



## **ThermalPy: Um kit de baixo custo de código aberto em Python para controle e modelagem de processos térmicos.**

DOI: 10.37702/2175-957X.COBENGE.2024.5212

**Autores:** HENRIQUE OLIVEIRA BALIEIRO PESSOA, YURI HENRIQUE REZENDE, HUGO SAKAI IDAGAWA

**Resumo:** *ThermalPy é um sistema operacional multiplataforma projetado para estudar processos térmicos. As opções tradicionais de software e hardware nesse campo costumam ser caras e grande. O ThermalPy aborda isso usando a linguagem de script Python de código aberto e o hardware de código aberto Arduino para oferecer controle preciso de temperatura (45-75°C) e simulação térmica a um preço acessível. O hardware é composto por um Arduino Nano, gerenciado por uma interface Python desenvolvida com tkinter (GUI) e matplotlib, capaz de ajustar e exibir dados. Este kit versátil oferece um design consistente, proporcionando inúmeras aplicações, como incubadoras, estufas e fornos de tratamento térmico, além de ser adaptável para diversos outros usos.*

**Palavras-chave:** Baixo custo, Código aberto, Python, Processos térmicos.

# THERMALPY: UM KIT DE BAIXO CUSTO DE CÓDIGO ABERTO EM PYTHON PARA ENSINO DE CONTROLE E MODELAGEM DE PROCESSO TÉRMICO

## 1 INTRODUÇÃO

Nos últimos anos, o conceito da Indústria 4.0, caracterizada por inovação tecnológica, automação e processos avançados, tem recebido significativa atenção (SAP, 2024). No entanto, a utilização de processos térmicos, que tem sido fundamental para a civilização humana por milênios, continua sendo um aspecto fundamental da indústria moderna e da vida diária. Apesar de sua importância, o acesso à educação e recursos em processos térmicos, especialmente em técnicas avançadas de simulação, pode ser desafiador (SHERNOFF et al, 2017). Vias tradicionais como programas de pós-graduação frequentemente apresentam barreiras como altos custos, equipamentos volumosos e acesso limitado a software de simulação sofisticado. Este trabalho embarca em uma missão para democratizar o acesso à educação em processos térmicos e ferramentas de simulação. Após identificar a falta de software de código aberto, multiplataforma e acessível para simulação térmica de alta precisão, o ThermalPy foi desenvolvido.

Esta solução de código aberto baseada em Python foi especificamente projetada para fornecer aos educadores e alunos uma plataforma amigável para aprendizado, experimentação e simulação em processos térmicos. Ao introduzir o ThermalPy, este trabalho tem como objetivo capacitar alunos e profissionais a explorarem e compreender os processos térmicos com maior profundidade e precisão. Nossa ferramenta não apenas facilita a aprendizagem prática, mas também fomenta a inovação e experimentação em diversos campos.

Por fim, ao reduzir as barreiras de acesso e aprimorar a qualidade da educação (FERREIRA, 2016) em processos térmicos, este trabalho vislumbra um futuro em que indivíduos de diversos contextos possam contribuir e se beneficiar dos avanços na tecnologia térmica.

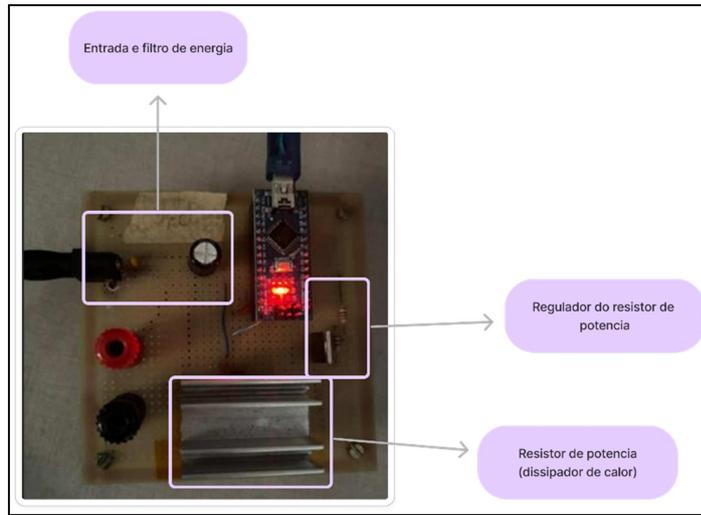
## 2 DESENVOLVIMENTO

O ThermalPy é uma ferramenta poderosa para controlar e simular processos térmicos. O sistema utiliza o Arduino Nano funciona como o controlador central do sistema (Figura 1), coletando dados do sensor de temperatura LM35DZ, conectado aos pinos digitais do Arduino, para medir com precisão a temperatura do dissipador. O Arduino envia sinais de controle para o atuador (resistor de potência) com o objetivo de ajustar a temperatura conforme os parâmetros definidos pelo usuário. O Arduino é responsável tanto por receber quanto enviar dados, que incluem a temperatura e o tempo em segundos, além dos valores de KP, KI e KD para a implementação do controle PID (CARDOZO, 2017).

Após receber esses dados, o resistor de potência aquece o dissipador, enquanto o sensor de temperatura coleta medições a cada 100 milissegundos. Esses dados são enviados de volta para o software, que os processa e apresenta gráficos detalhados, permitindo ao usuário analisar e compreender o comportamento térmico do sistema.

