

Simulador Didático de um Controle PID em Python para Aplicações de Engenharia de Automação

Wagner Guimarães dos Santos Pinto

Instituto Federal São Paulo (IFSP), Cubatão, São Paulo, Brasil

Dr. Marcelo Saraiva Coelho

Instituto Federal São Paulo (IFSP), Cubatão, São Paulo, Brasil

Resumo: O presente trabalho apresenta o desenvolvimento de um simulador didático de controle PID (Proporcional, Integral e Derivativo) aplicado a um sistema térmico, implementado integralmente em linguagem Python com interface gráfica desenvolvida em *tkinter*. O projeto tem como objetivo proporcionar uma ferramenta interativa para o estudo prático de conceitos de controle automático, integrando conhecimentos de modelagem matemática, programação e análise de sistemas dinâmicos. O simulador permite ajustar os parâmetros de controle K_p , K_i e K_d e visualizar, em tempo real, a resposta do sistema frente a variações e perturbações, promovendo a compreensão dos efeitos de estabilidade, tempo de resposta e erro estacionário. A interface amigável e o uso de bibliotecas científicas, como *NumPy* e *Matplotlib*, tornam a ferramenta acessível tanto para fins acadêmicos quanto profissionais. Os resultados obtidos demonstraram o comportamento esperado do controlador PID em diferentes configurações, validando sua aplicabilidade como instrumento de apoio ao ensino e à experimentação em Engenharia de Controle e Automação.

Palavras-chave: Controle PID. Python. Didático. Simulação.

Abstract: This paper presents the development of a didactic PID (Proportional, Integral, and Derivative) control simulator applied to a thermal system, fully implemented in Python programming language with a graphical interface built using *tkinter*. The project aims to provide an interactive tool for the practical study of automatic control concepts, integrating mathematical modeling, programming, and dynamic systems analysis. The simulator allows users to adjust control parameters (K_p , K_i , and K_d) and visualize, in real time, the system's response to variations and

disturbances, facilitating understanding of stability, response time, and steady-state error. The user-friendly interface and the use of scientific libraries such as *NumPy* and *Matplotlib* make the tool accessible for both academic and professional purposes. The results confirmed the expected behavior of the PID controller under different configurations, validating its effectiveness as a teaching and experimentation aid in Control and Automation Engineering.

Keywords: *PID Control. Python. Didactic. Simulation.*

1. INTRODUÇÃO

No contexto da Indústria 4.0, os sistemas automatizados e de controle de processos tornaram-se fundamentais para o aumento da eficiência e da confiabilidade em ambientes industriais. Essa transformação tecnológica exige profissionais com sólida base em Engenharia de Controle e Automação, capazes de compreender, modelar e ajustar sistemas de controle em diferentes tipos de processos dinâmicos. Entre as estratégias mais amplamente utilizadas, o controlador PID (Proporcional, Integral e Derivativo) destaca-se por sua simplicidade, robustez e aplicabilidade em uma ampla gama de sistemas físicos e digitais (Ogata, 2010; Nise, 2011).

Entretanto, o estudo de teoria de controle no ensino superior, frequentemente centrado em formulações matemáticas e análises em papel, pode limitar a compreensão intuitiva sobre o efeito das ações proporcional, integral e derivativa no comportamento dinâmico do sistema. Pesquisas em educação em engenharia indicam que abordagens práticas e de aprendizagem ativa, em que o estudante interage com o conteúdo por meio de simulações e experimentações, promovem maior retenção de conhecimento e melhor desempenho em disciplinas de ciências, tecnologia e matemática (Freeman *et al.*, 2014; National Research Council, 2000).

Nos últimos anos, o avanço das ferramentas de simulação computacional e dos laboratórios virtuais tem ampliado o acesso a ambientes de aprendizagem baseados em experimentação digital. Essas ferramentas permitem a observação de fenômenos

complexos sem os custos e riscos associados a experimentos físicos, além de possibilitar repetibilidade e controle preciso dos parâmetros (Simulation In Engineering Education, 2024). Em especial, os simuladores interativos têm demonstrado potencial pedagógico no ensino de controle de processos, permitindo que o estudante modifique parâmetros em tempo real e observe as respostas do sistema, desenvolvendo assim uma compreensão mais profunda dos conceitos envolvidos (Cooper, 2015; Yao *et al.*, 2017).

Nesse contexto, a adoção de linguagens modernas, acessíveis e amplamente utilizadas em ambientes acadêmicos, como o Python, tem contribuído significativamente para o ensino de engenharia. Sua sintaxe simples, a ampla disponibilidade de bibliotecas científicas e a facilidade de integração com interfaces gráficas tornam-no ideal para o desenvolvimento de simuladores didáticos (Negaban, 2024). O uso de Python em aplicações educacionais favorece não apenas a aprendizagem de controle, mas também o desenvolvimento de competências em programação e análise de dados, alinhando-se às demandas atuais da formação em engenharia.

