



Universidade Federal de Ouro Preto
Escola de Minas
CECAU - Colegiado do Curso de
Engenharia de Controle e Automação



Vinicius Augusto Mendes

Ferramenta Interativa Educacional Para Análise de Sistemas de Controle e Projeto de Controladores PID

Ouro Preto, 2025

Vinicius Augusto Mendes

Ferramenta Interativa Educacional Para Análise de Sistemas de Controle e Projeto de Controladores PID

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

Orientador: Prof. João Carlos Vilela de Castro

Ouro Preto

2025



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E
AUTOMACAO



FOLHA DE APROVAÇÃO

Vinicius Augusto Mendes

**Ferramenta Interativa Educacional para Análise de Sistemas de Controle
e Projeto de Controladores PID**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 05 de setembro de 2025

Membros da banca

M.e João Carlos Vilela de Castro - Orientador (Universidade Federal de Ouro Preto)
Dr. Danny Augusto Vieira Tonidandel - Convidado (Universidade Federal de Ouro Preto)
Dr. Paulo Marcos de Barros Monteiro - Convidado (Universidade Federal de Ouro Preto)

João Carlos Vilela de Castro, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 08/09/2025



Documento assinado eletronicamente por **Joao Carlos Vilela de Castro, PROFESSOR DE MAGISTERIO SUPERIOR**, em 08/09/2025, às 14:49, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0974192** e o código CRC **1D7F1747**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.011461/2025-61

SEI nº 0974192

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

Agradecimentos

Agradeço primeiramente a Deus, por ter me concedido fé e resiliência durante toda essa jornada.

Aos meus pais Danuza e Jairo e ao meu irmão Arthur que sempre estiveram ao meu lado me apoiando ao longo de toda a minha trajetória. Sem vocês nada disso seria possível.

Às minhas tias Débora e Paula e ao meu tio Bráulio meu eterno agradecimento por tudo que fizeram e ainda fazem por mim.

Ao meu orientador João Carlos, pelo incentivo e orientação neste trabalho.

Aos professores do curso de Engenharia de Controle e Automação agradeço por todo conhecimento compartilhado durante toda a graduação.

Aos colegas de curso por toda parceria durante todos esses anos.

À Escola de Minas, pela oportunidade e pela honra por fazer parte dessa história.

*A persistência é o caminho do
êxito.*

— Charles Chaplin.

Resumo

O ensino de Teoria de Controle na graduação em engenharia é frequentemente dificultado pela abstração de seus conceitos essenciais. Visando superar essa barreira pedagógica, o presente estudo teve como objetivo desenvolver e validar uma ferramenta computacional didática, interativa e de código aberto para o projeto e a análise de sistemas de controle. A solução foi desenvolvida como uma aplicação *web* em *Python*, utilizando a estrutura *Streamlit* e bibliotecas científicas consolidadas. A plataforma resultante oferece uma interface gráfica para simulação de sistemas via funções de transferência e projeto de controladores Proporcional-Integral-Derivativo (PID) com ajuste interativo. Disponibiliza-se um conjunto de análises visuais — como resposta ao impulso, degrau e Lugar Geométrico das Raízes —, além de permitir a extração de métricas de desempenho e a exportação de relatórios. O *software* desenvolvido reduz a lacuna entre a teoria e a intuição prática. Trata-se de uma contribuição relevante e acessível que fortalece o conjunto de ferramentas educacionais para a Engenharia de Controle e Automação.

Palavras-chaves: Teoria de Controle. Ferramenta Computacional Didática. Controladores PID.

Abstract

Teaching Control Theory in undergraduate engineering programs is often hampered by the abstraction of its essential concepts. Aiming to overcome this pedagogical barrier, the present study aimed to develop and validate an educational, interactive, and open-source computational tool for the design and analysis of control systems. The solution was developed as a web application in Python, using the Streamlit framework and established scientific libraries. The resulting platform offers a graphical interface for system simulation using transfer functions and the design of Proportional-Integral-Derivative (PID) controllers with interactive tuning. A set of visual analyses is provided—such as impulse response, step response, and root locus—in addition to allowing the extraction of performance metrics and the export of reports. The developed software bridges the gap between theory and practical intuition. This is a relevant and accessible contribution that strengthens the set of educational tools for Control and Automation Engineering.

Key-words: Control Theory. Educational Computational Tools. PID Controllers.

Lista de ilustrações

Figura 1 – Sistema de controle	13
Figura 2 – Sistema de controle em malha fechada	14
Figura 3 – Curva exponencial de resposta.	18
Figura 4 – Resposta ao impulso unitário do sistema.	19
Figura 5 – Curva de resposta ao degrau unitário de acordo com ζ	21
Figura 6 – Curva de resposta em degrau unitário que mostra t_d, t_r, t_p, M_p e t_s	22
Figura 7 – Código de módulos e bibliotecas para importação.	33
Figura 8 – Código para criar os controles deslizantes de K_p, K_i, K_d e N	34
Figura 9 – Código para comparação de métricas de desempenho para Sistemas de Segunda Ordem.	36
Figura 10 – Interface inicial da ferramenta.	38
Figura 11 – Interface principal da ferramenta de simulação.	38
Figura 12 – Análise gráfica do domínio do tempo.	39
Figura 13 – Métricas de desempenho e informações do sistema.	39
Figura 14 – Gráfico do Lugar Geométrico das Raízes.	39
Figura 15 – Análise do Lugar das Raízes.	40
Figura 16 – Exportação de resultados.	40
Figura 17 – Respostas ao Degrau e ao Impulso do Sistema 1.	42
Figura 18 – Métricas de desempenho do Sistema 1.	42
Figura 19 – Respostas ao Degrau e ao Impulso do Sistema 1 PID.	42
Figura 20 – Métricas de desempenho do Sistema 1 PID.	43
Figura 21 – Respostas ao Degrau e ao Impulso do Sistema 2.	44
Figura 22 – Métricas de desempenho do Sistema 2.	44
Figura 23 – Respostas ao Degrau e ao Impulso do Sistema 2 PID.	44
Figura 24 – Métricas de desempenho do Sistema 2 PID.	45
Figura 25 – Respostas ao Degrau e ao Impulso do Sistema 3.	46
Figura 26 – Métricas de desempenho do Sistema 3.	46
Figura 27 – Respostas ao Degrau e ao Impulso do Sistema 3 PID.	46
Figura 28 – Métricas de desempenho do Sistema 3 PID.	47
Figura 29 – Respostas ao Degrau e ao Impulso do Sistema 4.	47
Figura 30 – Métricas de desempenho do Sistema 4.	48
Figura 31 – Respostas ao Degrau e ao Impulso do Sistema 4 PID.	48
Figura 32 – Métricas de desempenho do Sistema 4 PID.	49

Sumário

1	INTRODUÇÃO	9
1.1	Justificativas e Relevância	10
1.2	Objetivos	10
1.2.1	Geral	10
1.2.2	Específicos	10
1.3	Organização e estrutura	11
2	REVISÃO DE LITERATURA	12
2.1	Conceitos Fundamentais em Sistemas de Controle	13
2.1.1	Sistemas de Controle em Malha Aberta	13
2.1.2	Sistemas de Controle em Malha Fechada	14
2.2	Modelagem Matemática e Função de Transferência em Sistemas LTI	15
2.2.1	A Função de Transferência	15
2.2.2	Sistemas de Primeira Ordem	17
2.2.2.1	Resposta ao Degrau Unitário	17
2.2.2.2	Resposta ao Impulso Unitário	18
2.2.3	Sistemas de Segunda Ordem	19
2.2.3.1	Ganho em Regime Permanente	20
2.2.3.2	Classificação pelo Fator de Amortecimento e Degrau Unitário	21
2.2.3.3	Parâmetros de Desempenho de um Sistema de Segunda Ordem	21
2.3	Análise de Desempenho e Estabilidade em Malha Fechada	23
2.3.1	Estabilidade	23
2.3.1.1	Definição de Estabilidade BIBO	23
2.3.1.2	Condição Fundamental para a Estabilidade	23
2.3.2	Erro em Regime Estacionário	24
2.4	Projeto de Controladores por Realimentação	25
2.4.1	O Método do Lugar Geométrico das Raízes (LGR)	26
2.4.2	Controladores Proporcional, Integral e Derivativo (PID)	27
2.4.2.1	Controlador Proporcional (P)	27
2.4.2.2	Controlador Proporcional-Integral (PI)	28
2.4.2.3	Controlador Proporcional-Integral-Derivativo (PID)	28
3	DESENVOLVIMENTO	30
3.1	Abordagem de Desenvolvimento	30
3.2	Etapas do Desenvolvimento do Projeto	30
3.2.1	Definição dos Requisitos	31

3.2.2	Escolha das Tecnologias e Ferramentas	31
3.2.2.1	<i>Python</i>	31
3.2.2.2	<i>Streamlit</i>	32
3.2.3	Estrutura da Ferramenta	33
3.2.4	Resultados da Ferramenta Desenvolvida	37
3.2.5	Testes e Validação	40
3.2.6	Documentação e Disponibilização da Ferramenta	40
4	RESULTADOS	41
4.1	Sistema 1 - Desempenho de Resposta Lenta.	41
4.2	Sistema 2 - Comportamento Oscilatório.	43
4.3	Sistema 3 - Erro em Regime Estacionário.	45
4.4	Sistema 4 - Sistema Instável.	47
5	CONSIDERAÇÕES FINAIS	50
5.1	Trabalhos Futuros	50
	Referências	51
	APÊNDICE A – CÓDIGO UTILIZADO	52

1 Introdução

A Engenharia de Controle e Automação se estabelece como uma área estratégica no cenário tecnológico atual, integrando ciência, tecnologia e inovação. Suas aplicações abrangem desde processos industriais complexos e manufatura avançada até áreas emergentes, como robótica colaborativa, veículos autônomos, sistemas embarcados e redes inteligentes de energia. O foco dessa área está na capacidade de modelar, analisar e intervir no comportamento de sistemas dinâmicos, assegurando operação conforme critérios de estabilidade, precisão, robustez e eficiência (OGATA, 2010; ÅSTRÖM; MURRAY, 2017; NISE, 2023). Essa abordagem é fundamentada em uma base matemática e computacional sólida, permitindo a abstração de fenômenos físicos em modelos representativos, que servem como base para o desenvolvimento de algoritmos de controle sofisticados aplicáveis em diversos contextos.