O software, implementado em Python com uma interface amigável, apresenta gráficos detalhados ao usuário para análise e compreensão dos resultados.

Figura 1 – Protótipo do kit.

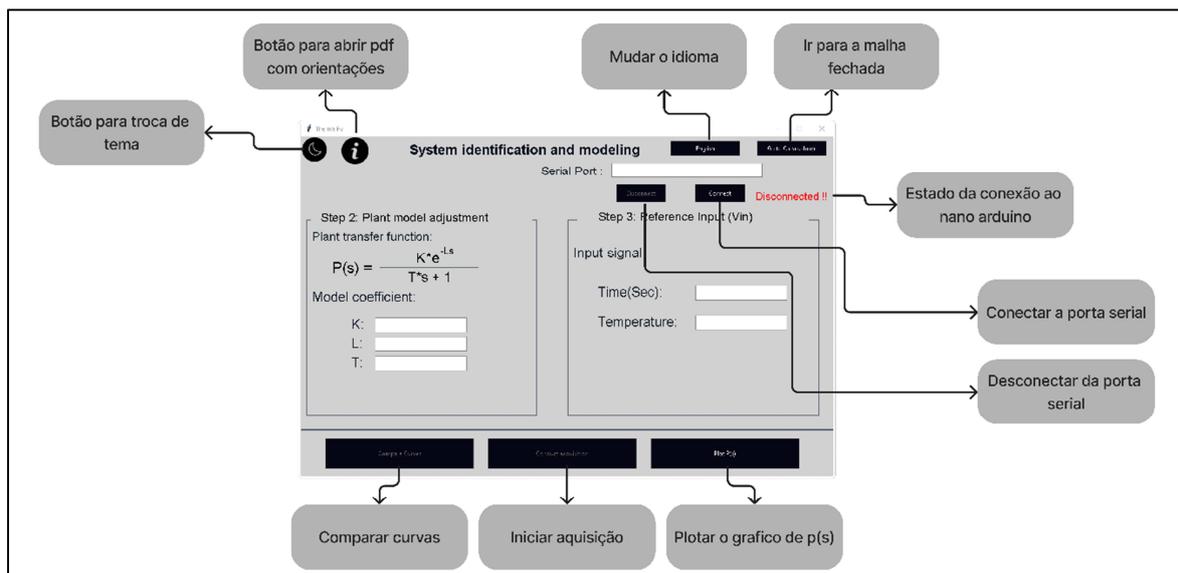


Fonte: autoria própria.

## 2.1 Interface do usuário

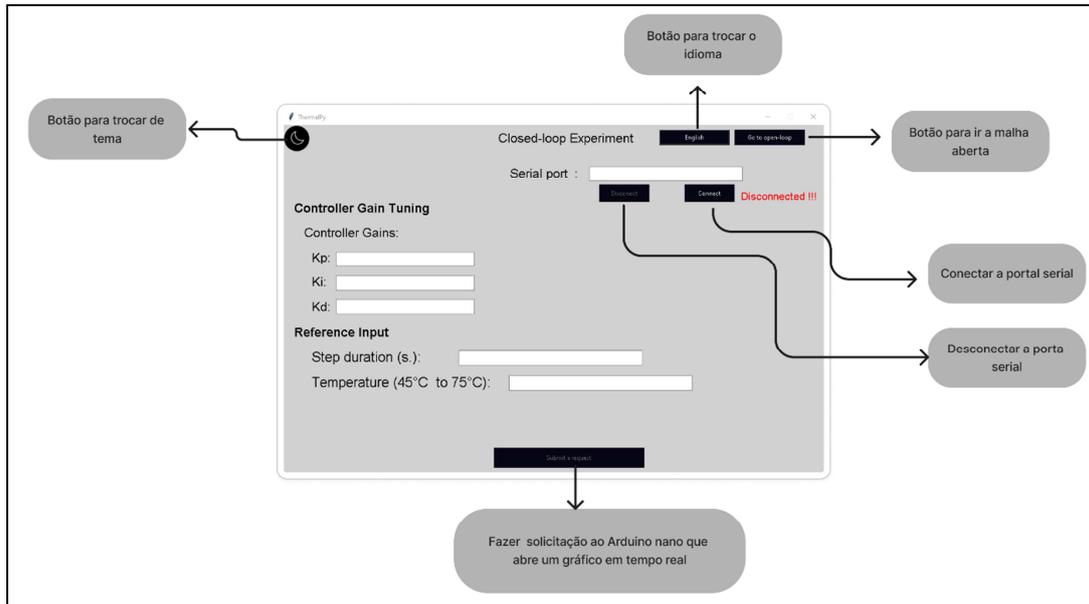
A linguagem de programação utilizada foi o Python 3.11 (MCKINNEY, 2024), com GUIs baseadas no Tkinter. O Tkinter é um módulo padrão do Python para desenvolvimento de GUI, popularmente usado em aplicações de desktop. Ele oferece uma ampla gama de widgets que são dispostos em uma janela raiz, permitindo a execução multiplataforma no Windows 10 ou posterior, bem como nos sistemas operacionais Linux. A programação desta aplicação é feita com duas janelas principais "identificação e modelagem do sistema" (Figura 2) e "experimentos em malha fechada" (Figura 3).