Dessa forma, o presente trabalho apresenta o desenvolvimento de um simulador didático de controle PID de temperatura, implementado em Python com interface gráfica e visualização em tempo real, como ferramenta de apoio ao ensino de controle e automação. O objetivo é fornecer uma plataforma acessível, modificável e visualmente intuitiva, capaz de aproximar a teoria da prática e incentivar a autonomia e o raciocínio crítico dos estudantes. Assim, o projeto busca contribuir para a melhoria da formação prática e interdisciplinar no ensino superior em Engenharia de Controle e Automação.

2. OBJETIVOS

O objetivo deste artigo é o desenvolvimento de um simulador para aplicações didáticas e interativas de um controlador PID, utilizando como foco as simulações de um sistema térmico de primeira ordem, estruturado na linguagem de programação Python por ser acessível e de fácil incrementação de funções e mudanças de parâmetros. Os diversos recursos que a ferramenta apresenta a tornaram atrativa

também por possuir uma interface gráfica intuitiva que permite ao usuário a visualização em tempo real do comportamento do sistema e a resposta ao sinal de controle. A proposta é focada em criar uma ferramenta que auxilia a formação de novos engenheiros de controle e automação e entendimento mais aprofundado das disciplinas correlatas com a proposta do artigo.

A aplicação prática dos conceitos aprendidos em aula, e o entendimento do papel desempenhado por cada variável do modelo matemático de controle reforçam o aprendizado e fixação das técnicas de forma dinâmica. O simulador projetado visa a aplicação em tempo real de ajustes nos parâmetros K_p , K_d e K_i , evidenciando ao usuário a resposta imediata das variações no ambiente simulado por meio da plotagem gráfica, identificando o tempo de acomodação, *overshoot*, estabilidade e erro em regime permanente.

Apesar da finalidade de aplicação técnica dos conhecimentos, o projeto também apresenta uma abordagem pedagógica, propondo o uso de recursos gratuitos e acessíveis no âmbito doméstico de cada aluno, como também dentro das dependências da instituição de ensino que ele é aplicado. Deste modo, o simulador PID pode auxiliar professores e alunos nas interações em salas de aula ao abordar os temas de controle e automação, integrando de forma prática, simples e rápida a teoria exposta em aula.

3. Fundamentação Teórica

Em frente às novas necessidades que surgem com os avanços da indústria, modelos mais rápidos e precisos se fazem mais necessários com o passar do tempo. Neste contexto, o controlador PID permanece como um dos métodos de controle mais utilizados nos ambientes industriais, laboratórios e educacionais, e grande parte disto se explica por sua simplicidade, aplicabilidade universal e robustez (Ogata,2010).

O funcionamento do controlador PID consiste no ajuste contínuo da variável de controle por meio da diferença entre o valor medido e a referência (*setpoint*), de modo a minimizar o erro. O termo proporcional (P) do controlador atua no erro instantâneo, enquanto a parcela integral (I) realiza o acúmulo do erro durante o processo, para eliminar os desvios apresentados, já a parcela derivativa (D) prevê variações

abruptas, suavizando a resposta. Ao configurar as três variáveis de forma adequada, permite ao sistema atingir a estabilidade, rapidez e precisão esperadas (Nise, 2011).

Pesquisas recentes desenvolvidas por docentes do Instituto Federal de São Paulo também exploram o avanço de técnicas de controle aplicadas a sistemas dinâmicos não lineares. Entre elas, destaca-se o trabalho de Carvalho Júnior *et al.* (2022), que implementaram um controlador por rede neural recorrente com neurônios paraconsistentes para o rastreamento de trajetória em um pêndulo invertido. Embora o enfoque seja distinto do controlador PID clássico abordado neste estudo, ambos compartilham a mesma base conceitual de controle automático e modelagem de sistemas dinâmicos. Assim, a pesquisa de Carvalho Júnior *et al.* (2022) evidencia a continuidade natural entre ferramentas didáticas, como o simulador desenvolvido neste trabalho, e aplicações mais complexas e auto adaptativas em engenharia de controle.

O modelo matemático escolhido foi o de primeira ordem com atraso, por sua ampla utilização nos estudos de controle de temperatura. A simulação do comportamento de plantas industriais deste tipo, apresentam atraso entre a ação de controle empregada e a resposta da variável medida. Por se tratar de um sistema simples, a lógica utilizada na programação consequentemente também se torna mais simples e objetiva para entendimento dos alunos.