Apesar de sua relevância prática, a transposição dos conceitos teóricos da Teoria de Controle para o ambiente de aprendizagem enfrenta desafios pedagógicos, especialmente no ensino de graduação. Tópicos centrais, como a análise de estabilidade no plano complexo e o Lugar Geométrico das Raízes (LGR), são altamente abstratos, dificultando a compreensão dos estudantes (DORF; BISHOP, 2018).

Tradicionalmente, o ensino de conteúdos de controle tem sido suportado por plataformas computacionais proprietárias, como *MATLAB* e sua ferramenta de programação em blocos *Simulink*, que são referências na indústria e na academia. Apesar de suas funcionalidades robustas, essas ferramentas têm como limitação o alto custo de licenciamento, o que dificulta o acesso em instituições com restrições orçamentárias (MATHWORKS INC., 2020). Além do impacto econômico, a dependência de tecnologias proprietárias restringe a democratização do conhecimento e limita a adoção de metodologias pedagógicas mais acessíveis e colaborativas. Diante desse cenário, a pesquisa busca responder à questão: **Como ferramentas computacionais de acesso livre podem contribuir para o aprimoramento do ensino-aprendizagem da teoria de controle, na forma de ferramentas interativas e de uso fácil, em instituições com recursos financeiros limitados?**

O presente trabalho visa desenvolver uma ferramenta alternativa, baseada em código aberto e voltada para fins educacionais, com o objetivo de democratizar o acesso a recursos de simulação e análise de sistemas dinâmicos. A proposta incorpora abordagens pedagógicas contemporâneas, como interatividade, experimentação virtual e aprendizado ativo. Assim, espera-se contribuir para uma compreensão mais profunda dos conteúdos e promover uma formação de engenheiros mais preparados para os desafios tecnológicos e

sociais atuais.

1.1 Justificativas e Relevância

A justificativa para o desenvolvimento deste trabalho está na combinação da democratização do acesso a ferramentas educacionais, a modernização das práticas pedagógicas em engenharia e a necessidade de alinhar a formação acadêmica às demandas profissionais.

O pilar da acessibilidade é garantido pela adoção de um ecossistema tecnológico de código aberto, desenvolvido integralmente em *Python* e suportado por bibliotecas como *python-control*, *NumPy* e *Matplotlib*, eliminando os custos de licenciamento, que são um obstáculo em muitas instituições. Essa abordagem está alinhada com uma tendência global na educação em engenharia que valoriza o uso de tecnologias livres para promover inclusão e autonomia no ambiente educacional. Assim, a solução se torna uma alternativa viável e sustentável, podendo ser livremente distribuída, utilizada e modificada pela comunidade acadêmica.

A interface interativa permite que os alunos manipulem variáveis de controle e observem, em tempo real, as respostas dinâmicas do sistema, transformando conceitos matemáticos abstratos em fenômenos observáveis. Isso facilita uma compreensão intuitiva dos princípios de controle, funcionando como uma ponte eficaz entre teoria e prática, simulando a experiência laboratorial sem os custos e riscos de um ambiente físico.

Por fim, a ferramenta indo além do papel didático ao incluir funcionalidades de documentação e comunicação técnica, como exportação de dados e geração de relatórios. Essa solução facilita a integração de simulações em projetos acadêmicos e contribui para o desenvolvimento de competências valorizadas no mercado de trabalho, como análise de dados e apresentação clara de resultados, essenciais para a formação do engenheiro moderno.

1.2 Objetivos

1.2.1 Geral

O objetivo geral deste trabalho é projetar, desenvolver e validar uma ferramenta de software interativa e educacional para a análise de sistemas de controle dinâmico para projeto de controladores PID, que permita a exploração intuitiva de conceitos teóricos e a exportação de resultados em formatos profissionais.

1.2.2 Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Realizar uma revisão bibliográfica sobre os desafios no ensino de Teoria de Controle e o uso de ferramentas computacionais de código aberto como suporte pedagógico;
- Especificar os requisitos funcionais e não funcionais da ferramenta, definindo as análises a serem implementadas (resposta ao degrau, impulso e Lugar Geométrico das Raízes) e as funcionalidades de interação;
- Implementar a aplicação *web* utilizando a linguagem *Python* e a estrutura *Streamlit*, integrando bibliotecas científicas para os cálculos de controle e a geração de visualizações gráficas;
- Analisar os resultados da validação e discutir o potencial da ferramenta desenvolvida como um recurso didático eficaz para o ensino e aprendizado de sistemas de controle.

1.3 Organização e estrutura

Este trabalho está estruturado em 5 capítulos. O presente Capítulo 1 introduziu o tema, contextualizando o problema de pesquisa, a justificativa, relevância e os objetivos do estudo. O Capítulo 2 apresenta a fundamentação teórica, abordando os conceitos essenciais de sistemas de controle lineares, controladores PID e as ferramentas computacionais utilizadas. O Capítulo 3 descreve a metodologia empregada no desenvolvimento da solução. O Capítulo 4 apresenta a ferramenta desenvolvida, seus resultados e a validação. Finalmente, o Capítulo 5 traz as conclusões, as contribuições do trabalho e sugestões para trabalhos futuros.

2 Revisão de literatura

As primeiras noções de controle automático surgiram no século XVIII com o regulador centrífugo de James Watt para máquinas a vapor, que é considerado o marco inicial dos sistemas de controle por realimentação, pois ajustava automaticamente a velocidade da máquina conforme as variações de carga, estabelecendo o princípio da realimentação negativa (OGATA, 2010; NISE, 2023).

Durante o século XX, a teoria de controle foi consolidada por meio de avanços matemáticos, como a Transformada de Laplace e representações em espaço de estados. Autores como Dorf e Bishop (2018) e Åström e Murray (2017) ressaltam que a Segunda Guerra Mundial impulsionou a pesquisa em sistemas de controle, especialmente em aplicações de radares, servomecanismos e controle de aeronaves.

Na década de 1960, a formulação moderna da teoria de controle, fundamentada em espaço de estados, tornou-se relevante, possibilitando a análise e o projeto de sistemas multivariáveis, não lineares e digitais.

Com o avanço computacional desde a década de 1980, o desenvolvimento de *softwares* como *MATLAB/Simulink*, *Octave* e bibliotecas em *Python* (*Python-Control*, *Simple-PID*, *SciPy*) democratizou o acesso à simulação e ao projeto de controladores, tornando essas ferramentas essenciais no ensino e na prática profissional.

A Teoria de Controle evoluiu de soluções mecânicas simples para métodos matemáticos e computacionais avançados, sendo atualmente aplicada em setores estratégicos como automação industrial, aeronáutica, robótica, energia, sistemas biomédicos e veículos autônomos (DORF; BISHOP, 2018; ÅSTRÖM; MURRAY, 2017).

A evolução histórica da Engenharia de Controle, conforme apresentado, evidencia uma contínua transição de uma abordagem prática e mecânica para a abstração matemática e, por fim, para a simulação computacional. Essa jornada, iniciada no século XVIII e acelerada ao longo dos séculos XX e XXI, transformou o que era uma solução para um problema específico em uma área do conhecimento que serve de base para muitos outros campos da tecnologia. Atualmente, a capacidade de modelar, analisar e controlar sistemas complexos é um requisito essencial para a inovação em setores que vão da automação industrial à robótica autônoma, demonstrando a importância central e a relevância contínua do tema.

2.1 Conceitos Fundamentais em Sistemas de Controle

A teoria de controle é fundamental na Engenharia de Controle e Automação, oferecendo métodos para modelar, analisar e projetar sistemas dinâmicos que atendem a especificações de desempenho e estabilidade.

Um sistema de controle é definido como um conjunto de dispositivos que gera comandos para manter ou ajustar uma grandeza, envolvendo a variável controlada (saída do sistema) e a variável manipulada (entrada ajustada pelo controlador), (ver figura 1) (OGATA, 2010). Os conceitos de estabilidade, tempo de resposta e erro em regime permanente são fundamentais na análise de desempenho desses sistemas, sendo essenciais para avaliar se atendem às especificações em regime transitório e estacionário (NISE, 2023). Os sistemas de controle podem ser classificados em malha aberta e malha fechada, com a realimentação (*feedback*) desempenhando um papel crucial ao proporcionar robustez a incertezas e rejeição de perturbações (KUO; GOLNARAGHI, 2014; ÅSTRÖM; MURRAY, 2017).

