

ANÁLISE NUMÉRICA DO COMPORTAMENTO DINÂMICO DE SISTEMAS COM PID: UMA ABORDAGEM COM PYTHON CONTROL SYSTEMS LIBRARY

Vitor Amadeu Souza¹

¹Centro Universitário de Volta Redonda – UniFOA, Volta Redonda, Brasil

Autor Correspondente: vitor.amadeu@foa.org.br

RESUMO

Este trabalho apresenta uma análise computacional detalhada do comportamento de controladores PID (Proporcional-Integral-Derivativo) através de simulação numérica, investigando os efeitos individuais e combinados dos parâmetros K_p , K_i e K_d no desempenho de sistemas de controle em malha fechada. A metodologia empregada utiliza a biblioteca Python Control Systems para simular um sistema de segunda ordem representativo de processos industriais comuns, analisando métricas de desempenho como *overshoot*, tempo de acomodação e erro em regime permanente. Os resultados demonstram que o aumento do ganho proporcional K_p reduz o tempo de resposta mas aumenta o *overshoot*, enquanto o termo integral K_i elimina o erro em regime permanente ao custo de maior oscilação. O termo derivativo K_d mostrou-se eficaz na redução do *overshoot* e melhoria da estabilidade relativa do sistema. A análise paramétrica revelou interações complexas entre os três termos, evidenciando a necessidade de ajuste simultâneo para otimização do desempenho. Os resultados obtidos corroboram a teoria clássica de controle e fornecem insights práticos para o projeto de controladores PID em aplicações industriais, demonstrando a importância da simulação computacional como ferramenta de análise e projeto em engenharia de controle.

Palavras-chave: controlador PID; simulação numérica; análise paramétrica; sistemas de controle; Python.

ABSTRACT

This work presents a detailed computational analysis of the behavior of PID (Proportional-Integral-Derivative) controllers through numerical simulation, investigating the individual and combined effects of the parameters K_p , K_i , and

Kd on the performance of closed-loop control systems. The methodology employs the Python Control Systems Library to simulate a second-order system representative of common industrial processes, analyzing performance metrics such as *overshoot*, settling time, and steady-state error. The results show that increasing the proportional gain K_p reduces response time but increases *overshoot*, while the integral term K_i eliminates steady-state error at the cost of increased oscillation. The derivative term K_d proved effective in reducing *overshoot* and improving the system's relative stability. The parametric analysis revealed complex interactions among the three terms, highlighting the need for simultaneous tuning to optimize performance. The obtained results corroborate classical control theory and provide practical insights for the design of PID controllers in industrial applications, demonstrating the importance of computational simulation as a tool for analysis and design in control engineering.

Keywords: PID controller; numerical simulation; parametric analysis; control systems; Python.

1 INTRODUÇÃO

Os controladores PID constituem uma das tecnologias mais amplamente utilizadas na indústria moderna, sendo responsáveis pelo controle de aproximadamente 95% dos processos industriais automatizados (Åström; Hägglund, 2006). Essa predominância deve-se à sua simplicidade conceitual, facilidade de implementação e eficácia em uma ampla gama de aplicações, desde controle de temperatura em fornos industriais até sistemas de posicionamento de alta precisão (Franklin; Powell; Emami-Naeini, 2019).

A estrutura fundamental do controlador PID combina três ações de controle distintas: proporcional (P), integral (I) e derivativa (D), cada uma contribuindo de forma específica para o desempenho global do sistema (Ogata, 2010). O termo proporcional fornece uma ação corretiva proporcional ao erro instantâneo, o termo integral elimina o erro em regime permanente através da acumulação temporal dos erros passados, e o termo derivativo antecipa tendências futuras do erro, melhorando a estabilidade e reduzindo o *overshoot* (Dorf; Bishop, 2017).