Figura 2 – Orientações da interface Malha Aberta.



Fonte: autoria própria.

Figura 3 – Orientações da interface Malha Fechada.



Fonte: autoria própria.

## 2.2 Descrição da identificação e modelagem do sistema

A interface principal do ThermalPy oferece funcionalidades cruciais tanto para controle em malha fechada quanto em malha aberta. Na seção dedicada à Identificação e Modelagem do Sistema, os usuários ajustam os parâmetros que descrevem o comportamento térmico do processo, permitindo uma personalização detalhada do modelo da planta térmica em controle. Além disso, a entrada de referência ("Temperature") permite que os usuários definam os valores desejados de temperatura que o sistema tentará alcançar e manter durante o experimento. Essas configurações são fundamentais para analisar como o ThermalPy responde a diferentes entradas e condições experimentais, proporcionando uma plataforma robusta para estudos em engenharia térmica. A interface também inclui botões de acessibilidade para alternar entre temas e selecionar idioma entre português e inglês (Figura 4), garantindo uma experiência de usuário mais intuitiva e acessível.

Adicionalmente, localizado no canto superior direito, há um botão que permite aos usuários alternarem para o modo em malha fechada. Dentro desta janela principal, os usuários podem estabelecer conexões com um Arduino Nano e ajustar as configurações do sistema.

A modelagem matemática do sistema é baseada na expressão:

$$P(s) = \frac{K * e^{ls}}{Ts + 1} \quad (1)$$

A Equação 1 representa uma função de transferência genérica para modelagem e análise de sistemas de controle de primeira ordem com atraso de tempo (FOPDT – "First Order Plus Dead Time"). Neste contexto:

- K representa o ganho do sistema, determinando a magnitude da resposta do sistema a uma entrada dada.
- L representa o atraso de tempo no sistema, indicando o tempo necessário para uma resposta completa a uma mudança na entrada.

- T é a constante de tempo do sistema, determinando quão rapidamente o sistema responde a uma mudança na entrada.

Para realizar esse cálculo, a entrada do usuário é recebida e passada para uma função chamada FOPDT, que manipula os dados e retorna um vetor usado para plotar um gráfico interativo. Durante a entrada de referência (aquisição no Arduino), os usuários podem inserir valores de temperatura e tempo para aquisição. Algumas aquisições podem levar horas, potencialmente atingindo 100°C (recomendado 75°C). Uma barra de progresso indica o andamento da aquisição (Figura 4), permitindo que os usuários continuem usando o software simultaneamente. Após a conclusão da aquisição, a barra de progresso desaparece, revelando o gráfico que exibe tempo de aquisição, temperatura e duas linhas estabilizadas (tracejadas) (Figura 6).

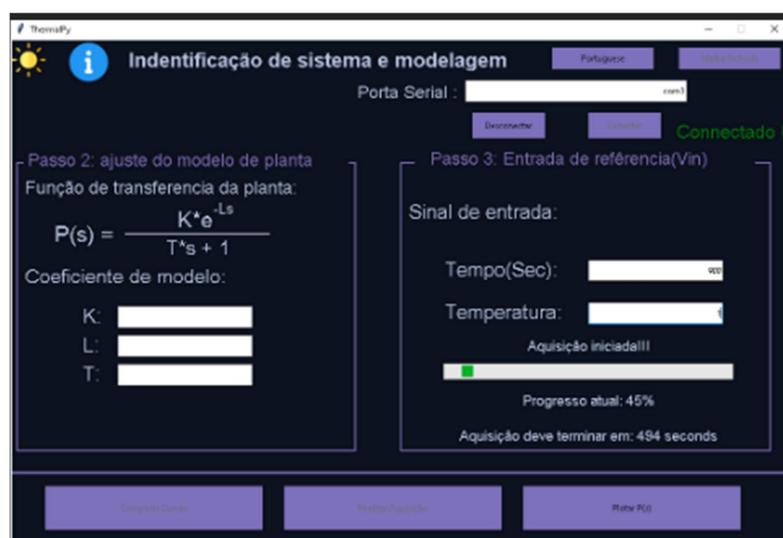
Este gráfico é interativo, permitindo que os usuários cliquem e usem comandos do teclado. Clicar com o botão esquerdo exibe a temperatura e o tempo do ponto clicado, enquanto clicar com o botão direito desenha uma linha tracejada até o ponto clicado, criando uma reta tangente, a qual é utilizada para calibrar o modelo conforme a Equação 1. Se essa tangente interceptar um ponto estabilizado, um ponto vermelho marca a interseção exata com os dados de temperatura e tempo. Os usuários podem pressionar a tecla de retrocesso para excluir o último ponto tangente, garantindo clareza no gráfico.

Além disso, o software oferece uma função de "temperatura suavizada" para reduzir o ruído no gráfico. Ao pressionar a tecla "r", o gráfico transita para uma versão com significativamente menos ruído, alcançado através de suavização exponencial "Função 2".

$$Smoothed(t) = \alpha \times Observation(t) + (1 - \alpha) \times Smoothed(t - 1) \quad (2)$$

Por fim, os usuários podem comparar curvas após adquirir e ajustar o sistema. A função de comparação de curvas consolida dois gráficos (modelo de ajuste e aquisição) na mesma tela e eixo, permitindo análise e comparação detalhadas de diferentes curvas.