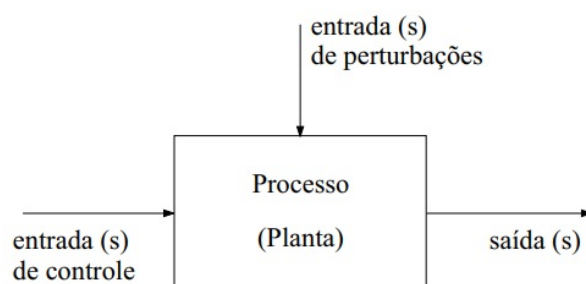


Figura 1 – Sistema de controle. Fonte: (BAZANELLA; SILVA JR., 2005).

2.1.1 Sistemas de Controle em Malha Aberta

Os sistemas em malha aberta se caracterizam por sua simplicidade estrutural, pois não necessitam de sensores de realimentação ou comparação entre saída e referência, tornando-se mais baratos e fáceis de projetar, sendo adequados para processos onde a precisão extrema não é crítica (OGATA, 2010). Eles são apropriados quando a relação entre entrada e saída é bem conhecida e constante ao longo do tempo (NISE, 2023). No entanto, essa simplicidade apresenta limitações, como a incapacidade de lidar com incertezas e perturbações, o que os torna vulneráveis a desvios de desempenho e restringe sua aplicação em sistemas complexos ou de alta criticidade (KUO; GOLNARAGHI, 2014).

Exemplos práticos de sistemas em malha aberta incluem uma máquina de lavar roupas, que segue etapas pré-programadas por tempos fixos sem avaliar o nível de limpeza, e um forno de micro-ondas, que opera com uma potência definida por um tempo determinado pelo usuário, sem monitorar a temperatura real do alimento.

Portanto, embora os sistemas em malha aberta ofereçam simplicidade e facilidade de implementação, sua eficácia depende fortemente de uma calibração precisa e de um ambiente controlado.

2.1.2 Sistemas de Controle em Malha Fechada

Um sistema de controle em malha fechada é caracterizado pela realimentação, onde a saída do processo, $y(t)$, é medida e comparada a um valor de referência, $r(t)$. A diferença entre esses sinais, chamada de erro, $e(t)$, é utilizada pelo controlador para ajustar a ação de controle, aproximando o sistema do comportamento desejado. Tal relação é escrita como:

$$e(t) = r(t) - y(t) \quad (2.1)$$

Essa estrutura proporciona ao sistema autorregulação, robustez contra perturbações externas e maior precisão no rastreamento de trajetórias.

Pela figura 2, a principal vantagem dos sistemas em malha fechada é sua capacidade de reduzir incertezas paramétricas e distúrbios externos, aumentando a confiabilidade do sistema em processos que exigem desempenho rigoroso (OGATA, 2010). A realimentação torna os sistemas menos sensíveis a variações não modeladas e ao ambiente externo, facilitando o cumprimento de critérios como tempo de resposta, estabilidade e erro em regime permanente, o que os torna mais adequados para processos industriais modernos (NISE, 2023). Além disso, a realimentação é um paradigma central na engenharia de controle, não apenas corrigindo erros, mas também introduzindo propriedades emergentes como adaptabilidade e robustez, ampliando sua aplicação em diversos sistemas (ÅSTRÖM; MURRAY, 2017).

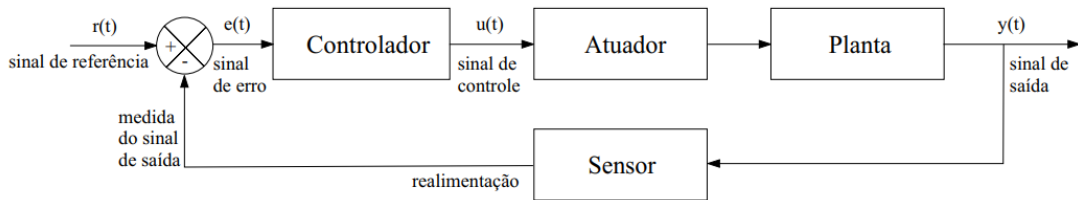


Figura 2 – Sistema de controle em malha fechada. Fonte: (BAZANELLA; SILVA JR., 2005).

Os principais exemplos de sistemas de controle em malha fechada incluem o termostato de um ar-condicionado e o piloto automático de um veículo. O termostato mede constantemente a temperatura do ambiente e, se houver desvio do valor desejado (*setpoint*), ajusta o compressor para corrigir o erro e manter a temperatura estável. Da mesma forma, o piloto automático compara a velocidade real do carro, medida por sensores, com a velocidade programada pelo motorista, ajustando a aceleração do motor para compensar variações, como subidas e descidas, garantindo uma velocidade constante.

Portanto, de maneira geral, pode-se dizer que os sistemas em malha aberta são simples e de baixo custo, enquanto os sistemas em malha fechada oferecem maior precisão e robustez, sendo essenciais em aplicações onde a estabilidade e a resposta a perturbações são críticas.

2.2 Modelagem Matemática e Função de Transferência em Sistemas LTI

Os sistemas dinâmicos são sistemas que evoluem ao longo do tempo, governados por variáveis internas, conhecidas como estados, que interagem com variáveis externas, chamadas entradas. Diferentemente dos sistemas estáticos, cuja saída depende apenas do valor instantâneo da entrada, os sistemas dinâmicos possuem memória, ou seja, seu comportamento atual é influenciado não só pelas entradas presentes, mas também pelo histórico de estímulos anteriores e pelas condições iniciais do sistema (OGATA, 2010; NISE, 2023). A descrição matemática desses sistemas geralmente utiliza equações diferenciais, sendo que sistemas de Engenharia de Controle e Automação são frequentemente representados por equações diferenciais ordinárias (EDOs).

No campo do projeto e análise de controladores, os modelos lineares invariantes no tempo (LTI) desempenham um papel central. Embora muitos sistemas físicos apresentem comportamentos não lineares, sua dinâmica pode frequentemente ser aproximada com grande precisão por um modelo LTI em torno de um ponto de operação específico. Essa abordagem é amplamente utilizada porque a estrutura matemática dos sistemas LTI, que obedece ao princípio da superposição e possui características constantes no tempo, facilita enormemente a aplicação de ferramentas de análise e projeto. Assim, a modelagem LTI representa uma ferramenta poderosa e eficaz para uma vasta gama de aplicações em controle automático (DORF; BISHOP, 2018).

2.2.1 A Função de Transferência

A análise de sistemas lineares invariantes no tempo (LTI), frequentemente descritos por equações diferenciais, pode se tornar complexa. Para simplificar este processo, a Transformada de Laplace é uma técnica amplamente empregada. Ela permite que as equações diferenciais, um problema de cálculo, sejam convertidas em equações algébricas, que são consideravelmente mais simples de manipular. Essa conversão facilita a resolução de problemas ligados ao comportamento dinâmico e possibilita uma análise mais clara e sistemática do sistema.

A Transformada de Laplace é uma ferramenta matemática, classificada como uma transformada integral, que converte uma função do domínio do tempo, denotada por $f(t)$,

para uma função no domínio da frequência complexa, denotada por $F(s)$. A sua definição formal é dada pela seguinte integral:

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt \quad (2.2)$$

Na engenharia de controle, a principal aplicação desta transformada reside em sua notável capacidade de converter equações diferenciais ordinárias lineares em equações algébricas. Essa transformação é extremamente vantajosa, pois problemas que exigiriam métodos de cálculo para serem resolvidos no domínio do tempo tornam-se problemas algébricos, cuja manipulação e solução são significativamente mais simples e sistemáticas.

A Transformada de Laplace possui um conjunto extenso de propriedades que facilitam a análise de sinais e sistemas. Contudo, para os objetivos deste trabalho, a discussão será focada apenas nas propriedades estritamente necessárias para a obtenção da função de transferência de um sistema a partir de sua equação diferencial modelada. Dentre estas, a propriedade da transformada para derivadas é de fundamental importância, pois ela permite converter os termos diferenciais que caracterizam a dinâmica do sistema em termos algébricos simples no domínio da frequência.

Assumindo condições iniciais nulas, a propriedade da transformada para derivadas é:

$$\mathcal{L}\left\{\frac{d^k x(t)}{dt^k}\right\} = s^k X(s). \quad (2.3)$$

Aplicando a Transformada de Laplace em ambos os lados da equação (2.3), obtemos:

$$(a_n s^n + \dots + a_0)Y(s) = (b_m s^m + \dots + b_0)U(s) \quad (2.4)$$

Agora, podemos isolar a relação entre a saída $Y(s)$ e a entrada $U(s)$, que é a Função de Transferência, $G(s)$:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (2.5)$$

Essa representação é de grande relevância pois permite analisar estabilidade, resposta transitória e comportamento em regime permanente (NISE, 2023).

Os conceitos de polos e zeros são fundamentais na análise de sistemas dinâmicos e decorrem diretamente da função de transferência (2.5) (OGATA, 2010; FRANKLIN; POWELL; EMAMI-NAEINI, 2019):

- Zeros são as raízes do numerador da função de transferência, isto é, os valores de s para os quais $Y(s) = 0$. Estão associados a características específicas da resposta, podendo introduzir atrasos de fase e alterar a velocidade do sistema;

- Polos são as raízes do denominador da função de transferência, correspondendo aos valores de s que anulam $U(s)$. Os polos determinam a dinâmica natural do sistema, incluindo estabilidade, frequência natural e taxa de amortecimento.