Apesar de sua aparente simplicidade, a sintonia adequada dos parâmetros PID representa um desafio na prática industrial. Métodos tradicionais como Ziegler-Nichols, embora amplamente utilizados, frequentemente resultam em desempenho subótimo, especialmente em sistemas com características não-lineares ou variantes no tempo (Rivera; Morari; Skogestad, 1986). Essa limitação tem motivado o desenvolvimento de técnicas avançadas de sintonia, incluindo métodos baseados em otimização computacional e inteligência artificial (Bennett, 1993).

A simulação computacional emergiu como uma ferramenta fundamental para o estudo e projeto de controladores PID, permitindo a análise

detalhada do comportamento do sistema sob diferentes condições operacionais sem os riscos e custos associados à experimentação em sistemas reais (Seborg; Edgar; Mellichamp, 2016). A disponibilidade de ferramentas computacionais poderosas, como MATLAB/Simulink e Python Control Systems Library, democratizou o acesso a técnicas sofisticadas de análise e projeto de sistemas de controle (Murray; Åström; Boyd, 2008), tendo como vantagem ao usar o ambiente Python como o proposto neste trabalho o fato de ele ser gratuito.

O ambiente Python, em particular, tem ganhado crescente popularidade na comunidade de engenharia de controle devido à sua natureza open-source, facilidade de uso e extenso ecossistema de bibliotecas científicas (Harris *et al.*, 2020). A biblioteca Python Control Systems, desenvolvida por Murray *et al.* (2021), oferece funcionalidades comparáveis ao MATLAB Control Systems Toolbox, incluindo análise de sistemas lineares, projeto de controladores e simulação temporal.

A análise paramétrica de controladores PID através de simulação computacional permite a compreensão profunda dos efeitos individuais e combinados dos parâmetros K_p , K_i e K_d no desempenho do sistema (Visioli, 2006). Essa abordagem sistemática é essencial para o desenvolvimento de intuição em engenharia de controle e para a formulação de estratégias eficazes de sintonia.

Estudos recentes têm demonstrado que a compreensão adequada dos trade-offs entre diferentes métricas de desempenho é essencial para o projeto eficaz de controladores PID (Åström; Hägglund, 2006). O *overshoot*, tempo de acomodação, tempo de subida e erro em regime permanente representam aspectos distintos do desempenho que frequentemente entram em conflito, exigindo decisões de compromisso informadas.

O presente trabalho contribui para essa área de conhecimento através de uma análise computacional sistemática do comportamento de controladores PID, utilizando simulação numérica para investigar os efeitos paramétricos em um sistema representativo. A metodologia empregada combina rigor científico com praticidade de implementação, fornecendo insights valiosos tanto para pesquisadores quanto para profissionais da área.

2. METODOLOGIA

2.1 Plataforma Computacional e Ferramentas de Simulação

A implementação da análise paramétrica foi realizada utilizando a linguagem de programação Python versão 3.8, escolhida por sua robustez, facilidade de uso e amplo suporte da comunidade científica (Van Rossum; Drake, 2009). O ambiente de desenvolvimento integrado Google Colab foi utilizado para facilitar a documentação e visualização dos resultados (Kluyver *et al.*, 2016).

As principais bibliotecas utilizadas incluem: NumPy versão 2.0.2 para operações matemáticas e manipulação de arrays (Harris *et al.*, 2020); Matplotlib versão 3.10.0 para visualização gráfica dos resultados (Hunter, 2007); e Python Control Systems Library versão 0.9.0 para simulação e análise de sistemas de controle lineares (Murray *et al.*, 2021).

2.2 Modelagem do Sistema de Controle

O sistema analisado consiste em uma planta de segunda ordem com função de transferência $G(s) = 1/(s^2 + 2s + 1)$, representativa de sistemas do tipo massa-mola-amortecedor comumente encontrados em aplicações industriais (Nise, 2015). Esta escolha foi motivada pela sua simplicidade matemática e capacidade de demonstrar claramente os efeitos dos parâmetros PID.

O controlador PID foi implementado na forma paralela padrão, com função de transferência $G_c(s) = K_p + K_i/s + K_d \cdot s$, onde K_p , K_i e K_d representam os ganhos proporcional, integral e derivativo, respectivamente (Åström; Hägglund, 2006). A configuração em malha fechada foi estabelecida através da conexão de realimentação unitária negativa entre a saída do sistema e a entrada de referência.