Devido à versatilidade, o projeto foi totalmente implementado em Python (versão 3.12.5). Por ser hoje uma das linguagens mais utilizadas na engenharia, ele possibilita integrações entre modelos matemáticos, simulações e visualizações gráficas. O Python conta com uma abordagem de código aberto, que fortalece a comunidade dos usuários e permite um amplo desenvolvimento de ferramentas interativas e didáticas, estimulando a exploração da lógica aplicada no desenvolvimento, e a melhoria contínua da aplicação.

O Visual Studio Code (VSCode) foi a plataforma escolhida para a arquitetura do simulador pela facilidade de acesso dos professores e alunos do instituto por meio de licenças acadêmicas originais fornecidas, além de ser leve e conta com opções variadas de customizações e facilidades oferecidas ao desenvolvedor, como por exemplo para depuração e execução de código Python.

O projeto utiliza o Tkinter para a construção de uma interface gráfica interativa, permitindo a visualização dos parâmetros a serem inseridos na resposta do sistema conforme sua configuração. A biblioteca NumPy é empregada como ferramenta fundamental para o processamento numérico necessário ao desenvolvimento do modelo. Já a Matplotlib é utilizada para a plotagem dos gráficos, possibilitando a visualização da resposta temporal e do sinal de controle em tempo real.

3.1. MODELO MATEMÁTICO DO CONTROLADOR PID E DO SISTEMA TÉRMICO

Para o modelo matemático empregado no programa, foi utilizada a equação diferencial que relaciona as variáveis de entrada, saída e parâmetros do processo conforme na equação (1):

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

Onde:

- $u(t)$ é o sinal de controle aplicado ao sistema;
- $e(t) = r(t) - y(t)$ representa o erro entre o *setpoint* $r(t)$ e a saída medida $y(t)$;
- K_p , K_i e K_d ganhos proporcional, integral e derivativo, respectivamente.

Com o objetivo de simular uma planta industrial de forma simplificada, a equação para modelagem do sistema térmico estudado se dá pela equação (2) (Ogata, 2010):

$$T_{n+1} = T_n + (P_n - T_n * a) * dt * b \quad (2)$$

Onde ' a ' e ' b ' são constantes ajustadas empiricamente, e ' P_n ' representa a potência de aquecimento no instante ' n '.

Ambos os modelos quando empregados em conjunto na simulação, contribuem para formar uma base consistente de estudo e de análise experimental da estabilidade, erro e velocidade de resposta do sistema. Por meio desta abordagem, a observação de resultados realistas e de baixo custo computacional permitem uma compreensão intuitiva do efeito de cada parcela de controle.

4. Desenvolvimento

A estrutura do código do simulador PID foi planejada para aproximar a teoria da prática, facilitando o aprendizado do controle automático de forma acessível. A fundamentação teórica destaca como métodos interativos e o uso de simulações podem melhorar a compreensão dos estudantes em Engenharia de Controle e Automação. Pensando nisso, o código foi organizado em etapas que permitem ao usuário ajustar parâmetros, visualizar as respostas do sistema em tempo real e experimentar diferentes condições, colocando em prática conceitos como ações proporcionais, integrais, derivativas e resposta a distúrbios. O uso de Python e das bibliotecas Tkinter e Matplotlib torna o simulador visualmente intuitivo e fácil de ser manipulado em ambiente educacional.

Ao traduzir os conceitos teóricos para o funcionamento do simulador, o projeto garante que cada parte da experiência do usuário, desde o momento da entrada dos dados até a análise dos gráficos, reflita as etapas fundamentais do controle PID estudadas em aula. Assim, a estrutura do código apoia o desenvolvimento da autonomia e do raciocínio crítico, além de permitir novas explorações práticas com segurança e rapidez.

4.1. Descrição da Estrutura do Simulador

O funcionamento do simulador PID foi dividido em blocos bem definidos, seguindo uma sequência lógica que garante clareza durante o uso:

- a) **Inicialização:** O programa é iniciado com a exibição de uma interface gráfica amigável, onde o usuário encontra campos para preencher os parâmetros de controle (K_p , K_i , K_d), o valor de referência (*setpoint*) e a seleção do tipo de planta a ser simulada.
- b) **Entrada de Dados:** O usuário insere os valores desejados e aciona a simulação através de um botão. O software valida essa entrada, confirmando que os parâmetros são adequados para o experimento digital.