Figura 4 – Interface com tema escuro.



Fonte: autoria própria.

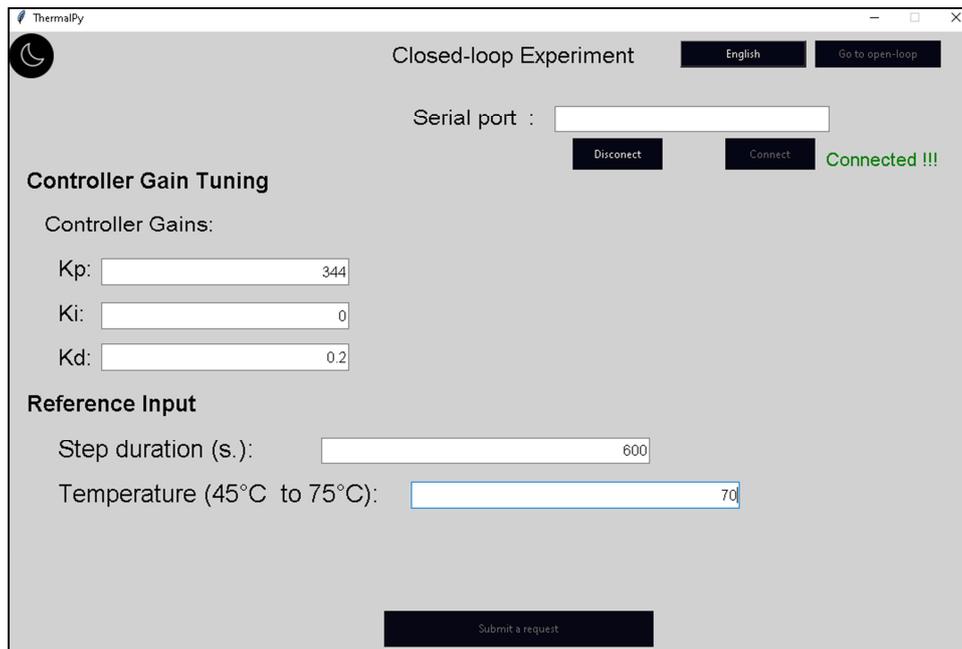
No sistema em malha fechada, existem os mesmos dois botões de acessibilidade mencionados anteriormente, juntamente com um botão para retornar à tela de identificação e modelagem do sistema (malha aberta).

Em um sistema de controle em malha fechada, a saída do sistema é comparada com a entrada de referência, e o erro resultante é utilizado para ajustar o sistema e minimizar as discrepâncias. Antecipamos a recepção de valores para  $k_p$ ,  $k_i$  e  $k_d$ , além do tempo e da temperatura de referência desejada (variando de 45°C a 75°C), conforme apresentado na Figura 5.

A temperatura representa a variável de processo controlada pelo sistema. No controle de temperatura em malha fechada, a temperatura real, medida pelo sensor, é comparada com a temperatura desejada (setpoint). Qualquer disparidade entre eles é utilizada para regular o sistema. Por exemplo, se a temperatura medida estiver abaixo da temperatura desejada, o controlador aumenta a potência do aquecedor para elevar a temperatura. Por outro lado, se ultrapassar o nível desejado, o controlador pode reduzir a potência do aquecedor para diminuí-la.

Após a entrada do usuário desses valores, uma tela exibe dois gráficos em tempo real ao lado do Arduino Nano. O primeiro gráfico apresenta a temperatura medida em função do tempo, enquanto o segundo ilustra o ciclo de trabalho versus o tempo (esforço de controle do PID). Esse gráfico de resposta permite ao usuário caracterizar a qualidade do projeto do controlador (tempo de resposta, erro estacionário, saturação, etc).

Figura 5 – Valores na Interface malha fechada.



Fonte: autoria própria.