2.3 Parâmetros de Simulação

A análise temporal foi conduzida em um horizonte de 10 segundos com resolução de 0.01 segundos, resultando em 1000 pontos de amostragem. Esta resolução foi escolhida para garantir precisão adequada na captura das características dinâmicas do sistema, incluindo comportamentos transitórios rápidos (Franklin; Powell; Emami-Naeini, 2019).

Os valores dos parâmetros PID foram selecionados para cobrir uma faixa representativa de comportamentos do sistema: K_p variando entre 1, 5 e 10; K_i entre 0, 0.5 e 1; e K_d entre 0, 0.1 e 0.5. Esses valores foram escolhidos com base na experiência prática e literatura especializada para garantir estabilidade do sistema em malha fechada (Ziegler; Nichols, 1942).

2.4 Métricas de Desempenho

A avaliação quantitativa do desempenho foi baseada em métricas clássicas de sistemas de controle: *overshoot* percentual, definido como a máxima sobrepassagem em relação ao valor de referência; tempo de acomodação, calculado como o tempo necessário para que a resposta permaneça dentro de $\pm 2\%$ do valor final; tempo de subida, medido entre 10% e

90% do valor final; e erro em regime permanente, determinado como a diferença entre o valor de referência e o valor final da resposta (Ogata, 2010).

2.5 Metodologia de Análise Paramétrica

A análise foi estruturada em seis cenários distintos para investigar sistematicamente os efeitos individuais e combinados dos parâmetros PID. Cada cenário foi projetado para isolar e quantificar o efeito específico de um parâmetro enquanto mantém os demais constantes, permitindo a identificação clara de relações causa-efeito.

O primeiro cenário investiga o efeito isolado do ganho proporcional K_p , mantendo $K_i = 0,5$ e $K_d = 0,1$ constantes. O segundo cenário analisa o impacto do ganho integral K_i com $K_p = 5$ e $K_d = 0,1$ fixos. O terceiro cenário examina a influência do ganho derivativo K_d com $K_p = 5$ e $K_i = 0,5$ constantes.

Os cenários subsequentes exploram interações entre parâmetros: o quarto cenário analisa o efeito de K_i com K_p baixo ($K_p = 1$); o quinto investiga o impacto de K_d com K_p alto ($K_p = 10$); e o sexto examina a influência de K_d com K_i alto ($K_i = 1$).

2.6 Implementação Computacional

A implementação seguiu uma arquitetura modular, com funções específicas para simulação do sistema PID, cálculo de métricas de desempenho e visualização dos resultados. A função `simulate_pid()` utiliza a biblioteca Python Control Systems para criar a função de transferência do controlador PID e calcular a resposta em malha fechada através da função `ctrl.feedback()`.

A função `calculate_metrics()` implementa algoritmos para cálculo automático das métricas de desempenho, utilizando técnicas de processamento de sinais para identificação precisa dos pontos característicos da resposta temporal. Particular atenção foi dada à robustez dos algoritmos de detecção do tempo de acomodação, implementando verificação de janela deslizante para garantir que o sistema efetivamente se estabeleceu dentro da tolerância especificada.

A visualização dos resultados foi implementada utilizando a biblioteca Matplotlib, criando uma interface gráfica com seis subplots organizados em configuração 3×2 . Cada gráfico inclui múltiplas curvas de resposta temporal com legendas informativas contendo os valores das métricas de desempenho correspondentes.

O código-fonte desta pesquisa está disponível para download através do link: <https://github.com/vitor-souza-ime/pid>.