- c) **Execução da Simulação:** Com os dados validados, uma função central realiza um *loop* repetido para cada instante de tempo. Nesse processo, ocorre:
- Cálculo do erro como diferença entre *setpoint* e valor medido;
 - Cálculo das ações proporcional, integral e derivativa, conforme as fórmulas clássicas do PID;
 - Aplicação do sinal de controle ao modelo matemático da planta, atualizando a variável de processo de acordo com o tipo de sistema escolhido;
 - Inserção de possíveis perturbações, simulando eventos inesperados.
- d) **Registro dos Resultados:** O simulador armazena o histórico dos valores da variável de processo, dos sinais de controle e dos erros em listas, permitindo depois a análise completa do funcionamento do sistema.
- e) **Visualização Gráfica:** Ao fim da simulação, o sistema utiliza a biblioteca Matplotlib para mostrar gráficos interativos: um para o acompanhamento da variável de processo ao longo do tempo em relação ao *setpoint*, e outro para o sinal de controle. Esses gráficos ajudam o usuário a perceber diretamente como as mudanças nos parâmetros afetam o comportamento do sistema.
- f) **Interatividade Contínua:** Depois de analisar os resultados, o usuário pode modificar os parâmetros e iniciar novo ciclo de simulação, tornando o aprendizado exploratório e dinâmico, em sintonia com as premissas abordadas na fundamentação teórica sobre aprendizagem ativa e uso de simulações no ensino de engenharia.

Essa estrutura torna o simulador uma ferramenta didática flexível e alinhada com as demandas modernas de ensino, permitindo integrar teoria, prática e análise crítica em um único ambiente digital.

O núcleo do código do simulador é o conjunto de funções que realiza a simulação do controlador PID. Nela, o algoritmo calcula o erro, atualiza as ações proporcional, integral e derivativa, e modifica a variável de processo conforme o tipo de planta escolhido. Os ensaios foram realizados utilizando as três plantas disponíveis no

simulador: térmica, nível e motor, cada uma modelada matematicamente pela equação discreta correspondente no código, que define a evolução da variável de saída `temp[i]`. O trecho da Figura 1 resume este núcleo:

Figura 1 - Captura de tela do código de configuração dos parâmetros de planta e inserção de erro

```
14     for i in range(1, steps):
15         error = setpoint - temp[i - 1]
16         integral += error * dt
17         derivative = (error - last_error) / dt
18         last_error = error
19
20         power = kp * error + ki * integral + kd * derivative
21         power = max(0, min(power, 100)) # saturação
22
23         control[i] = power
24
25         # Simulação de diferentes plantas
26         if planta == "termica":
27             temp[i] = temp[i - 1] + (power - temp[i - 1] * 0.05) * dt * 0.1
28         elif planta == "nivel":
29             temp[i] = temp[i - 1] + (power - 0.1 * temp[i - 1]) * dt * 0.05
30         elif planta == "motor":
31             temp[i] = temp[i - 1] + (power - temp[i - 1] * 0.2) * dt * 0.07
32
33         # #1. Malha fechada com distúrbio
34         if i == 50:
35             temp[i] -= 10
36
37     return temp, control
```

Fonte: Autores (2025)

O coeficiente multiplicador final (0,1 para térmica, 0,05 para nível e 0,07 para motor) diferencia a velocidade de resposta em cada planta, sendo a térmica a mais rápida, o nível a mais lenta e o motor com resposta intermediária.

Durante os testes, os parâmetros do controlador PID — proporcional (K_p), integral (K_i) e derivativo (K_d) — foram ajustados para determinar seus efeitos sobre o comportamento dinâmico do sistema. Por exemplo, foi possível observar como a variação de K_p afeta a rapidez e a amplitude da resposta, enquanto o K_i altera a capacidade do controlador em eliminar o erro estacionário, e o K_d influencia na suavização das oscilações e na estabilidade.

Os testes foram conduzidos principalmente usando entradas em degrau para o *setpoint*, além de perturbações simuladas no sistema, que foram aplicadas para analisar a robustez do controlador. A manipulação desses parâmetros foi feita

diretamente na interface gráfica, permitindo a visualização em tempo real da resposta do sistema, o que favoreceu a experimentação ativa e o aprendizado prático do controle automático.

Dessa forma, a metodologia adotada no simulador integra a teoria de controle PID com a prática da sintonia e análise dos parâmetros, reforçando a compreensão sobre o impacto que cada ação do controlador tem no comportamento do sistema dinâmico.

O sistema completo do simulador PID é estruturado para oferecer ao usuário uma experiência didática e interativa de controle automático, replicando etapas essenciais de projetos industriais em ambiente virtual. Logo ao iniciar o programa, ocorre a configuração da interface gráfica com Tkinter, onde o usuário os valores dos parâmetros K_p , K_i , K_d do controlador PID, define o *setpoint* desejado e seleciona o tipo de planta a ser simulada entre as opções térmica, nível ou motor.

Ao pressionar o botão de execução, o programa verifica se os valores informados são válidos. Caso haja algum erro de digitação, a interface solicita a correção dos dados antes de prosseguir, garantindo robustez e segurança para a etapa de simulação.