2.2.2 Sistemas de Primeira Ordem

Um sistema linear e invariável no tempo (LTI) de primeira ordem é aquele cuja dinâmica pode ser descrita por uma equação diferencial linear de primeira ordem:

$$\tau \frac{dy(t)}{dt} + y(t) = Ku(t) \quad (2.6)$$

Aplicando a transformada de Laplace na equação (2.6), considerando condições iniciais nulas ($y(0) = 0$), temos que a função de transferência do sistema de primeira ordem é:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} \quad (2.7)$$

Onde:

- $Y(s)$ é a transformada de Laplace do sinal de saída do sistema;
- $U(s)$ é a transformada de Laplace do sinal de entrada do sistema;
- K é o ganho estático;
- τ é a constante de tempo, que indica a rapidez da resposta do sistema.

2.2.2.1 Resposta ao Degrau Unitário

A Resposta ao Degrau Unitário é o comportamento da saída de um sistema, denotada como $y(t)$, quando sua entrada é submetida a um sinal de degrau unitário. Em outras palavras, ela descreve como o sistema reage a uma mudança súbita e constante em seu comando ou referência, de forma análoga a ligar um interruptor ou ajustar um termostato para uma nova temperatura. Trata-se do teste padrão mais fundamental na engenharia de controle, pois revela as características dinâmicas essenciais de um sistema.

A análise desta resposta é tipicamente dividida em duas fases principais:

- Regime Transitório: Corresponde ao comportamento inicial do sistema, imediatamente após a aplicação do degrau. É a fase de adaptação, onde podem ocorrer oscilações e a saída está em processo de mudança para atingir seu novo estado. Esta fase descreve a dinâmica do sistema.
- Regime Estacionário: Refere-se ao comportamento da saída após um longo período de tempo, quando o regime transitório já cessou e o sistema se estabilizou. Esta fase descreve o comportamento final e a precisão do sistema.

Se aplicarmos um degrau como sinal de entrada, cuja transformada de Laplace é $U(s) = 1/s$, a saída no domínio de Laplace será:

$$Y(s) = \frac{K}{\tau s + 1} \cdot \frac{1}{s} = \frac{K}{s(\tau s + 1)} \quad (2.8)$$

Fazendo a transformada inversa de Laplace, obtemos a resposta no tempo:

$$y(t) = K(1 - e^{-t/\tau}), \quad t \geq 0 \quad (2.9)$$

Em regime permanente, a saída do sistema tende a se estabilizar em torno de K . Ao substituir t por τ na equação (2.9), observa-se que o valor da saída é 0,632, indicando que nesse instante, ou seja, em uma constante de tempo, a resposta do sistema atingiu 63,2% de sua variação total. Logo, quanto menor a constante de tempo τ , mais rápido o sistema irá responder. Com isso, de acordo com a figura 3, em que a constante de tempo é representada por T , nota-se que, para o sistema atingir e permanecer a 2% de seu valor final, são necessárias quatro constantes de tempo. Assim, para $t \geq 4T$, a resposta se mantém a 2% da linha do valor final.

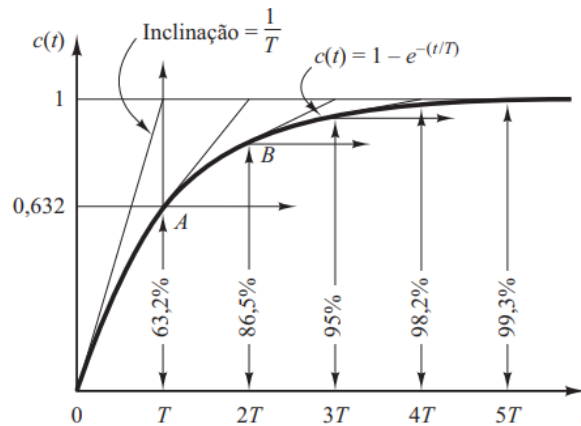


Figura 3 – Curva exponencial de resposta. Fonte: (OGATA, 2010).

2.2.2.2 Resposta ao Impulso Unitário

A Resposta ao Impulso Unitário descreve o comportamento da saída de um sistema quando sua entrada é submetida a um sinal de impulso unitário ($u(t)=\delta(t)$). Para fins de compreensão, o impulso pode ser visualizado como um sinal teórico de curtíssima duração e altíssima amplitude, cuja área total é igual a 1. Fisicamente, ele representa uma perturbação instantânea, como uma martelada, um choque súbito ou uma injeção de energia em um instante de tempo muito breve.

Assim, a resposta ao impulso revela a dinâmica natural de um sistema. Ela demonstra como o sistema reage e dissipa a energia introduzida por essa perturbação instantânea,

quando deixado a evoluir livremente. Observar se a resposta decai a zero, oscila ou cresce indefinidamente fornece uma indicação visual e imediata sobre a estabilidade do sistema.

Se aplicarmos um impulso unitário ($u(t)=\delta(t)$), cuja transformada de Laplace é $U(s)=1$, a saída será:

$$Y(s) = \frac{K}{\tau s + 1} \quad (2.10)$$

Aplicando a transformada inversa de Laplace, temos:

$$y(t) = \frac{K}{\tau} e^{-t/\tau}, \quad t \geq 0 \quad (2.11)$$

Observando a figura 4, temos a curva de resposta dada pela equação 2.11

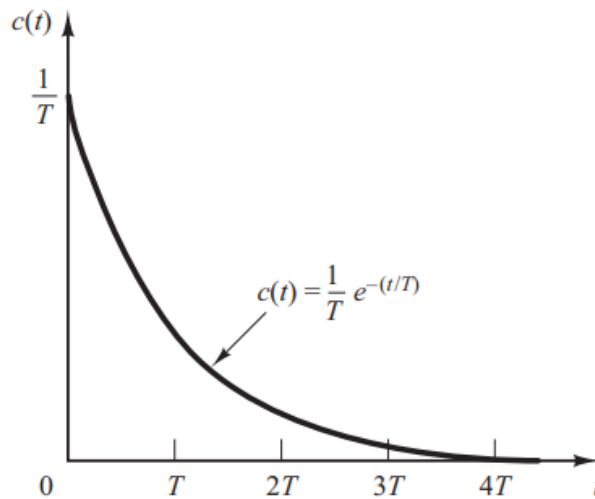


Figura 4 – Resposta ao impulso unitário do sistema. Fonte: (OGATA, 2010).

Em resumo, enquanto a resposta ao degrau avalia como um sistema reage a uma mudança de referência sustentada, a resposta ao impulso caracteriza o comportamento fundamental do sistema frente a uma perturbação instantânea, sendo uma ferramenta teórica poderosa para a compreensão de sua estabilidade e dinâmica natural.

2.2.3 Sistemas de Segunda Ordem

Um sistema dinâmico linear de segunda ordem, com parâmetros concentrados, pode ser descrito pela equação diferencial:

$$\frac{d^2 y(t)}{dt^2} + 2\zeta\omega_n \frac{dy(t)}{dt} + \omega_n^2 y(t) = \omega_n^2 u(t) \quad (2.12)$$

Onde:

- ω_n é frequência natural não amortecida (rad/s);
- ζ é o fator de amortecimento;

- $u(t)$ é a entrada;
- $y(t)$ é a saída.

Aplicando a transformada de Laplace na equação 2.12, a função de transferência de um sistema de 2ª ordem é:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.13)$$

2.2.3.1 Ganho em Regime Permanente

A função de transferência apresentada na equação 2.13 é a forma padrão normalizada. Uma característica importante a ser observada é o seu ganho em regime permanente, também conhecido como ganho estático. Este ganho representa a razão entre a saída e a entrada do sistema após um longo período de tempo, quando as transitórias já se extinguíram. Ele pode ser calculado fazendo $s \rightarrow 0$ na função de transferência:

$$G(0) = \frac{\omega_n^2}{0^2 + 2\zeta\omega_n(0) + \omega_n^2} = \frac{\omega_n^2}{\omega_n^2} = 1 \quad (2.14)$$

Isso demonstra que esta forma normalizada possui um ganho estático unitário. No entanto, em sistemas físicos reais, é comum que este ganho seja um valor diferente de 1. A forma geral da função de transferência de um sistema de segunda ordem, que inclui o ganho estático K , é dada por:

$$G(s) = \frac{K \cdot \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.15)$$

Onde K é o ganho em regime permanente do sistema. É crucial notar que o ganho K impacta o comportamento do sistema da seguinte maneira:

- Não altera a dinâmica transitória: O comportamento da resposta, o sobressinal percentual e o tempo de acomodação continuam dependendo exclusivamente de ζ e ω_n .
- Escala a amplitude da saída: O ganho K atua como um multiplicador direto no valor final da resposta. Para uma entrada degrau unitário, a saída do sistema se estabilizará no valor K .
- Introduz erro em regime estacionário: Se o objetivo é que a saída siga uma referência de valor 1, um ganho $K \neq 1$ resultará em um erro em regime estacionário (e_{ss}) igual a $1-K$.

2.2.3.2 Classificação pelo Fator de Amortecimento e Degrau Unitário

Ogata (2010) explica que o comportamento dinâmico do sistema de segunda ordem depende do fator de amortecimento (ζ), (ver figura 5). Nesse sentido pode-se classificar o sistema como:

1. Subamortecido ($0 < \zeta < 1$):

- Resposta oscilatória decrescente.
- Pólos complexos conjugados em:

$$s = -\zeta\omega_n \pm j\omega_d \quad (2.16)$$

onde: $\omega_d = \omega_n \sqrt{1 - \zeta^2}$

2. Amortecimento crítico ($\zeta=1$):

- Resposta mais rápida sem oscilações.
- Pólos reais e iguais.

3. Superamortecido ($\zeta > 1$):

- Resposta lenta, sem oscilações.
- Pólos reais e distintos.