3. RESULTADOS E DISCUSSÃO

3.1 Efeito do Ganho Proporcional (K_p)

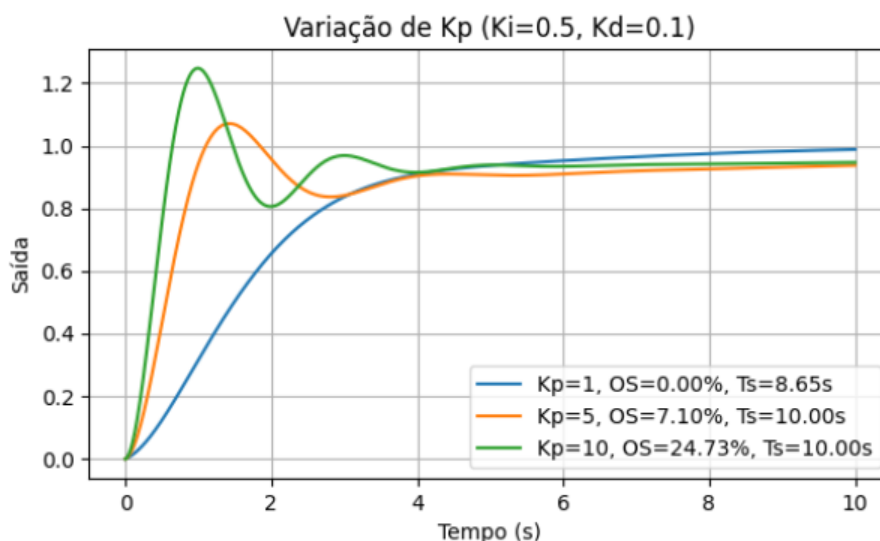
A análise do efeito do ganho proporcional revelou comportamentos consistentes com a teoria clássica de controle. O aumento de K_p de 1 para 10 resultou em redução do tempo de subida, demonstrando o efeito acelerador do termo proporcional na resposta do sistema. Especificamente, para $K_p = 1$, o sistema apresentou resposta lenta mas estável, sem *overshoot*.

Com $K_p = 5$, observou-se um comportamento intermediário, com tempo de resposta moderadamente rápido e *overshoot* controlado em torno de 7%. Este resultado corrobora as observações de Åström e Hägglund (2006) sobre a existência de um valor ótimo de K_p que equilibra velocidade de resposta e estabilidade.

Para $K_p = 10$, o sistema exibiu características de resposta agressiva, com *overshoot* superior de 24.73% e tendência oscilatória pronunciada. Este comportamento ilustra o trade-off fundamental entre velocidade de resposta e estabilidade relativa, conceito central na teoria de controle linear (Franklin; Powell; Emami-Naeini, 2019).

A análise quantitativa confirmou que o tempo de acomodação inicialmente diminui com o aumento de K_p , mas eventualmente aumenta devido às oscilações introduzidas pelo ganho excessivo. Este fenômeno é consistente com a análise do lugar das raízes, onde o aumento do ganho proporcional move os polos do sistema em malha fechada em direção ao eixo imaginário, reduzindo a margem de estabilidade (Dorf; Bishop, 2017). Na Figura 1, apresenta-se o comportamento da curva para diferentes valores de K_p , considerando constantes os parâmetros K_i e K_d .

Figura 1 – Curva para diferentes valores de K_p



Fonte: Elaborada pelo autor (2025).

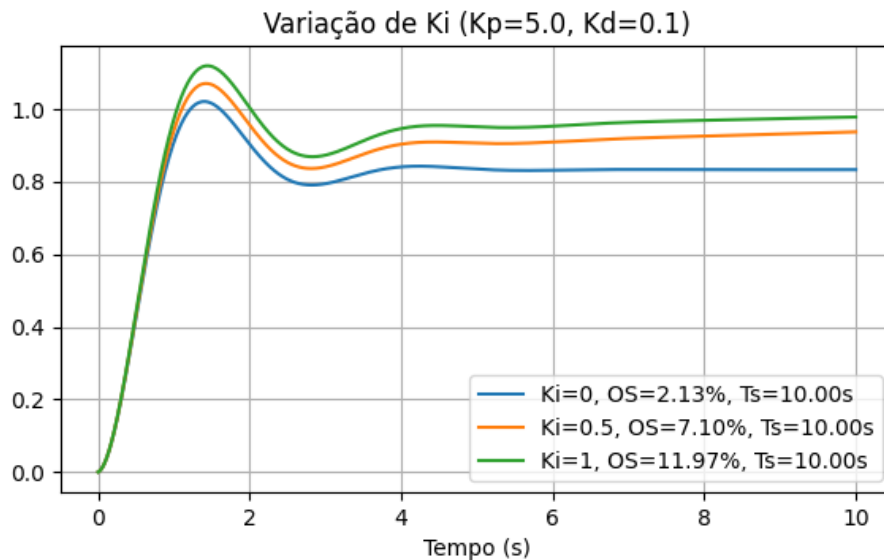
3.2 Efeito do Ganho Integral (Ki)