Uma vez validados os parâmetros, é realizada a simulação propriamente dita. Internamente, o código entra em um *loop* de cálculo que representa o funcionamento do controlador PID em malha fechada. A cada ciclo, são computados o erro, os termos integral e derivativo, a respectiva ação proporcional, o sinal de controle gerado e a atualização da variável de estado da planta. Essa dinâmica é ajustada conforme o modelo matemático escolhido (térmica, nível ou motor), afetando diretamente a velocidade de resposta, com cada planta apresentando comportamento próprio determinado pelos coeficientes do código.

No meio da simulação, há uma perturbação programada no tempo $t=50$, que simula uma condição adversa real encontrada em processos industriais, permitindo ao usuário observar a atuação do controlador PID na correção do distúrbio.

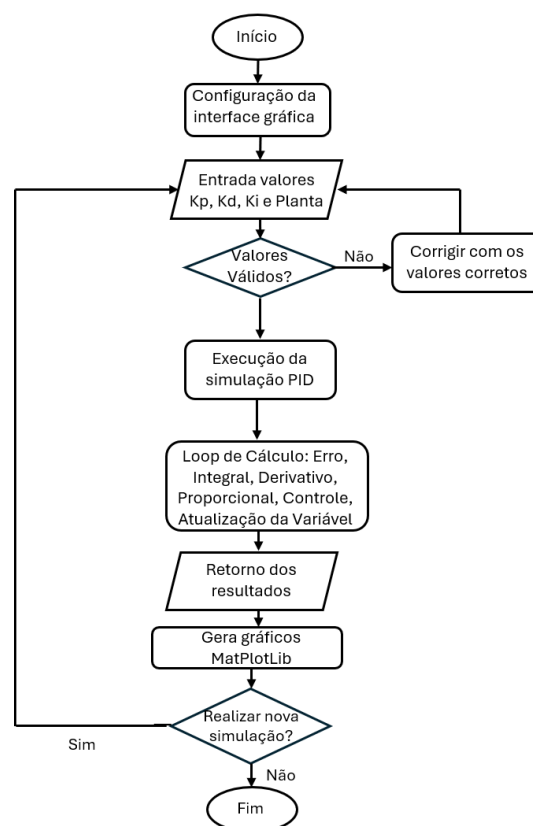
Após os cálculos, o simulador retorna os valores dos resultados e gera gráficos com matplotlib, apresentando a evolução da variável controlada (por exemplo, temperatura) e da potência aplicada pelo controlador ao longo do tempo. Esses dados

visuais permitem análise detalhada da resposta dinâmica do sistema frente às configurações adotadas.

Por fim, o usuário pode optar por realizar uma nova simulação com diferentes valores e plantas, repetindo o ciclo de maneira didática e flexível. Caso não deseje novos testes, o programa encerra sua execução.

O fluxograma apresentado na Figura 2 representa de forma visual e sequencial essas operações, facilitando a compreensão do funcionamento interno do simulador.

Figura 2 - Fluxograma do programa



Fonte: Autores (2025)

5. Resultados e Testes

Para observar as diferentes respostas do sistema para a entrada de controle, a tabela 1 demonstra as configurações dos parâmetros K_p , K_d e K_i utilizadas para cada teste realizado, sendo esperados o regime permanente, oscilatório e amortecido.

Cada uma das configurações demonstraram a estabilidade do sistema, a velocidade de resposta da planta ao sinal de controle, *overshoot* e o erro permanente.

Exemplos de Configuração:

Tabela 1 – Parâmetros de entrada do controlador PID simulado

Kp	Kd	Ki	Observação
2	0,1	1	Permanente
100	0,56	0	Oscilatório
1	0	5	Amortecido

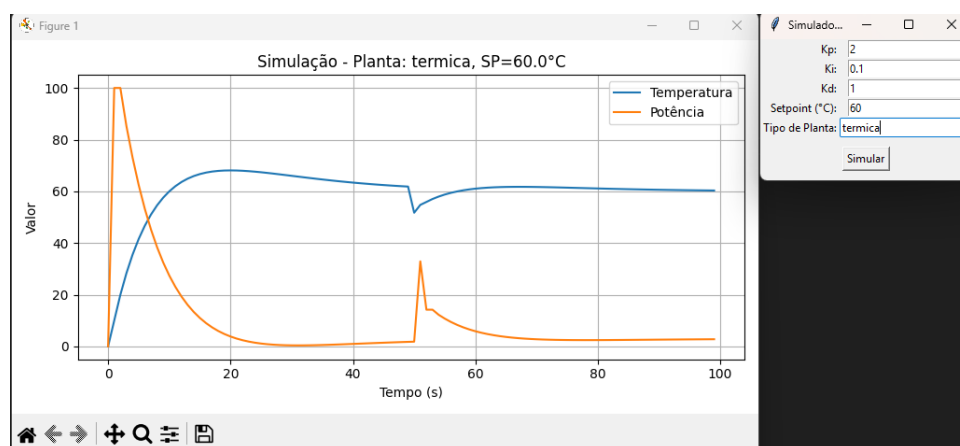
Fonte: Autores (2025)

Nas simulações realizadas, há a inserção de um distúrbio aos 50 segundos de cada uma delas para simular um ambiente de operação real, e acompanhar o sinal de controle PID corrigindo a resposta da planta ao ruído induzido.