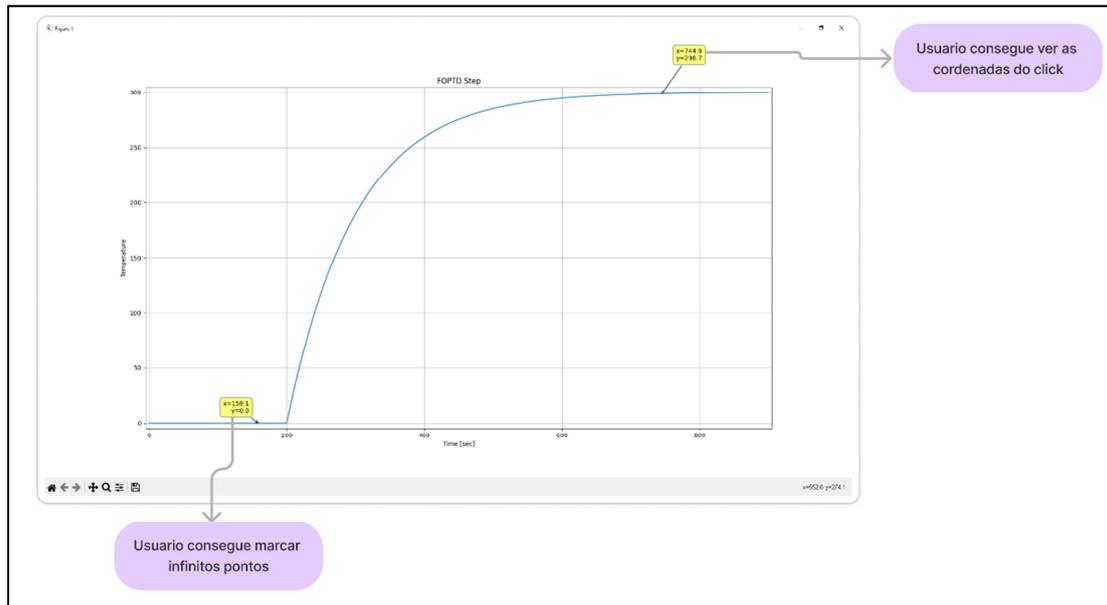
### 2.3 Visualização dos dados

A visualização de dados foi realizada utilizando a biblioteca Matplotlib (HUNTER, 2024), complementada pelo mpcursors para melhor interatividade. Além disso, o NumPy (OLIPHANT, 2024) foi utilizado para manipulação de expressões, incluindo algumas funções personalizadas desenvolvidas no projeto para simplificar o processo e oferecer uma visualização precisa de dados e também para uma compreensão fácil. As Figuras 6 a 9 apresentam exemplos de gráficos de visualização que o ThermalPy é capaz de gerar. A Figura 6 corresponde a um gráfico de resposta do modelo (FOPDT) com os

parâmetros de ajuste determinados pelo usuário, enquanto que a Figura 7 apresenta uma resposta de aquisição do sistema também em malha aberta. Esse último gráfico é útil ao usuário, pois permite que os parâmetros de modelo da planta sejam identificados utilizando-se o método da reta tangente de Ziegler-Nichols (1942). Após a identificação do modelo da planta, o resultado teórico e o experimental pode ser comparado (Figura 9).

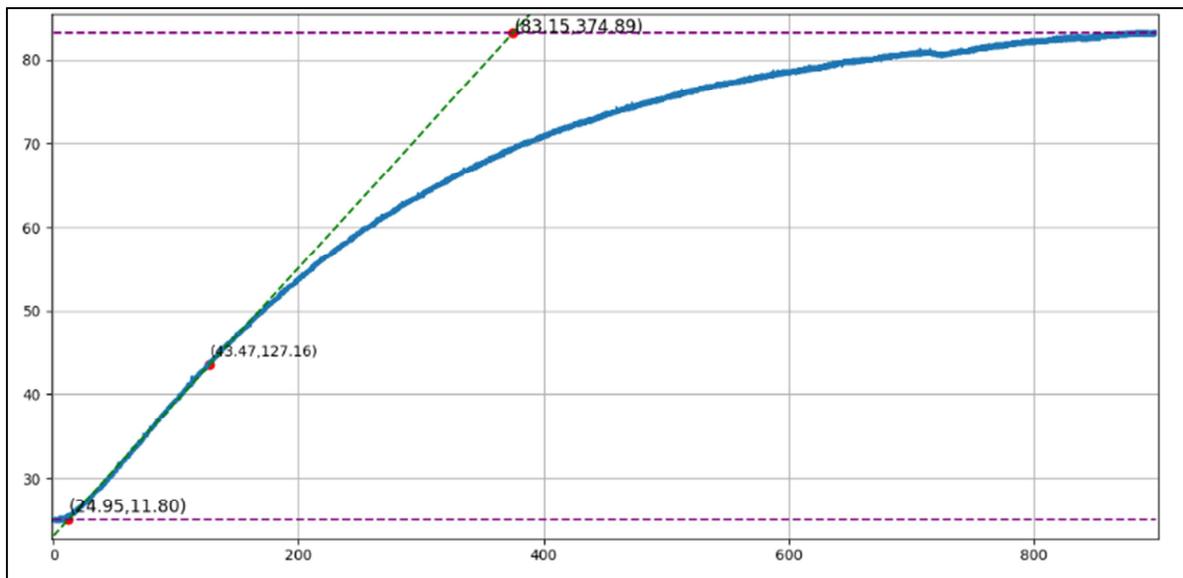
Finalmente, a Figura 8 apresenta a resposta do sistema em malha fechada após a calibração dos ganhos do controlador PID. Nessa etapa, diferentes métodos de projeto de controladores podem ser aplicados e as características de resposta podem ser determinadas por esse gráfico.

Figura 6 – Gráfico de resposta do modelo da planta.