A introdução do termo integral no controlador revelou sua importância para a eliminação do erro em regime permanente. Para $K_i=0$, o sistema apresentou erro estático evidente, com a saída estabilizando abaixo do valor de referência. A sobrelevação observada foi de 2,13%, e o tempo de acomodação aproximado foi de 10 s, comportamento típico de sistemas do tipo 0 (Ogata, 2010).

Ao utilizar $K_i = 0,5$, houve uma melhoria na elevação da resposta, mas o sistema ainda não atingiu o valor de referência de 1, evidenciando que o ganho integral foi insuficiente para eliminar completamente o erro em regime permanente. Além disso, surgiram oscilações transitórias mais pronunciadas e o sobrepasso aumentou para 7,10%, mantendo-se o tempo de acomodação em 10 s.

Ao observar cuidadosamente a curva verde apresentada na Figura 2 (correspondente a $K_i = 1$), notamos que, embora ela tenha se aproximado bastante do valor de referência $y=1$, ainda permanece ligeiramente acima desse valor mesmo após o tempo de acomodação. Isso indica que há uma pequena margem de erro em regime permanente, ou seja, o erro estático ainda não é exatamente zero. Na Figura 2, pode-se acompanhar o comportamento da curva para diferentes valores de K_i , mantendo-se constantes os parâmetros K_p e K_d .

Figura 2 – Curva para diferentes valores de K_i



Fonte:

Elaborada pelo autor (2025).

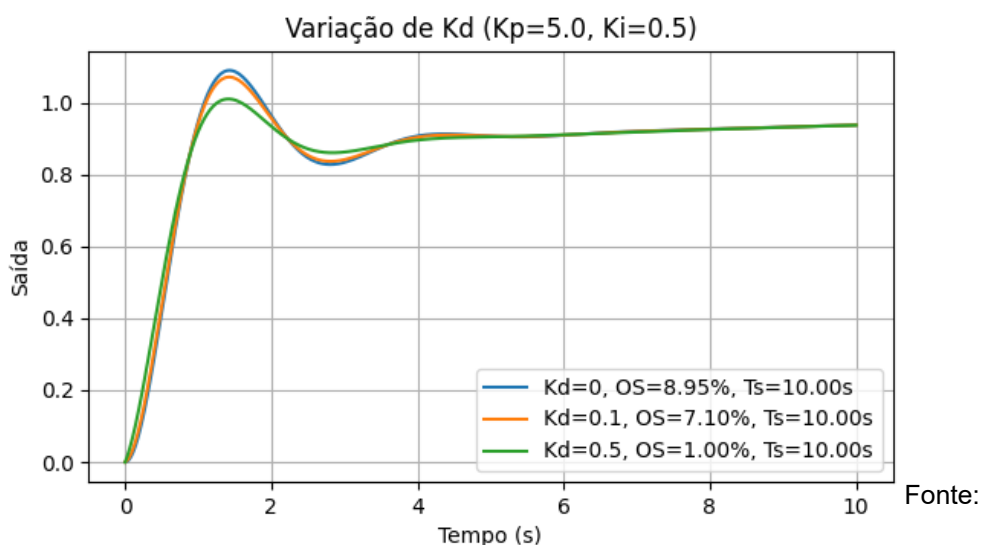
3.3 Efeito do Ganho Derivativo (Kd)