5.1. Resposta do sistema em condição estável

Para a configuração estável, foi efetuado o teste conforme Figura 3:

Figura 3 - Resposta estável do sistema



Fonte: Autores (2025)

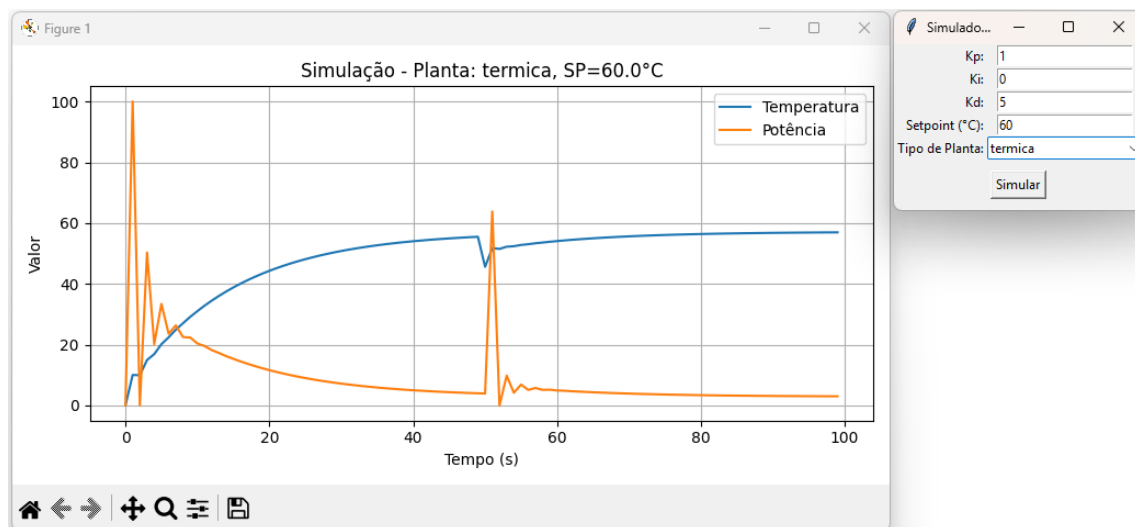
A resposta estável em sistemas controlados por PID é caracterizada pela aproximação ao *setpoint* de forma controlada, com oscilações reduzidas e erro praticamente nulo em regime permanente. Conforme Ogata (2010), essa condição é

obtida através de parâmetros bem ajustados, nos quais o ganho proporcional promove rapidez suficiente para atingir o valor desejado, enquanto as ações integral e derivativa atuam para eliminar desvios residuais e evitar oscilações acentuadas. Esse equilíbrio é observado experimentalmente nos testes realizados (ver Tabela 1 e Figura 2), reforçando a teoria sobre sintonia de sistemas de primeira ordem com atraso, como discutido por Ogata (2010) e Nise (2012).

5.2. Resposta do sistema em condição superamortecida

Segundo Nise (2012), um sistema superamortecido caracteriza-se por atingir o *setpoint* de forma lenta, mas estável, sem apresentar *overshoot*. Isto é alcançado com um ganho proporcional reduzido e uma ação derivativa mais destacada, configurando uma resposta robusta contra oscilações e ideal para processos em que a segurança operacional supera a necessidade de velocidade de resposta. A Figura 4 exemplifica esta configuração, bem como a resposta do sistema ao controle.