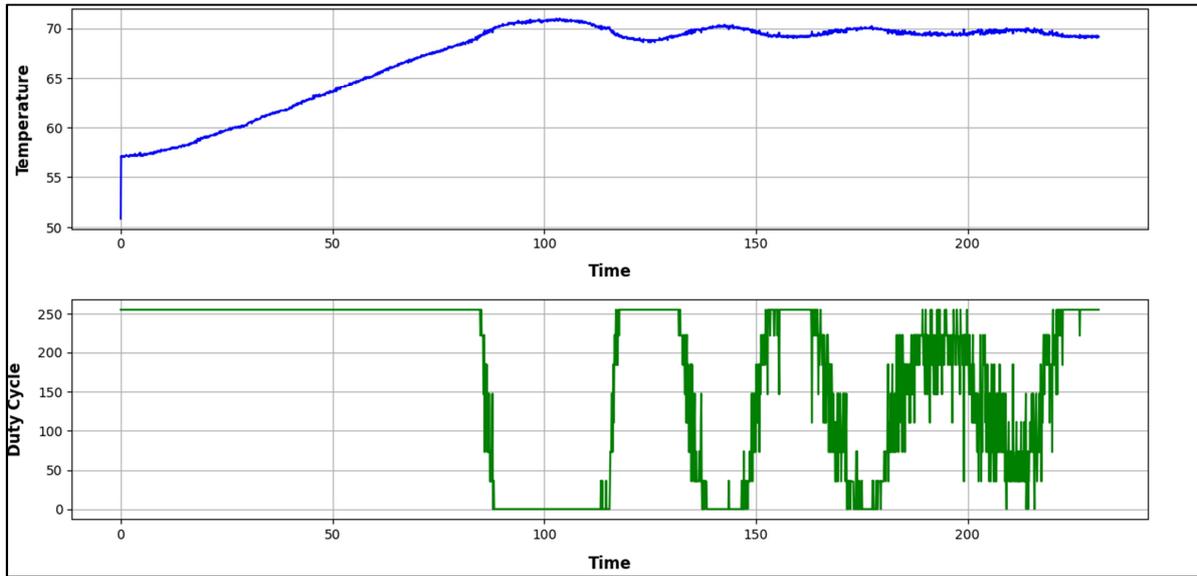
Fonte: autoria própria.

Figura 7 – Gráfico de resposta em malha aberta com estaque para seleção do ponto tangente à curva utilizada para calibrar o modelo teórico.



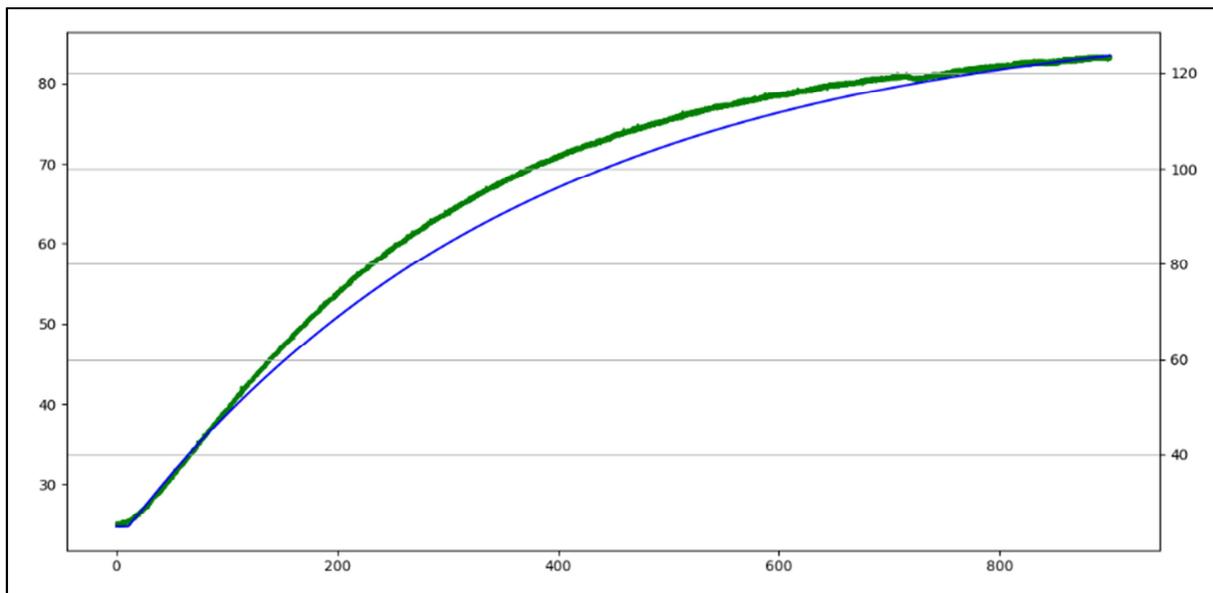
Fonte: autoria própria.

Figura 8 – Gráfico de resposta em malha fechada (temperatura em azul e o esforço de controle em verde) para uma temperatura de referência em 70°C.



Fonte: autoria própria.

Figura 9 – Gráfico de malha aberta comparando-se a resposta experimental (verde) e a teórica (azul).



Fonte: autoria própria.

## 2.4 Validação e caracterização

Para garantir a eficácia do ThermalPy como uma ferramenta educacional, foi desenvolvido um conjunto abrangente de procedimentos experimentais e metodológicos. O desenvolvimento do ThermalPy começou com a identificação dos requisitos educacionais específicos para o ensino de processos térmicos. Após essa fase, procedeu-se para a implementação técnica, onde foi utilizado o microcontrolador Arduino Nano como núcleo e desenvolvida uma interface de usuário em Python, usando a biblioteca Tkinter para garantir uma experiência de usuário intuitiva.

