].



Fonte: O autor (2022)

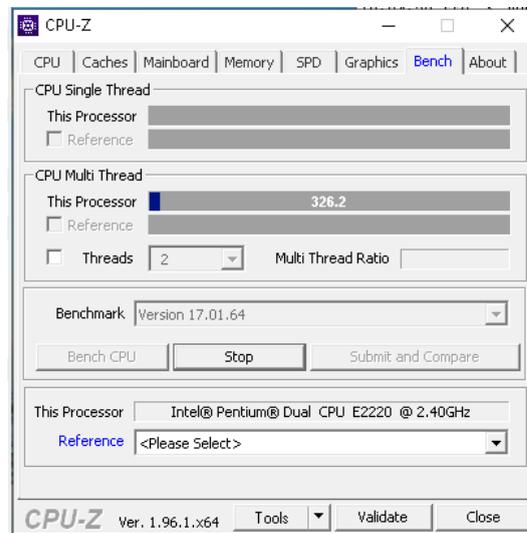
Figura 47 - Gráfico das leituras de temperatura para a condição de estresse e *setpoint* de 50 [°C].



Fonte: O autor (2022)

Para o teste representado na Figura 47, também foi feita a análise de performance, seguindo as mesmas condições e o resultado é demonstrado na Figura 48.

Figura 48 - Resultado de performance na condição de estresse para um *setpoint* de 40 [°C].



Fonte: O autor (2022)

6 CONCLUSÃO

O protótipo desenvolvido atuou conforme o previsto, em condições nominais de operação do processador, foi possível manter sua temperatura próxima do valor definido no *setpoint*, mesmo escolhendo um valor de temperatura abaixo dos valores em que um processador normalmente opera.

O ponto de maior valia para o protótipo é a validação da performance alcançada quando o processador foi posto sob a condição de estresse, porque assim é possível analisar o protótipo nas condições mais severas de uso, e os resultados obtidos são louváveis, para os dois testes, *setpoint* de 40 [°C] e 50 [°C], os resultados de performance obtidos nos testes de estresse foram excelentes, sendo as duas temperaturas definidas na faixa normal de operação, o processador manteve um *score* mínimo de 326 pontos. Comparando os dois testes em que se faz o uso do protótipo com o teste em que se utiliza o dissipador convencional, pode-se observar que o *score* de performance cai para 173 pontos e a temperatura medida alcança os 60 [°C], as condições mostraram que a temperatura se elevaria ainda mais, mas para proteger o processador, pois nessas condições o mesmo começou a apresentar falhas, foi religado o protótipo.

7 TRABALHOS FUTUROS

Para trabalhos futuros e pensando em um protótipo mais versátil, é sugerido a implementação de uma interface para alterar os valores de *setpoint* sem a necessidade de alterar o código implementado e compilá-lo.

A substituição do sensor de temperatura (LM35) pelo sensor interno do processador trará economia na montagem do protótipo, leituras mais exatas e menos ruidosas.

Por fim, pensando em um produto real, há a necessidade de estudar a durabilidade da pastilha quando a mesma é integrada ao controlador, pois para atingir o valor de *setpoint* é necessário alterar muitas vezes o valor de tensão aplicada à pastilha de Peltier e não há estudos de como ela se comportará ao longo do tempo.