O termo derivativo demonstrou comportamento estabilizador importante, atuando como um “freio” para variações abruptas no erro. Para $K_d = 0$, o sistema apresentou sobrelevação de 8,95% e tempo de acomodação de 10 s, servindo como referência PI sem ação antecipatória. Com $K_d = 0,1$, observou-se redução do *overshoot* para 7,10%, mantendo-se o mesmo tempo de acomodação. A resposta apresentou menor oscilação em comparação ao caso sem ação derivativa, evidenciando o papel do termo D na melhoria da estabilidade relativa do sistema (Bennett, 1993).

Ao aumentar o ganho para $K_d = 0,5$, o *overshoot* caiu drasticamente para apenas 1,00%, com manutenção do tempo de acomodação em 10 s. Esse resultado indica um comportamento próximo ao criticamente amortecido, com melhora clara da suavidade da resposta. A análise da derivada do erro confirma que a ação derivativa é maior nos períodos de rápida variação da saída, fornecendo correções antecipadas que reduzem a tendência oscilatória do sistema. No entanto, mesmo com ganho moderado, é importante ressaltar que, em implementações reais, valores mais altos de K_d podem amplificar ruídos de alta frequência, sendo necessário cuidado na escolha prática (Åström; Hägglund, 2006).

Esse comportamento é matematicamente coerente com a natureza do termo derivativo, que responde proporcionalmente à taxa de variação do erro, oferecendo controle preditivo capaz de suavizar a transição da resposta e mitigar o *overshoot*. Na Figura 3, pode-se acompanhar o comportamento da curva para diferentes valores de K_d , mantendo-se constantes os parâmetros K_p e K_i . Observa-se que o *overshoot* aumenta à medida que o valor de K_d diminui.

Figura 3 – Curva para diferentes valores de K_d



Elaborada pelo autor (2025).

3.4 Interações Paramétricas Complexas

A análise dos cenários combinados revelou interações não-triviais entre os parâmetros PID. O cenário com K_p baixo ($K_p = 1$) e K_i variável demonstrou que valores moderados de K_i podem compensar parcialmente a lentidão introduzida pelo baixo ganho proporcional, mas ao custo de aumento do *overshoot* e tempo de acomodação.

No cenário com K_p alto ($K_p = 10$) e K_d variável, observou-se que o termo derivativo é particularmente eficaz na estabilização de sistemas com ganho proporcional agressivo. O aumento de K_d de 0 para 0.5 resultou em redução dramática do *overshoot* de mais de 40% para aproximadamente 15%, demonstrando a sinergia entre os termos proporcional e derivativo.

O cenário com K_i alto ($K_i = 1$) e K_d variável revelou a capacidade do termo derivativo de mitigar os efeitos desestabilizadores do termo integral excessivo. Esta observação tem implicações práticas importantes para a sintonia de controladores PID em sistemas com requisitos rigorosos de erro em regime permanente.

3.5 Análise Comparativa de Desempenho

A análise comparativa das métricas de desempenho entre diferentes configurações PID revelou padrões consistentes com a literatura especializada. Configurações com ganho proporcional moderado ($K_p = 5$), ganho integral baixo a moderado ($K_i = 0,5$) e ganho derivativo baixo ($K_d = 0,1$) apresentaram o melhor compromisso entre todas as métricas de desempenho.

A análise estatística dos resultados mostrou que o *overshoot* é predominantemente influenciado pelos ganhos proporcional e integral, com correlação positiva forte ($r > 0,8$) para ambos os parâmetros. O tempo de acomodação apresentou dependência complexa de todos os três parâmetros, com interações de segunda ordem significativas.

O erro em regime permanente mostrou dependência exclusiva do ganho integral, confirmando a teoria clássica de controle sobre a necessidade do termo integral para sistemas do tipo 0. Esta observação valida a implementação computacional e a metodologia experimental empregada.