Os experimentos foram realizados em duas etapas principais: a identificação e modelagem do sistema em malha aberta e a execução de experimentos em malha fechada. Na fase de malha aberta, os usuários puderam ajustar parâmetros como o

ganho do sistema (K), o atraso de tempo (L) e a constante de tempo (T) através de uma interface interativa. A aquisição de dados foi feita conectando-se o Arduino Nano ao sistema, onde os usuários inseriram valores de temperatura e tempo para monitorar as respostas térmicas. Durante a aquisição, uma barra de progresso foi exibida, permitindo que os usuários acompanhassem o desenvolvimento do experimento em tempo real.

Para a malha fechada, é possível configurar o sistema para controlar a temperatura automaticamente. Os valores de KP, KI e KD foram ajustados com base nas respostas do sistema, e gráficos em tempo real mostraram a temperatura e o ciclo de trabalho. Esse processo permitiu aos usuários entenderem como o sistema responde a diferentes entradas e como ajustar os parâmetros de controle para alcançar o desempenho desejado.

Para validar o gráfico e os dados, foram conduzidas numerosas aquisições ao longo de um período de aproximadamente 900 segundos (15 minutos), durante o qual a temperatura alcançou picos de cerca de 90°C. Isso foi feito para confirmar a precisão dos valores. Posteriormente, após o ajuste do modelo, é possível plotar o gráfico e comparar as curvas. Ao comparar as curvas teóricas e experimentais, observou-se que o software demonstrou uma precisão notável, com desvios mínimos entre os dados simulados e os reais, comprovando a eficácia do ThermalPy em representar com exatidão o comportamento do sistema térmico.

Além dos testes, coletou-se um feedback qualitativo por meio de entrevistas. Os estudantes valorizaram a interatividade do ThermalPy e a interface amigável, destacando a visualização de dados em tempo real como intuitiva e eficaz. O professor relatou que a ferramenta trará mais dinamismo para as aulas, sendo facilmente integrada ao currículo devido à sua interface de fácil compreensão e à abrangência dos tópicos de controle térmico, adequando-se tanto a cursos de pós-graduação quanto de graduação. Muitos alunos afirmaram que a visualização gráfica das respostas do sistema facilitou a compreensão dos conceitos teóricos, tornando a aprendizagem mais prática e envolvente.

## 2.5 Aplicações potenciais e possíveis Atualizações

O ThermalPy é uma ferramenta versátil com aplicações potenciais em diversos contextos industriais, como fornos industriais, estufas e incubadoras. Além disso, pode ser utilizado em ambientes residenciais para aplicações como fornos ou micro-ondas. No entanto, seu foco principal está nos sistemas educacionais, experimentos de laboratório.

Para atender às necessidades emergentes da educação em engenharia e melhorar continuamente a funcionalidade do ThermalPy, várias atualizações estão planejadas:

- Melhoria no Design para Tema Escuro: aprimorar o design do tema escuro para proporcionar uma experiência mais esteticamente agradável e amigável ao usuário, garantindo visibilidade e conforto ótimos.
- Integração de Recursos Educacionais: adicionar opções dentro do software para acessar recursos educacionais, como documentos em PDF com descrições detalhadas dos recursos do ThermalPy e dos processos térmicos, permitindo aos usuários aprofundarem seus conhecimentos.
- Expansão para Outras Distribuições Linux: estender a compatibilidade do ThermalPy para outras distribuições Linux, como Arch Linux e Fedora, para torná-lo acessível a uma base de usuários mais ampla e garantir compatibilidade com diferentes sistemas operacionais.
- Melhoria de Desempenho: melhorar continuamente o desempenho do ThermalPy para otimizar sua velocidade, responsividade e eficiência, garantindo operação

suave e experiência do usuário sem problemas, mesmo ao lidar com tarefas complexas ou conjuntos de dados grandes.

Recomenda-se também realizar estudos longitudinais para avaliar o impacto do ThermalPy no aprendizado ao longo do tempo, comparando seu desempenho com outras ferramentas educacionais disponíveis, como o OpenTCC (SÁNCHEZ, 2020). Incorporar um sistema de feedback contínuo dos usuários pode oferecer insights valiosos para ajustes imediatos na interface e funcionalidades do ThermalPy, melhorando ainda mais sua eficácia como ferramenta de ensino em engenharia térmica. Ao implementar essas atualizações, o ThermalPy visa aprimorar sua versatilidade, acessibilidade e usabilidade, capacitando assim os usuários em diversos ambientes educacionais, científicos e industriais a utilizar efetivamente as capacidades de controle e monitoramento térmico para suas necessidades e aplicações específicas.

## 2.6 Impacto

O ThermalPy é uma resposta direta às necessidades educacionais contemporâneas no campo da engenharia térmica, proporcionando uma plataforma versátil e acessível para a exploração prática de processos térmicos. Em um cenário onde a integração de tecnologia e prática é essencial, o ThermalPy se destaca ao oferecer uma interface intuitiva que facilita a compreensão dos conceitos complexos de controle térmico.

Atendendo às demandas atuais por aprendizado prático e aplicado, o ThermalPy não só fortalece a base teórica dos estudantes, mas também os capacita com habilidades práticas essenciais para o mercado de trabalho. A capacidade de simular e controlar processos térmicos em tempo real enriquece a experiência educacional, preparando os alunos para enfrentar desafios reais em ambientes industriais.