REFERÊNCIAS

- ADRIEL KOTVISKI. **TecMundo**, 2009. Disponível em: <https://www.tecmundo.com.br/hardware/1766-processador-em-chamas-.htm>. Acesso em: 10 jul. 2022.
- AGUIRRE, L. A. **Introdução à identificação de sistemas**. 4. ed. Belo Horizonte: Editora da UFMG, 2007.
- BOLTON, W. **Engenharia de Controle**; tradução Valcere Vieira Rocha e Silva; revisão técnica Antônio Pertence Junior: São Paulo: Editora Makron Books, 1995.
- BRITNEY. **Defirnitec**, 2022. Disponível em: <https://definirtec.com/dissipador-de-calor/>. Acesso em: 11 jul. 2022.
- SALUTES, B. **Canaltech**, 2021. Disponível em: <https://canaltech.com.br/hardware/quais-sao-as-principais-diferencas-entre-coolers/>. Acesso em: 11 jul. 2022.
- CALLEN, H. **Thermodynamics and introduction to thermostatics**. New York: John Wiley Sons, 1985.
- CHAPMAN, S.J. **Programação em MATLAB para engenheiros**. Tradução técnica: Flávio Soares Correa da Silva, São Paulo, Thomson Learning, 2006.
- COHEN, G. H., & Coon, G. **Theoretical consideration of retarded contro**, Transactions of the American Society of Mechanical Engineers, v. 75, n. 5, p 827-834, 1953.
- COUGHANOWR, D.R.; Koppel, L. B. **Análise e Controle de de Processos**. Editora Guanabara S.A, Rio de Janeiro ,1986.
- DANVIC. **Equipamentos Danvic**. 2016. Disponível em: <http://www.peltier.com.br/>. Acesso em: 2 out. 2022.
- DORF, R. C & Bishop, R. H. **Sistemas de Controle Moderno**. 2001. Addison Wesley Longman, 8ª Edição, 2001.
- LAGHARI, Dr. J. A. **Tune PID Controller using PID Tuner in MATLAB SIMULINK** , 2022. Disponível em: <https://www.mathworks.com/matlabcentral/fileexchange/93830-tune-pid-controller-using-pid-tuner-in-matlab-simulink/>. Acesso em: 01 dez. 2022.
- EQUIPE EDITORIAL DE CONCEITO. **Conceito de**. 2021. Disponível em: <https://conceito.de/calor>. Acesso em: 6 ago. 2022.
- GUGGENHEIM, E.A. **Thermodynamics. An Advanced Treatment for Chemists and Physicists**, fifth revised edition, North-Holland Publishing Company, Amsterdam, page 157.

GASPAR, A. **Física**. São Paulo, v.2, Ática, 2002.

HELERBROCK, Rafael. **Condução térmica**. 2022. Disponível em: <https://brasilescola.uol.com.br/fisica/conducao-termica.htm>. Acesso em 14 de setembro de 2022.

Nagrath, M. Gopal J. , **“Control System Engineering”**, New Age International Publications, 3rd Edition, 2002.

LJUNG, Lennart. **System identification toolbox: User's guide**. Natick, MA, USA: MathWorks Incorporated, 1995.

MAIA, M.R. Controlador PID-AA: **Desenvolvimento de Novas Técnicas de Pré-ajuste e de Compensação de Atraso de Transporte**. 1993, 120 f. Dissertação (Mestrado) -Curso de Engenharia Elétrica, UFSC, Florianópolis, 1993.

MATHWORKS. **PID Tuning Algorithm. Switzerland: The MathWorks, Inc.** 2022. Disponível em: <https://ch.mathworks.com/help/control/getstart/pid-tuning-algorithm.html>. Acesso em: 1 dez. 2022.

MINNICH, A.; DRESSELHAUS, M.; REN, Z.; CHEN, G. **Bulk nanostructure thermoelectric materials**: Current research and future prospects. *Energy Environmental Science*, v. 2, 2009.

MOHAPATRA, P.K.J., Mandal, P. and Bora, M.C. **Introduction to System Dynamics Modelling**, Universities Press of India, Hyderabad, 1994.

MUNDO POSITIVO. **Mundo Positivo**, 2021. Disponível em: <https://www.meupositivo.com.br/doseu jeito/tecnologia/o-que-e-processador/>. Acesso em: 10 jul. 2022.

NEIL, Kuyvenhove. **PID Tuning Methods An Automatic PID Tuning Study with MathCad**, Calvin College ENGR. 19 dez, 2002.

PROJETO DE ENERGIA TERMOAMBIENTE. **Projeto de energia termoambiente**. 2022. Disponível em: <https://www.energiatermoambiente.com.br/projeto/201602/resumo-efeito-seebeck-efeito-termoeletrico-efeito-peltier>. Acesso em: 17 jul. 2022.

SANTOS, A. V. dos. **As Pastilhas Termopar Nas Aulas Experimentais de Termologia**. 2016. 106 f. Dissertação (Mestrado) - Curso de Física, UFF. Volta Redonda: 2016.