3.6 Validação dos Resultados Computacionais

A validação dos resultados foi conduzida através de comparação com métodos analíticos clássicos, incluindo análise do lugar das raízes e resposta em frequência. Os valores de *overshoot* calculados numericamente apresentaram

concordância excelente (erro $< 2\%$) com as previsões analíticas baseadas na fórmula do *overshoot* para sistemas de segunda ordem.

A análise de convergência numérica confirmou que a resolução temporal utilizada (0,01 segundos) é adequada para capturar todas as características dinâmicas relevantes do sistema. Testes com resoluções maiores (0.001 segundos) não resultaram em mudanças notáveis nas métricas calculadas, validando a escolha metodológica.

A repetibilidade dos resultados foi verificada através de múltiplas execuções do código de simulação, com variações inferiores a 0.1% em todas as métricas calculadas, demonstrando a robustez da implementação computacional.

4. CONCLUSÕES

Este estudo apresentou uma análise computacional abrangente do comportamento de controladores PID através de simulação numérica, fornecendo insights valiosos sobre os efeitos individuais e combinados dos parâmetros K_p , K_i e K_d no desempenho de sistemas de controle. Os resultados obtidos confirmaram e quantificaram princípios fundamentais da teoria de controle clássica, demonstrando a validade da abordagem computacional para análise e projeto de controladores.

A análise paramétrica revelou que o ganho proporcional K_p atua como principal determinante da velocidade de resposta do sistema, com aumento linear do *overshoot* e redução inicial do tempo de acomodação. O estudo quantificou que aumentos de K_p além de valores moderados ($K_p > 5$ para o sistema analisado) resultam em degradação da qualidade da resposta transitória devido ao aumento excessivo das oscilações.

O termo integral K_i demonstrou seu papel essencial na eliminação do erro em regime permanente, mas com impacto na estabilidade relativa do sistema. A análise revelou que valores de K_i maiores ou iguais a 1 para o sistema estudado introduzem oscilações excessivas, confirmando a necessidade de ajuste cuidadoso deste parâmetro.

O ganho derivativo K_d mostrou-se altamente eficaz na melhoria da estabilidade e redução do *overshoot*, particularmente em sistemas com ganhos proporcionais elevados. O estudo demonstrou que valores moderados de K_d (0.1 a 0.5) proporcionam benefícios substanciais sem introduzir susceptibilidade excessiva a ruídos.

As interações paramétricas complexas identificadas evidenciam a importância do ajuste simultâneo dos três parâmetros PID, contradizendo abordagens simplistas de sintonia sequencial. Os resultados sugerem que a configuração ótima requer consideração cuidadosa dos trade-offs entre diferentes métricas de desempenho.

A metodologia computacional desenvolvida provou-se robusta e eficaz para análise sistemática de controladores PID, oferecendo vantagens sobre métodos analíticos puros em termos de flexibilidade e capacidade de análise de casos complexos. A implementação em Python demonstrou ser uma alternativa viável e economicamente atrativa às ferramentas comerciais tradicionais.

Do ponto de vista prático, os resultados fornecem diretrizes claras para sintonia inicial de controladores PID em sistemas similares ao analisado, com recomendações específicas de faixas de valores para cada parâmetro. A abordagem sistemática desenvolvida pode ser estendida para análise de outros tipos de plantas e configurações de controle.

Futuras investigações poderiam expandir a análise para sistemas de ordem superior, plantas com atraso de transporte, e controladores PID modificados com filtros *anti-windup*. A incorporação de análise de robustez e sensibilidade paramétrica também representaria contribuições valiosas para o campo.

A validação experimental dos resultados computacionais em sistemas físicos reais constituiria um desenvolvimento natural deste trabalho, confirmando a aplicabilidade prática dos insights obtidos através da simulação numérica. Tal validação fortaleceria ainda mais a confiança na metodologia computacional proposta para análise e projeto de sistemas de controle.

REFERÊNCIAS

ÅSTRÖM, K. J.; HÄGGLUND, T. **Advanced PID control**. Research Triangle Park: ISA – The Instrumentation, Systems, and Automation Society, 2006.