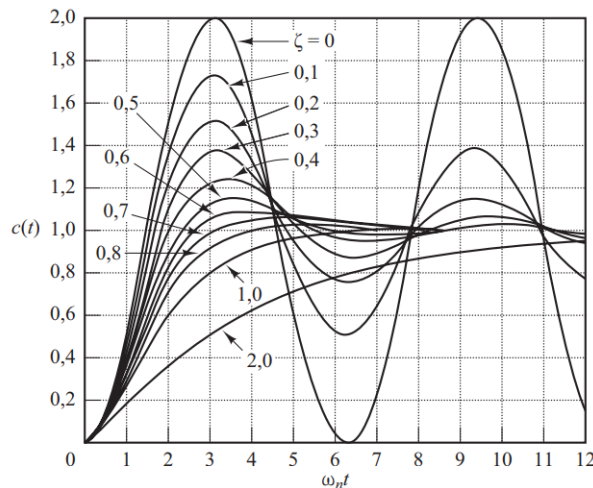


Figura 5 – Curva de resposta ao degrau unitário de acordo com ζ . Fonte: (OGATA, 2010).

2.2.3.3 Parâmetros de Desempenho de um Sistema de Segunda Ordem

Segundo Dorf e Bishop (2018) e Ogata (2010), a resposta transitória refere-se ao comportamento do sistema desde a aplicação de um sinal até sua estabilização. Essa resposta transitória pode ser caracterizada por parâmetros, sendo os principais ilustrados na figura 6:

- Tempo de atraso, t_d : É o tempo requerido para que a resposta alcance metade de seu valor final pela primeira vez.
- Tempo de subida, t_r : É o tempo que a resposta leva para ir de 10% a 90% do seu valor final em sistemas superamortecidos. Para sistemas subamortecidos, às vezes, é definido como o tempo para ir de 0% a 100%.
- Tempo de acomodação, t_s : É o tempo que a resposta leva para entrar e permanecer dentro de uma faixa de tolerância em torno do valor final. Caso seja utilizado o critério de 2%, o t_s será de $t_s \approx 4\tau = \frac{4}{\zeta\omega_n}$. Caso seja utilizado o critério de 5%, o t_s será de $t_s \approx 3\tau = \frac{3}{\zeta\omega_n}$.
- Tempo de pico, t_p : É o tempo para que a resposta atinja o máximo sobre-sinal, podendo ser calculado da seguinte forma:

$$t_p = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (2.17)$$

- Máximo sobre-sinal - 'overshoot' (em porcentagem), M_p : É a diferença percentual entre o pico máximo da resposta e o valor de regime permanente. Pode ser calculado por:

$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% \quad (2.18)$$

$$M_p = 100e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (2.19)$$

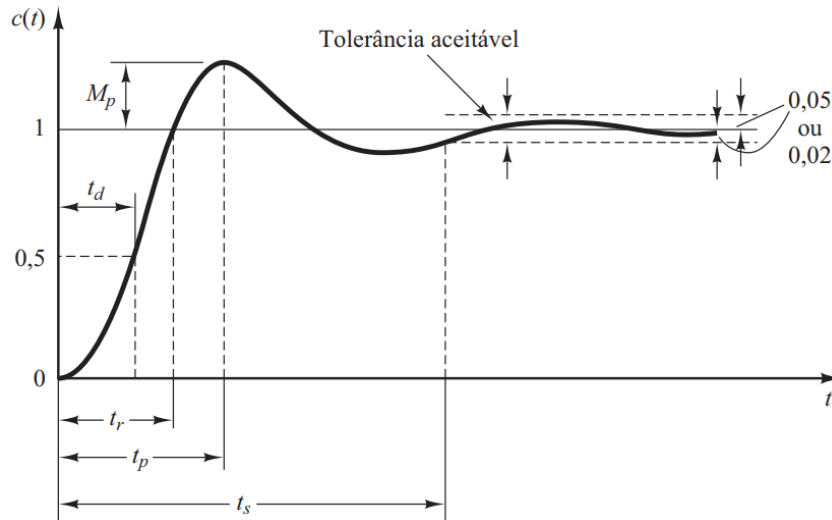


Figura 6 – Curva de resposta em degrau unitário que mostra t_d , t_r , t_p , M_p e t_s . Fonte: (OGATA, 2010).

2.3 Análise de Desempenho e Estabilidade em Malha Fechada

2.3.1 Estabilidade

A estabilidade é um conceito fundamental na análise e no projeto de sistemas de controle, garantindo que o sistema se comporte de maneira previsível e segura. Uma das definições mais importantes de estabilidade é a estabilidade BIBO — '*Bounded-Input, Bounded-Output*' (Entrada Limitada, Saída Limitada).

2.3.1.1 Definição de Estabilidade BIBO

Um sistema é considerado estável no sentido BIBO se, para qualquer sinal de entrada que seja limitado em amplitude, o sinal de saída correspondente também for limitado em amplitude. Em outras palavras, uma entrada que não cresce indefinidamente não resultará em uma saída que cresça indefinidamente (NISE, 2023).

Matematicamente, se a entrada do sistema, $u(t)$, satisfaz a condição:

$$|u(t)| \leq M < \infty \quad \forall t \geq 0 \quad (2.20)$$

onde M é uma constante finita, então a saída do sistema, $y(t)$, também deve satisfazer a condição:

$$|y(t)| \leq N < \infty \quad \forall t \geq 0 \quad (2.21)$$

onde N é outra constante finita. Se essa propriedade for válida para todas as entradas limitadas possíveis, o sistema é classificado como BIBO estável (OGATA, 2010).

2.3.1.2 Condição Fundamental para a Estabilidade

Para um sistema linear e invariante no tempo (LTI) contínuo, representado por sua função de transferência:

$$G(s) = \frac{N(s)}{D(s)}, \quad (2.22)$$

onde $D(s)$ é o polinômio característico associado, a condição de estabilidade é (OGATA, 2010; NISE, 2023):

O sistema é BIBO estável se, e somente se, todos os polos de $G(s)$ estiverem no semi-plano esquerdo do plano- s .

Ou seja, todos os polos devem ter parte real negativa:

$$\Re(s_i) < 0, \quad \forall s_i \in \text{Polos de } G(s). \quad (2.23)$$

O plano- s é uma representação complexa onde o eixo horizontal representa a parte real (σ) e o eixo vertical representa a parte imaginária ($j\omega$) dos polos do sistema. Os polos são as raízes do denominador da função de transferência de malha fechada do sistema.

A localização desses polos determina a natureza da resposta temporal do sistema. Polos no semi-plano do plano-s garantem uma resposta transitória que decai exponencialmente. Polos no eixo imaginário ($\Re(s)=0$) indicam estabilidade marginal, enquanto polos no semi-plano direito ($\Re(s)>0$) resultam em instabilidade (DORF; BISHOP, 2018).

2.3.2 Erro em Regime Estacionário

O erro em regime estacionário (e_{ss}) é definido como a diferença entre o valor desejado da saída de um sistema e o valor efetivamente atingido pela resposta quando $t \rightarrow \infty$. Em termos matemáticos, pode ser expresso como:

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{t \rightarrow \infty} [r(t) - y(t)], \quad (2.24)$$

onde $r(t)$ é a entrada de referência e $y(t)$ a saída do sistema (OGATA, 2010; NISE, 2023).

Nos sistemas lineares invariantes no tempo (LTI), a determinação do erro em regime estacionário pode ser realizada com base no teorema do valor final, aplicado à função erro:

$$e_{ss} = \lim_{s \rightarrow 0} sE(s), \quad (2.25)$$

com $E(s)=R(s)-C(s)$, sendo $R(s)$ a transformada de Laplace da entrada e $C(s)$ a transformada da saída (DORF; BISHOP, 2018).

A análise do erro em regime estacionário é particularmente importante para avaliar o desempenho de sistemas de controle diante de entradas típicas, como:

- Entrada degrau unitário: mede a precisão para rastreamento de referências constantes;
- Entrada rampa unitária: avalia a capacidade de seguimento de sinais lineares crescentes;
- Entrada parábola: verifica a robustez frente a variações aceleradas (OGATA, 2010).

De acordo com a teoria clássica, o valor de e_{ss} depende diretamente do tipo de sistema. Essa classificação é determinada pelo número de polos na origem ($s=0$), também conhecidos como polos integradores, presentes na função de transferência de malha aberta do sistema, que em um sistema controlado corresponde ao produto $G(s)C(s)$. Sistemas do tipo 0, 1 ou 2 (degrau, rampa ou parábola) apresentam diferentes capacidades de rejeição ao erro frente a entradas polinomiais (NISE, 2023).

Além disso, pesquisas recentes reforçam a importância do erro em regime estacionário em aplicações modernas, como controle adaptativo, sistemas robóticos e processos industriais. Trabalhos como o de Silva, Costa e Lima (2019) analisam o impacto do erro em regime estacionário em sistemas discretos, enquanto García, Ponsart e Theilliol (2021)

destacam sua relevância em estratégias de controle preditivo. Estudos mais recentes, como o de [Zhou e Wang \(2022\)](#), exploram técnicas avançadas de compensação do erro estacionário em sistemas não lineares.

2.4 Projeto de Controladores por Realimentação

O comportamento natural de um sistema físico, tecnicamente chamado de planta, muitas vezes não é adequado para a sua aplicação final, podendo ser lento, impreciso ou instável. Para corrigir essas deficiências, utiliza-se a estratégia de controle em malha fechada, na qual a saída do sistema é continuamente medida e comparada com o valor desejado, referência. A diferença entre eles, o erro, é usada por um controlador para ajustar a entrada da planta, forçando-a a operar conforme as especificações de desempenho predeterminadas ([NISE, 2023](#)).