SILVA, Rodolfo Pacheco. **Introdução à Identificação de Sistemas utilizando o System Identification Toolbox do MATLAB em conjunto com o Arduino para o Laboratório de Controle Linear**. 2015. 60 f. Trabalho de conclusão de curso (Bacharelado) –Engenharia Elétrica, Universidade Estadual Paulista, Guaratinguetá, 2015.

VAN WYLEN, G.J.; SONNTAG, R.E.; BORGNAKKE, C. **Fundamentos da termodinâmica**. 5. ed. São Paulo: Edgard Blucher Ltda., 1998. 537p.

WELTY, J. R. et al. **Fundamentals of Momentum, Heat, and Mass Transfer**. 5. ed.
Danvers: John Wile & Sons, 2008.

APÊNDICES

APÊNDICE A – Código desenvolvido para coleta de dados.

```
#define pin_lm35 A1 //Pino o qual o sensor está conectado
int saidaPWM = 3; //Pino conectado à saída do PWM
int numeroLeituras = 0;
float temperatura = 0;
void setup() {
  Serial.begin(9600); // Inicia a comunicação com
  pinMode(pin_lm35, INPUT); // Define o sensor como uma entrada de sinal
  pinMode(saidaPWM, OUTPUT); //Define o PWM como uma saída de sinal
}

void loop() {
  delay(10000); //Aguarda 10 segundos para iniciar
  while(numeroLeituras <= 255){ //loop para captura dos valores
    temperatura = 0;
    for(int i=0;i<10;i++){ //loop para calcular a média da medida
      float valor_analog_lm35 = float(analogRead(pin_lm35)); // Obtém o valor analógico que
      varia de 0 a 1023
      float tensao = (valor_analog_lm35 * 5) / 1023; // Converter esse valor para tensão elétrica
      temperatura = temperatura + (tensao / 0.010); // Dividimos a tensão por 0.010 que representa
      os 10 mV
    }
    Serial.print("A temperatura é ");
    Serial.println(temperatura/10); // Mostra na serial a temperatura do sensor
    Serial.print("para um PWM de: ");
    Serial.println(numeroLeituras); // Mostra na serial o numero da medida que é igual ao valor
    do pwm
    analogWrite(saidaPWM,numeroLeituras); //atualiza valor PWM
    delay(10000); // aguarda 10 segundos
    numeroLeituras++;
  }
}
```

```
analogWrite(saidaPWM,90); //Atualiza valor PWM para não ficar no máximo e dar gelo  
}
```

APÊNDICE B – Código desenvolvido para implementação do controlador PID.

```

// Declaração das pinagem
#define pwm 3
#define pin_lm35 A1

// Declaração das variáveis para controle da malha
float out = 0.0; // Temperatura atual
float setpoint = 0.0; // Valor do setpoint
float control = 0.0; // Sinal de controle em PWM
float error = 0.0; // Sinal de erro
float lastError = 0.0; // Erro calculado no momento anterior
float cumError = 0.0; // Integral do erro
float rateError = 0.0; // Derivada do erro

// Ganhos calculados a partir do método adotado no projeto
float K[3] = {5.734, 0.024, 0.0};
void setup()
{
  Serial.begin(9600);
  pinMode(pin_lm35, INPUT);
  pinMode(pwm, OUTPUT);
}
void loop()
{
  setpoint = 50; //Define o valor de setpoint
  out = obterTemperatura();
  error = out-setpoint; // Sinal de erro
  cumError += error;
  rateError = (error-lastError);
  lastError = error;
  control = ((K[0])*error + (K[1])*cumError + (K[2])*rateError);

```

```
if(control > 255.0){
control = 255.0;
}
else if(control < 0.0){
control = 0.0;
}
analogWrite(pwm,control); //Escreve na saída o novo valor para o PWM
delay(10);
Serial.print(control);
Serial.println(out);
}
float obterTemperatura(){
float valor_analog_lm35 = float(analogRead(pin_lm35)); // Obtém o valor analógico que
varia de 0 a 1023
float tensao = (valor_analog_lm35 * 5) / 1023; // Converte o valor para tensão elétrica
float temperatura = tensao / 0.010; // Dividi tensão por 0.010 que representa os 10 mV
return temperatura;
}
```