Para atender às necessidades emergentes da educação em engenharia, o ThermalPy está continuamente evoluindo. Planos futuros incluem melhorias na interface do usuário para torná-la ainda mais acessível e adaptável a diferentes contextos educacionais. Além disso, a expansão das capacidades do ThermalPy para suportar novas tecnologias e plataformas de hardware garantirá que ele continue relevante e eficaz no ensino de engenharia térmica, promovendo um aprendizado mais dinâmico e alinhado com as demandas industriais atuais e futuras.

O ThermalPy é um software de código aberto, de baixo custo e multiplataforma, projetado para simulações de alta precisão. Seu design versátil visa aprimorar a educação, especialmente em programas de pós-graduação. Usuários proficientes em inglês ou português podem facilmente aprender sobre processos térmicos, que são integrais para a vida diária. Além disso, o ThermalPy pode contribuir significativamente para avanços na pesquisa.

O objetivo foi desenvolver um kit (composto por software e hardware) que seja portátil e possa ser usado com PCs básicos e consumo mínimo de energia. Isso garante acessibilidade e facilidade de uso para pesquisadores e educadores, maximizando assim seu impacto em diversos campos.

## 3 CONCLUSÃO

Em conclusão, este trabalho visa democratizar o ensino de processos térmicos, disponibilizando um sistema de software e hardware versátil capaz de simular e monitorar o aquecimento em ambientes controlados. Utilizando o ThermalPy, um software de código aberto multiplataforma, este sistema oferece recursos avançados de visualização e

análise de dados em tempo real. Com um compromisso de oferecer o menor custo possível, o ThermalPy busca tornar acessível a estudantes e instituições educacionais ferramentas de alta tecnologia para experimentação e aprendizagem prática. Ao facilitar o acesso a ferramentas de alta tecnologia, este projeto não apenas melhora a compreensão teórica, mas também promove uma prática experimental robusta e eficaz. Espera-se que o ThermalPy contribua significativamente para o avanço no ensino e na pesquisa em engenharia térmica, promovendo uma aprendizagem mais dinâmica e acessível para uma ampla gama de usuários.

#### 4 REFERÊNCIAS

CHRIS, Liechti. **Pyserial's**. Disponível em:

<https://pyserial.readthedocs.io/en/latest/pyserial.html/>. Acesso em: 23 jan. 2024.

CARDOZO, Florisvaldo. **Modelagem de Funções de Transferência de Plantas Industriais em Malha**. Disponível em:

<https://repositorio.ufu.br/bitstream/123456789/20410/3/ModelagemFuncoesTransferencia.pdf>. Acesso em: 22 dez. 2023.

FERREIRA, Vinicius Vidal; JÓSE, Francisco Gomes. Plataforma gráfica interativa para educação em controle. In: XLIV Congresso Brasileiro de Educação em Engenharia, 2016. **Anais**. Disponível em:

<https://www.abenge.org.br/cobenge/legado/arquivos/3/anais/anais/159722.pdf>. Acesso em: 27 jan. 2024.

HUNTER, John; DROETTBOOM, Michael. **Matplotlib**. Disponível em:

<https://matplotlib.org/>. Acesso em: 22 dez. 2023.

MCKINNEY, Wes. **Python for data analysis: data wrangling with pandas, numpy, and ipython**. 2. ed, Sebastopol: O'Reilly Media, 2017.

SAP. **what is industry 4.0**. Disponível em:

<https://www.sap.com/products/scm/industry-4-0/what-is-industry-4-0.html>. Acesso em: 26 fev. 2024.

SÁNCHEZ, Carlos. **OpenTCC: An open source low-cost temperature-control chamber**, [S.L.], v. 7, abril.2020. HardwareX (Elsevier).

<https://doi.org/10.1016/j.ohx.2020.e00099>.

SHERNOFF, D.J.; SINHA, S.; BRESSLER, D.M.; GINSBURG, L. **Assessing teacher education and professional development needs for the implementation of integrated approaches to STEM education**. International Journal of STEM Education, v.4, n.13, 2017.

TRAVIS, Oliphant. **Numpy**. Disponível em:

<https://numpy.org/>. Acesso em: 10 nov. 2023.

ZIEGLER, J. G.; NICHOLS, N. B. et al. **Optimum settings for automatic controllers**.trans. ASME,v. 64, n. 11, 1942.

## THERMALPY: A LOW-COST, OPEN-SOURCE PYTHON KIT FOR CONTROLLING AND MODELING THERMAL PROCESSES

**Abstract:** *ThermalPy is a cross-platform operating system designed for studying thermal processes. Traditional software and hardware options in this field are often expensive and large. ThermalPy addresses this by using open-source Python scripting language and Arduino's open-source hardware to deliver precise temperature control (45-75°C) and thermal simulation at an affordable price point. The hardware comprises an Arduino nano, managed through a Python interface developed with tkinter (GUI) and matplotlib, which is able to adjust and display data. This versatile kit boasts a consistent design, offering numerous applications such as egg incubators, greenhouses, and heat treatment furnaces, while also being adaptable for various other uses.*

**Keywords:** *low-cost, open-source, Tkinter, Python, cross-plataform*