A chave para alterar o comportamento de um sistema reside na manipulação de suas características dinâmicas fundamentais. Na teoria de controle, essas características são representadas matematicamente pelos pólos da função de transferência de malha fechada. Pode-se entender os pólos como o "DNA" da dinâmica do sistema: sua localização em um plano complexo dita se a resposta será rápida ou lenta, se irá oscilar e, mais importante, se será estável. Portanto, o projeto de um controlador se resume a posicionar estrategicamente esses pólos em locais que garantam o desempenho desejado.

Para realizar essa tarefa, a Engenharia de Controle e Automação dispõe de ferramentas de análise e projeto consolidadas. Uma das mais poderosas é o método do Lugar Geométrico das Raízes (LGR). O LGR funciona como um "mapa" que mostra todas as possíveis localizações para os pólos de malha fechada que podem ser alcançadas ao se variar um único parâmetro do controlador, geralmente o seu ganho. Essa análise gráfica permite ao projetista prever e entender como o ajuste do controlador afetará a estabilidade e a resposta do sistema.

Uma vez que a análise indica onde os pólos devem estar, é preciso implementar um controlador com a arquitetura adequada para posicioná-los. A estrutura mais consagrada e versátil para esta finalidade é o controlador Proporcional-Integrativo-Derivativo (PID). Conforme descrito por [Ogata \(2010\)](#), o controlador PID atua sobre o erro do sistema de três formas complementares — uma ação Proporcional ao erro presente, uma Integral baseada no erro acumulado no passado e uma Derivativa que antecipa o erro futuro —, oferecendo uma solução robusta e eficaz para a grande maioria dos problemas de controle.

Este capítulo, portanto, apresentará a metodologia de projeto de controladores em duas etapas. Primeiramente, será detalhado o método do Lugar Geométrico das Raízes como ferramenta de análise para guiar o projeto. Em seguida, serão exploradas as estruturas e as ações de controle dos controladores da família PID, que serão a base para

o desenvolvimento prático do sistema de controle proposto neste trabalho.

2.4.1 O Método do Lugar Geométrico das Raízes (LGR)

O Lugar Geométrico das Raízes (LGR) constitui uma das ferramentas gráficas mais importantes da teoria de controle clássico. Desenvolvido inicialmente por Walter R. Evans em 1948, o método fornece um meio sistemático e visual para analisar a influência de parâmetros do sistema — em especial, o ganho de malha — sobre a localização dos polos de malha fechada (OGATA, 2010; DORF; BISHOP, 2018).

A essência do LGR reside na representação gráfica do movimento dos polos de malha fechada no plano complexo à medida que o ganho do controlador K é variado de zero ao infinito. Como os polos determinam a estabilidade, o tempo de resposta e as características transitórias de um sistema dinâmico, o LGR estabelece uma conexão direta entre estabilidade e desempenho. Dessa forma, o projetista pode avaliar como ajustes no ganho ou em parâmetros compensadores deslocam os polos, permitindo uma análise preditiva do comportamento do sistema (NISE, 2023).

Do ponto de vista matemático, considere um sistema de controle em malha fechada com função de transferência:

$$T(s) = \frac{KG(s)}{1 + KG(s)H(s)} \quad (2.26)$$

em que $G(s)$ representa a planta, $H(s)$ a função de realimentação e K o ganho ajustável. Os polos de malha fechada são obtidos pela solução da equação característica:

$$1 + KG(s)H(s) = 0. \quad (2.27)$$

Rearranjando, obtém-se:

$$KG(s)H(s) = -1. \quad (2.28)$$

Assim, para cada valor real e positivo de K , os polos se distribuem no plano- s de acordo com a condição de ângulo e de magnitude, o que define as trajetórias do lugar das raízes (CHEN, 2021).

A principal vantagem do LGR está em sua capacidade de integrar diferentes aspectos do projeto de controle:

- Estabilidade: se as trajetórias dos polos permanecem no semi-plano esquerdo, o sistema é considerado estável.
- Velocidade de resposta: a velocidade de resposta de um sistema é influenciada pela localização dos polos. Quanto mais distantes os polos estiverem do eixo imaginário, mais rápida será a resposta transitória, resultando em um sistema que atinge seu estado final de forma mais ágil.

- Oscilações: polos complexos conjugados indicam comportamento oscilatório, e a razão de amortecimento pode ser inferida pela posição relativa desses polos em relação ao eixo real. Quanto mais próximos os polos estiverem do eixo real, menor será a razão de amortecimento e, conseqüentemente, mais oscilações ocorrerão.
- Projeto de compensadores: o LGR permite visualizar como elementos adicionais, como polos e zeros de controladores PI, PD ou PID, podem moldar a trajetória dos polos e, conseqüentemente, melhorar o desempenho dinâmico.

Em geral, o LGR atua como um elo conceitual que conecta estabilidade, resposta transitória e ajustes de parâmetros de controle em uma única ferramenta gráfica. Por esse motivo, é amplamente utilizado tanto no ensino quanto na prática da engenharia de controle, servindo como uma ponte entre a modelagem matemática e o projeto prático de controladores (OGATA, 2010; NISE, 2023).

2.4.2 Controladores Proporcional, Integral e Derivativo (PID)

Os controladores Proporcional (P), Proporcional-Integral (PI) e Proporcional-Integral-Derivativo (PID) constituem a classe mais comum de controladores utilizados em sistemas de controle automático, devido à sua simplicidade estrutural e à eficácia na regulação de processos dinâmicos. A aplicação desses controladores está presente em diferentes setores, como sistemas industriais, automotivos, aeronáuticos e até mesmo em sistemas embarcados de baixo custo, configurando-se como a base da engenharia de controle aplicada (ÅSTRÖM; MURRAY, 2017; OGATA, 2010).

2.4.2.1 Controlador Proporcional (P)

O controlador proporcional ajusta a saída de controle de acordo com a magnitude do erro, definido como a diferença entre a referência e a saída do sistema. A equação no tempo é:

$$u(t) = K_p e(t) \quad (2.29)$$

em que $u(t)$ é o sinal de controle, $e(t)$ o erro e K_p o ganho proporcional.

No domínio de Laplace é:

$$\frac{U(s)}{E(s)} = K_p \quad (2.30)$$

O aumento de K_p reduz o erro em regime estacionário e acelera a resposta, mas pode introduzir oscilações e até instabilidade se o ganho for elevado. Controladores P são eficazes em sistemas onde a eliminação total do erro não é imprescindível, mas a simplicidade e a robustez são desejadas (OGATA, 2010; NISE, 2023).

2.4.2.2 Controlador Proporcional-Integral (PI)

O controlador PI introduz, além da ação proporcional, um termo integral que acumula o erro ao longo do tempo:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau \quad (2.31)$$

No domínio de Laplace, temos:

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \quad (2.32)$$

onde K_i é o ganho integral. O termo integral ($\frac{K_i}{s}$) garante a eliminação do erro em regime estacionário, tornando esse controlador amplamente empregado em processos industriais onde o erro permanente não é tolerado. Entretanto, o acréscimo da integral pode causar oscilações e aumentar o tempo de resposta, exigindo ajuste cuidadoso dos parâmetros (DORF; BISHOP, 2018; OGATA, 2010).

2.4.2.3 Controlador Proporcional-Integral-Derivativo (PID)

O controlador PID combina as três ações: proporcional, integral e derivativa:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.33)$$

No domínio de Laplace, temos:

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (2.34)$$

O termo derivativo (K_d) antecipa a tendência do erro, adicionando uma ação preditiva que melhora a estabilidade e reduz oscilações. Esse comportamento confere ao PID maior capacidade de ajuste fino, possibilitando redução do sobre-sinal (*overshoot*) e melhor desempenho transitório. Na prática, costuma-se empregar um filtro de primeira ordem na parte derivativa (conhecido como filtro N):

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + \frac{K_d N s}{s + N} \quad (2.35)$$

onde N é o parâmetro do filtro, geralmente escolhido grande ($N \gg 1$) para aproximar a ação derivativa ideal sem amplificação excessiva de ruído. Esse filtro evita que pequenas variações rápidas no sinal de erro comprometam a robustez do sistema (OGATA, 2010).

Os controladores P, PI e PID são amplamente aplicados devido ao equilíbrio entre simplicidade matemática, facilidade de implementação e desempenho satisfatório para uma ampla gama de sistemas dinâmicos. O PID, em especial, é considerado o “padrão

industrial” de controle, estando presente em mais de 90% das malhas de controle industriais em operação (ÅSTRÖM; HÄGGLUND, 2006). Contudo, o ajuste dos parâmetros K_p , K_i e K_d é uma tarefa crítica, sendo abordada por métodos como Ziegler-Nichols, Cohen-Coon e técnicas modernas de sintonia assistida por software.

3 Desenvolvimento

Este capítulo apresenta os procedimentos metodológicos adotados para o desenvolvimento e a validação da ferramenta interativa educacional, que constitui o objeto central deste estudo. Inicialmente, descrevem-se a abordagem de desenvolvimento, os recursos tecnológicos empregados e os métodos de validação aplicados. Ressalta-se que a revisão bibliográfica necessária para embasar a definição dos requisitos e o desenvolvimento da ferramenta foi apresentada no Capítulo 2.