Figura 4 - Resposta Superamortecida do sistema



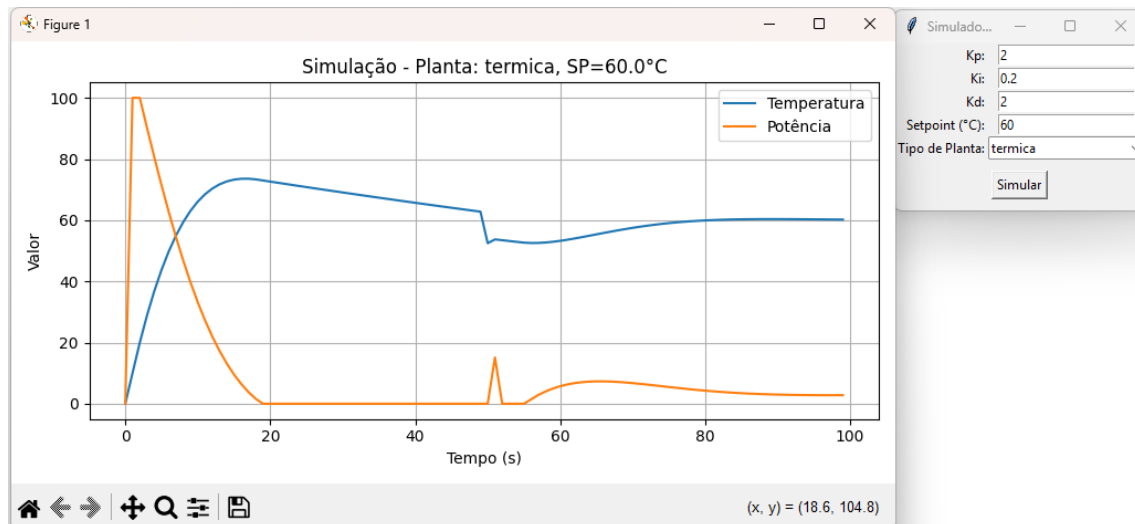
Fonte: Autores (2025)

Nos experimentos realizados, essa configuração aparece claramente, especialmente em processos de nível e temperatura, nos quais oscilações podem ser prejudiciais. O comportamento é ilustrado e detalhado na Figura 3.

5.3. Resposta do sistema em condição criticamente amortecida

A resposta criticamente amortecida é desejável quando se busca o compromisso entre tempo de acomodação curto e ausência de ultrapassagem expressiva do *setpoint*, conforme a Figura 5:

Figura 5 - Resposta criticamente amortecida do sistema



Fonte: Autores (2025)

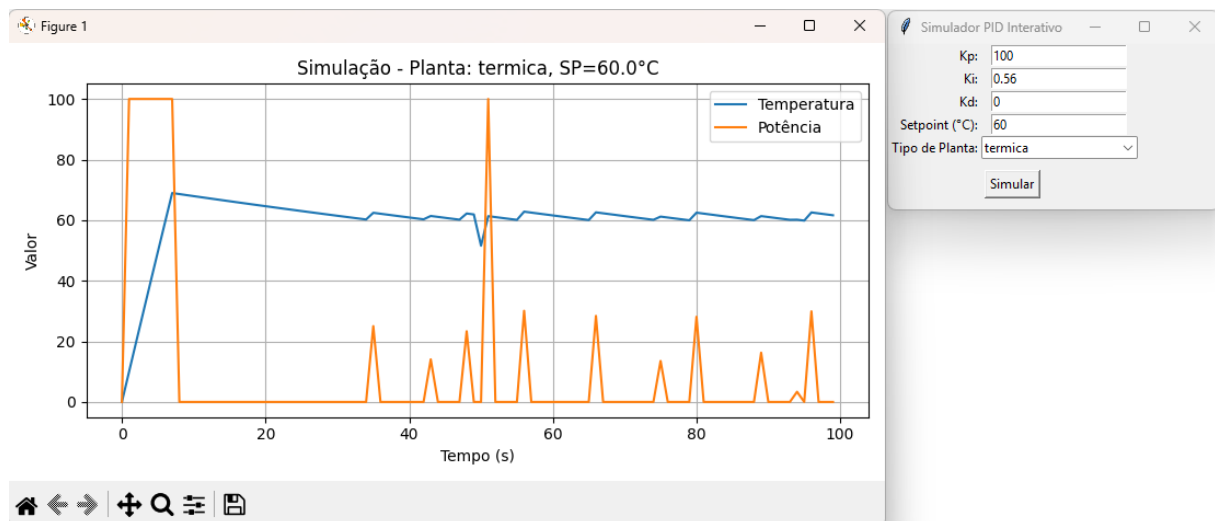
Ogata (2010) aponta que tal condição ocorre quando os parâmetros do controlador são ajustados para minimizar o valor do *overshoot*, atingindo rapidamente a referência desejada com estabilidade. Nos testes realizados, a sintonia dos parâmetros permitiu observar exatamente essa característica, confirmando a aplicabilidade das recomendações teóricas de Ogata (2010) para sistemas de controle realimentados.

5.4. Resposta do sistema em condição oscilatória

Nise (2012) e Ogata (2010) apontam que, ao configurar o ganho proporcional em um valor muito alto, enquanto se mantém o ganho derivativo e o integral muito baixos ou até nulos, é comum presenciar o surgimento de oscilações contínuas e até

crescentes no sistema. A simulação desta resposta oscilatória pode ser vista na Figura 6:

Figura 6 - Resposta Oscilatória do sistema



Fonte: Autores (2025)

Na prática, isso significa que o sistema perde sua capacidade de se estabilizar e passa a apresentar movimentos repetitivos e cada vez mais amplos em torno do *setpoint*. Apesar de ser uma situação indesejada para qualquer aplicação real, principalmente em ambientes industriais onde a estabilidade é crítica, esse fenômeno é extremamente importante do ponto de vista didático. Ele permite compreender, de forma visual e direta, como a sintonia dos parâmetros do PID influencia o equilíbrio do sistema e evidencia a fronteira entre um controle seguro e uma resposta instável. Ao experimentar o simulador, torna-se fácil perceber esse comportamento: basta aumentar o K_p e não compensar isso com ações integral ou derivativa para que rapidamente surjam oscilações visíveis no resultado, conforme Figura 5.