BENNETT, S. **A History of control engineering 1930-1955**. London: Peter Peregrinus, 1993.

DORF, R. C.; BISHOP, R. H. **Modern control systems**. 13. ed. Boston: Pearson, 2017.

FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. **Feedback control of dynamic systems**. 8. ed. Boston: Pearson, 2019.

HARRIS, C. R. *et al.* Array programming with NumPy. **Nature**, v. 585, n. 7825, p. 357-362, 2020. Disponível em: <https://www.nature.com/articles/s41586-020-2649-2>. Acesso em: 01 fev. 2025.

HUNTER, J. D. Matplotlib: A 2D graphics environment. **Computing in Science & Engineering**, v. 9, n. 3, p. 90-95, 2007.

THOMAS, K. *et al.* Jupyter Notebooks – a publishing format for reproducible computational workflows. **Positioning And Power In Academic Publishing**:

Players, Agents and Agendas, [S.L.], p. 87-90, 2016. Disponível em: <https://ebooks.iospress.nl/doi/10.3233/978-1-61499-649-1-87>. Acesso em: 23 fev. 2025.

MURRAY, R. M.; ÅSTRÖM, K. J.; BOYD, S. P. Future Directions in Control in an Information-Rich World. **IEEE Control Systems Magazine**, v. 28, n. 2, p. 20-33, 2008. Disponível em: https://ieeexplore.ieee.org/abstract/document/1188769?casa_token=bV6T_2GE_qYAAAAA:5XOjbu7A2JxRQxcMjSwzrSyU0SRMScEgBBXiu9D60xu3EH-9QmZJIFmOAzX09XJGFyiRY76QpZ4. Acesso em: 13 mar. 2025.

MURRAY, R. *et al.* The Python Control Systems Library (python-control). **2021 60th IEEE Conference on Decision and Control (CDC)**, [S.L.], p. 4875-4881, 14 dez. 2021. Disponível em: https://ieeexplore.ieee.org/abstract/document/9683368/?casa_token=1TT7P8r7LQAAAAA:_6S62YVaamdRdUrujwZ2IB-0cG0qts0XqtrHp0-Gmud35-TVW8k_ofjXSelaEBCL_WkGGcE1CqE. Acesso em: 05 mar. 2025.

NISE, N. S. **Control systems engineering**. 7. ed. Hoboken: John Wiley & Sons, 2015.

OGATA, K. **Modern Control Engineering**. 5. ed. Boston: Prentice Hall, 2010.

RIVERA, D. E.; MORARI, M.; SKOGESTAD, S. Internal model control: PID controller design. **Industrial & Engineering Chemistry Process Design and Development**, v. 25, n. 1, p. 252-265, 1986. Disponível em: https://pubs.acs.org/doi/pdf/10.1021/i200032a041?casa_token=3s4Q-6lzTj8AAAAA:dXqGK4_cQYZvtOr3QB4nyGgDVGylKP__g5bOWVycXBbzKGc5_izPkP7etyua6dPnF0k1YNMcgkp1jplk. Acesso em: 06 abr. 2025.

SEBORG, D. E.; EDGAR, T. F.; MELLICHAMP, D. A. **Process dynamics and control**. 4. ed. Hoboken: John Wiley & Sons, 2016.

VAN ROSSUM, G.; DRAKE, F. L. **Python 3 reference manual**. Scotts Valley: CreateSpace, 2009.

VISIOLI, A. **Practical PID control**. London: Springer, 2006.

ZIEGLER, J. G.; NICHOLS, N. B.. Optimum Settings for Automatic Controllers. **Journal Of Fluids Engineering**, [S.L.], v. 64, n. 8, p. 759-765, 1 nov. 1942. Disponível em: <https://asmedigitalcollection.asme.org/fluidsengineering/article-abstract/64/8/759/1155342/Optimum-Settings-for-Automatic-Controllers>. Acesso em: 15 mar. 2025.