No processo de elaboração deste trabalho, foram utilizadas ferramentas de inteligência artificial, como *Google Gemini* e *ChatGPT*, exclusivamente como instrumentos de apoio à escrita. Essas tecnologias foram aplicadas em tarefas específicas, tais como:

- **Aprimoramento textual:** reestruturação de parágrafos com o objetivo de aprimorar a clareza, a coesão e a fluidez da redação acadêmica;
- **Revisão gramatical e acadêmica:** identificação de inconsistências linguísticas, correção ortográfica e sugestão de vocabulário mais técnico e formal, em conformidade com a norma culta da língua;
- **Apoio à organização de ideias:** auxílio na estruturação de argumentos e na síntese de informações para a composição de determinadas seções do texto.

É importante destacar que o uso da inteligência artificial ocorreu unicamente como ferramenta de produtividade. Todo o material gerado foi submetido a criteriosa revisão, edição e validação pelo autor. A responsabilidade integral pelo conteúdo — incluindo a precisão técnica das informações, a originalidade, a coerência argumentativa e as conclusões apresentadas — é de exclusiva responsabilidade do autor.

3.1 Abordagem de Desenvolvimento

Para a construção da ferramenta, adotou-se a abordagem de prototipagem. Esta metodologia foi escolhida por sua agilidade e flexibilidade, características que se alinham com a proposta de um projeto de desenvolvimento rápido e focado na experiência do usuário.

3.2 Etapas do Desenvolvimento do Projeto

O processo de criação da ferramenta foi estruturado nas seguintes etapas:

3.2.1 Definição dos Requisitos

Na primeira etapa, definiram-se os requisitos funcionais da aplicação, estabelecendo as funcionalidades necessárias para que a ferramenta pudesse atender tanto a fins didáticos quanto a fins de análise prática em sistemas de controle. Entre os requisitos definidos, destacaram-se:

- Implementação de uma interface gráfica acessível via navegador;
- Capacidade de simulação de sistemas lineares invariantes no tempo (LTI) de primeira e segunda ordem;
- Implementação de controladores clássicos P, PI e PID;
- Disponibilização de representações gráficas, incluindo resposta ao degrau, resposta ao impulso e Lugar Geral das Raízes (LGR);
- Possibilidade de exportação dos resultados obtidos;

3.2.2 Escolha das Tecnologias e Ferramentas

A segunda etapa do desenvolvimento do trabalho consistiu na escolha da linguagem de programação *Python* e sua ferramenta *Streamlit*.

3.2.2.1 *Python*

A linguagem *Python* foi selecionada como base para o desenvolvimento da ferramenta proposta, em razão de suas características técnicas e de sua ampla aceitação nos meios acadêmico, científico e industrial. *Python* é uma linguagem de código aberto, gratuita e multiplataforma, o que assegura acessibilidade e elimina custos de licenciamento, frequentemente associados a *softwares* proprietários, como o *MATLAB*. Essas características são coerentes com os objetivos deste trabalho, que incluem a democratização do acesso a ferramentas de apoio ao ensino em engenharia de controle.

Entre as vantagens mais relevantes da linguagem, destaca-se sua sintaxe clara e intuitiva, que se aproxima do pensamento natural e reduz a complexidade no processo de desenvolvimento. Tal característica favorece não apenas a prototipagem rápida, mas também a manutenção e a expansão do código, permitindo que a comunidade acadêmica possa contribuir com melhorias futuras.

Outro aspecto essencial é o ecossistema robusto de bibliotecas científicas disponíveis, que conferem ao *Python* grande versatilidade no tratamento de problemas de engenharia. No presente trabalho, foram utilizadas bibliotecas consolidadas como:

- ***Python Control***: voltada para a modelagem, simulação e análise de sistemas dinâmicos e de controle;
- ***NumPy***: empregada em operações numéricas de alto desempenho, manipulação de vetores e matrizes, fundamentais para cálculos matemáticos em sistemas dinâmicos;
- ***SciPy***: utilizada em rotinas de análise matemática avançada, incluindo solução de equações diferenciais;
- ***Pandas***: aplicada à manipulação e organização de dados, viabilizando a estruturação dos resultados obtidos;
- ***Matplotlib***: destinada à geração de gráficos e representações visuais dos resultados, facilitando a interpretação dos fenômenos analisados.

A integração natural entre essas bibliotecas e o *Streamlit*, responsável pela construção da interface gráfica, possibilitou a criação de uma ferramenta robusta, interativa e didática. Essa sinergia entre simplicidade, flexibilidade e poder computacional fez do *Python* a linguagem mais adequada para o desenvolvimento da solução proposta neste trabalho.

Dessa forma, a escolha pelo *Python* não se deu apenas por sua popularidade, mas também por sua capacidade de equilibrar rigor científico, acessibilidade e aplicabilidade prática, atendendo plenamente aos requisitos técnicos e pedagógicos da ferramenta educacional desenvolvida.

3.2.2.2 *Streamlit*

A interface gráfica da ferramenta foi implementada utilizando o *Streamlit*, uma estrutura em *Python* que possibilita o desenvolvimento de aplicações *web* interativas de forma simples e acessível. Essa escolha se mostrou estratégica, pois a tecnologia permite criar aplicações funcionais sem a necessidade de conhecimento avançado em programação *web*, o que viabilizou concentrar esforços no conteúdo técnico relacionado à modelagem, simulação e análise de sistemas de controle.

O *Streamlit* se destaca por oferecer componentes prontos e interativos, como controles deslizantes, botões e caixas de seleção, que foram empregados no presente trabalho para o ajuste dinâmico dos parâmetros de controladores P, PI e PID. Essa funcionalidade possibilitou que os usuários manipulassem variáveis de forma intuitiva e observassem, em tempo real, os efeitos das alterações nas respostas do sistema, como as curvas de degrau, impulso e o Lugar Geométrico das Raízes.

Além da interatividade, a ferramenta se integra nativamente a bibliotecas científicas já consolidadas no ecossistema *Python*, como *NumPy*, *SciPy*, *Pandas* e *Matplotlib*, que

foram fundamentais para os cálculos matemáticos, análises de desempenho e geração de representações gráficas. Essa integração garantiu que a aplicação tivesse uma base sólida, combinando precisão científica com acessibilidade.

Outro aspecto relevante é a facilidade de disponibilização da aplicação. O *Streamlit* possibilita que a ferramenta seja executada localmente em qualquer computador com *Python* instalado. Isso amplia significativamente o alcance da solução, tornando-a acessível a estudantes, independentemente de infraestrutura laboratorial ou de softwares proprietários.

No contexto deste trabalho, o uso do *Streamlit* contribuiu para a criação de um ambiente didático e experimental, que promove uma aproximação entre a teoria e a prática no ensino de sistemas de controle. Dessa forma, a aplicação resultante não se limita a apresentar cálculos e gráficos, mas oferece uma experiência de aprendizado interativa e imersiva, favorecendo a compreensão de conceitos tradicionalmente abstratos e de difícil visualização no ensino convencional.

3.2.3 Estrutura da Ferramenta

A ferramenta desenvolvida neste projeto segue uma organização modular e escalável, facilitando manutenção, testes e evolução futura. Abaixo é detalhada a função de cada diretório e arquivo dentro da estrutura do projeto.

Arquivo Principal

O arquivo *app* é o núcleo da aplicação. A figura 7 a seguir mostra a importação de módulos e bibliotecas para inicializar a interface, coletar as configurações do sistema e do controlador PID definidas pelo usuário na barra lateral, e usar essas informações para construir o modelo matemático do sistema de controle. Em seguida, ele delega a análise (geração de gráficos e métricas) a módulos especializados e, por fim, exibe os resultados organizadamente nas abas da interface principal.

```
#Importação das bibliotecas
import streamlit as st
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
import os

# Importação dos módulos lógicos
from src.ui import render_sidebar, render_main_tabs
from src.models import get_plant_examples, parse_transfer_function
from src.controllers import create_pid_controller
from src.analysis_time import analyze_time_domain
from src.metrics import calculate_step_metrics
```

Figura 7 – Código de módulos e bibliotecas para importação. Fonte: Autor.

Src (Source)

O diretório *src* contém todo o código-fonte da ferramenta e é organizado em módulos lógicos:

- ***ui***: Responsável por renderizar todos os elementos visuais com os quais o usuário interage, como a barra lateral de configurações, os botões, os seletores de opções e a organização do conteúdo em abas (veja figura 8). Sua função é traduzir as funcionalidades do sistema em uma experiência de usuário clara e intuitiva, garantindo que a interação com a ferramenta seja simples e direta, sem que o usuário precise conhecer a lógica de programação subjacente.

```
# PID Controller Section
st.subheader("2. Controlador PID")

col1, col2 = st.columns(2)
with col1:
    kp = st.slider(
        "Kp (Proporcional)",
        min_value=0.0,
        max_value=50.0,
        value=1.0,
        step=0.1,
        help="Ganho proporcional - afeta a velocidade de resposta")

    ki = st.slider(
        "Ki (Integral)",
        min_value=0.0,
        max_value=50.0,
        value=0.0,
        step=0.1,
        help="Ganho integral - elimina erro estacionário")

with col2:
    kd = st.slider(
        "Kd (Derivativo)",
        min_value=0.0,
        max_value=10.0,
        value=0.0,
        step=0.01,
        help="Ganho derivativo - melhora estabilidade")

    n_filter = st.slider(
        "N (Filtro D)",
        min_value=1,
        max_value=100,
        value=10,
        help="Filtro do termo derivativo - reduz ruído")
```

Figura 8 – Código para criar os controles deslizantes de K_p , K_i , K_d e N . Fonte: Autor.