6. Conclusão

O projeto cumpriu com sucesso os objetivos propostos, fornecendo uma plataforma prática para o estudo de controladores PID aplicados ao controle térmico. A utilização de Python e VSCode demonstrou ser eficaz e acessível, proporcionando ao aluno experiência com ferramentas profissionais e com o processo de desenvolvimento de soluções técnicas completas.

A integração entre modelagem, programação e análise visual reforça a importância da interdisciplinaridade na engenharia moderna, promovendo o raciocínio crítico e a autonomia no aprendizado.

O simulador desenvolvido, baseado em modelos térmico, de nível e motora, pode futuramente ser ampliado para incluir sistemas de natureza biológica, aproximando-se de aplicações reais como as apresentadas por Coelho (2022), que desenvolveu um controlador adaptativo automatizado para a regulação da pressão arterial média em situações hipotensivas com uso de fenilefrina. Enquanto o presente trabalho tem caráter didático, voltado à compreensão prática dos princípios do controle PID, o estudo de Coelho (2022) demonstra o potencial desses mesmos fundamentos quando aplicados a sistemas fisiológicos de alta criticidade. Dessa forma, a integração de um modelo matemático de controle da pressão arterial ao simulador representaria uma evolução natural do projeto, ampliando seu valor educacional e interdisciplinar ao conectar a engenharia de controle às aplicações biomédicas.

Apesar dos resultados positivos, o sistema apresenta limitações, que abrem caminhos para futuros aprimoramentos. Entre elas, destaca-se a necessidade de ampliar a variedade de plantas simuladas para incluir modelos mais complexos e realistas, a implementação de interfaces mais intuitivas e responsivas, e a incorporação de módulos que permitam avaliações quantitativas do desempenho e acompanhamento da evolução dos alunos durante o processo de aprendizado. Além disso, a inclusão de estudos empíricos envolvendo aplicação em salas de aula, com coleta de dados qualitativos e quantitativos sobre o engajamento e aprendizagem dos estudantes, enriqueceria substancialmente a validade e o impacto do simulador.

REFERÊNCIAS

- CARVALHO JÚNIOR, A. de; ANGÉLICO, B. A.; JUSTO FILHO, J. F.; OLIVEIRA, A. M. de; SILVA FILHO, J. I. da. **Model reference control by recurrent neural network built with paraconsistent neurons for trajectory tracking of a rotary inverted pendulum**, Applied Soft Computing, 2022, 109927, ISSN 1568-4946, DOI: 10.1016/j.asoc.2022.109927.
- COELHO, M. S. **Controlador adaptativo automatizado para pressão arterial média em situações hipotensivas com uso de fenilefrina**. Tese de Doutorado — Universidade de Mogi das Cruzes, Mogi das Cruzes, 2022.
- COOPER, D. **An Interactive Simulator for Process Control Education**. *International Journal of Engineering Education*, v. 31, n. 5, p. 1350–1357, 2015.
- FREEMAN, S.; EDDY, S. L.; MCDONOUGH, M.; SMITH, M. K.; OKOROAFOR, N.; JORDT, H.; WENDEROTH, M. P. **Active Learning Increases Student Performance in Science, Engineering, and Mathematics**. *Proceedings of the National Academy of Sciences*, v. 111, n. 23, p. 8410–8415, 2014.
- NATIONAL RESEARCH COUNCIL. **How People Learn: Brain, Mind, Experience, and School** – Expanded Edition. Washington, D.C.: The National Academies Press, 2000.
- NEGABAN, A. **Simulation in engineering education: the transition from physical experimentation to digital immersive simulated environments**. *Simulation*, v. 100, n. 7, p. 621–633, jul. 2024.
- NISE, N. S. **Engenharia de sistemas de controle**. 6 ed. São Paulo: Editora LTC, 2012.
- OGATA, K. **Engenharia de controle moderno**. 5. ed. São Paulo: Prentice Hall, 2010.
- YAO, J.; LIMBERIS, L.; WILLIAMS, R.; HOWARD, E. **An Efficient PID Control Teaching Module with LabVIEW Simulation**. *International Journal of Engineering Pedagogy*, v. 7, n. 4, p. 64–71, 2017.