- ***models***: Sua principal função é processar e validar a função de transferência que

descreve o sistema a ser controlado, seja ela inserida manualmente ou selecionada a partir de exemplos pré-definidos. Essencialmente, este módulo garante que as especificações do sistema sejam corretamente convertidas em um modelo matemático válido, que servirá de base para todas as análises subsequentes realizadas pela aplicação.

- **controllers:** Com base nos ganhos (K_p , K_i , K_d) e no parâmetro do filtro derivativo (N) fornecidos pelo usuário, este módulo constrói a função de transferência que representa matematicamente o controlador. Ele encapsula a lógica do controle PID, permitindo que o arquivo principal, *app*, solicite a criação de um controlador de forma simplificada para então integrá-lo ao sistema completo.
- **analysis-time:** Recebe o modelo matemático completo do sistema de controle e executa simulações para gerar o gráfico da resposta ao degrau, uma das análises mais fundamentais em sistemas de controle. Este arquivo cuida de todos os cálculos e da plotagem visual dos resultados temporais, permitindo que o usuário observe como o sistema reage a um estímulo padrão.
- **root-locus** A partir dos dados da resposta ao degrau, ele calcula indicadores numéricos chave, como tempo de subida, sobressinal e tempo de acomodação. A função deste arquivo é traduzir o comportamento visual do gráfico de resposta em métricas objetivas e padronizadas, oferecendo uma avaliação precisa e técnica da eficiência e estabilidade do controlador projetado.
- **export-utils:** Gera um gráfico que ilustra como a estabilidade do sistema em malha fechada é afetada pela variação de um ganho geral. Essa visualização é crucial para entender a robustez do sistema e prever seu comportamento sob diferentes condições de operação, oferecendo uma perspectiva mais profunda sobre a dinâmica do controle para além da resposta a um único estímulo.
- **metrics:** Permite que o usuário salve os dados gerados, como gráficos e tabelas de métricas, em formatos de arquivo universais, como Excel. O propósito deste arquivo é estender a utilidade da ferramenta para além da simples visualização, permitindo que os resultados das simulações sejam documentados, compartilhados e utilizados em relatórios ou outras análises externas.

Examples

O arquivo **plants**, localizado no diretório *examples*, funciona como uma biblioteca de sistemas pré-configurados para a aplicação. Ele armazena uma coleção de sistemas a serem controlados, que são apresentadas ao usuário como exemplos de seleção rápida na interface. Isso permite que novos sistemas sejam adicionados ou modificados facilmente,

sem qualquer alteração na lógica principal do *app* ou nos arquivos de análise, garantindo que o software seja facilmente expansível e de simples manutenção.

Tests

- ***test-closed-loop***: Este arquivo de teste automatizado verifica a integridade da lógica de controle principal do sistema. Sua função é simular diversas configurações de controle para confirmar que o comportamento do sistema em malha fechada é estável e corresponde aos resultados esperados pela teoria, garantindo a confiabilidade da ferramenta.
- ***test-metrics***: Este módulo é focado em validar a precisão dos cálculos de desempenho da aplicação. Ele compara as métricas geradas pela ferramenta (como sobressinal e tempo de acomodação) com valores teóricos conhecidos para sistemas padrão, assegurando que os resultados numéricos exibidos para o usuário sejam corretos e confiáveis (veja a figura 9).

```
def test_second_order_system_metrics():
    """Test metrics calculation for a standard second order system"""

    # Second order system:  $wn^2 / (s^2 + 2*zeta*wn*s + wn^2)$ 
    wn = 2.0 # Natural frequency
    zeta = 0.5 # Damping ratio

    # Create transfer function
    num = [wn**2]
    den = [1, 2*zeta*wn, wn**2]
    tf = control.TransferFunction(num, den)

    # Calculate step response
    t = np.linspace(0, 10, 1000)
    t_step, y_step = control.step_response(tf, T=t)

    # Calculate metrics
    metrics = calculate_step_metrics(t_step, y_step, 1.0) # Unit step

    # Theoretical values for second order system
    theoretical_overshoot = 100 * np.exp(-zeta * np.pi / np.sqrt(1 - zeta**2))
    theoretical_peak_time = np.pi / (wn * np.sqrt(1 - zeta**2))
    theoretical_settling_time = 4 / (zeta * wn) # 2% criterion

    # Test with reasonable tolerances
    assert abs(metrics['overshoot'] - theoretical_overshoot) < 2.0
    assert abs(metrics['peak_time'] - theoretical_peak_time) < 0.1
    assert abs(metrics['settling_time'] - theoretical_settling_time) < 0.5
    assert abs(metrics['steady_state_error']) < 0.01 # Should be nearly zero
    assert metrics['final_value'] == pytest.approx(1.0, rel=0.01)
```

Figura 9 – Código para comparação de métricas de desempenho para Sistemas de Segunda Ordem. Fonte: Autor.

Arquivos de configuração e documentação

- ***README.md***: Este arquivo é o guia rápido do projeto. Ele contém as instruções essenciais de instalação e execução da ferramenta, servindo como um manual prático para garantir que qualquer usuário possa operar o software corretamente.
- ***pyproject.toml***: Lista todas as dependências Python necessárias para execução da ferramenta, permitindo que o ambiente seja reproduzido facilmente via *pip install ..*

Considerações sobre a Estrutura

A organização segue boas práticas de desenvolvimento e programação de *software* em *Python* priorizando código limpo e legível, com divisão e organização, clareza e manutenibilidade. A separação em módulos facilita testes e evolução da aplicação sem comprometer a estabilidade de partes existentes do sistema.

3.2.4 Resultados da Ferramenta Desenvolvida

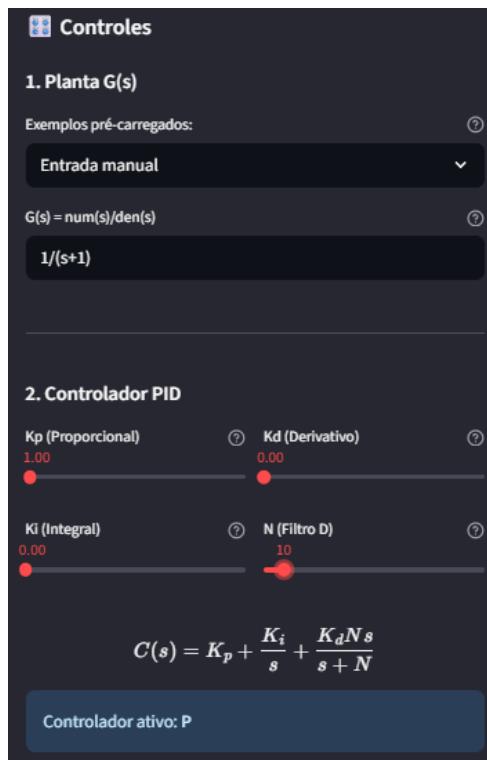
Nesta seção são apresentados os principais recursos implementados na ferramenta que possui uma interface web interativa construída em Streamlit, permitindo:

- Inserção de funções de transferência personalizadas pelo usuário;
- Seleção de controladores P, PI ou PID, com ajuste dinâmico de parâmetros;
- Geração automática de gráficos (resposta ao degrau, impulso e LGR);
- Cálculo e exibição de métricas de desempenho (tempo de acomodação, overshoot, etc.);
- Exportação de resultados em formato de relatório (.csv, .png, .pdf).

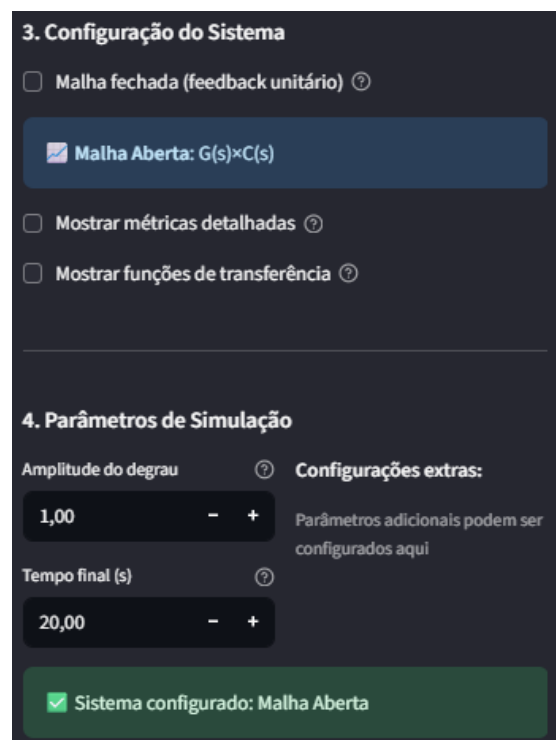
As figuras 10 a 16, apresentadas a seguir, ilustram a interface da ferramenta desenvolvida. São detalhados os seus principais componentes, como os campos de entrada, os controles deslizantes para ajuste de parâmetros, as opções de configuração do sistema em malha aberta e fechada, a exibição de métricas detalhadas, os parâmetros de simulação, além das áreas de visualização e exportação de gráficos.



Figura 10 – Interface inicial da ferramenta. Fonte: Autor.



(a) Painel de Controles da Planta e do Controlador PID.



(b) Painel de Configuração do Sistema e Parâmetros de Simulação.

Figura 11 – Interface principal da ferramenta de simulação desenvolvida no trabalho. Fonte: Autor.

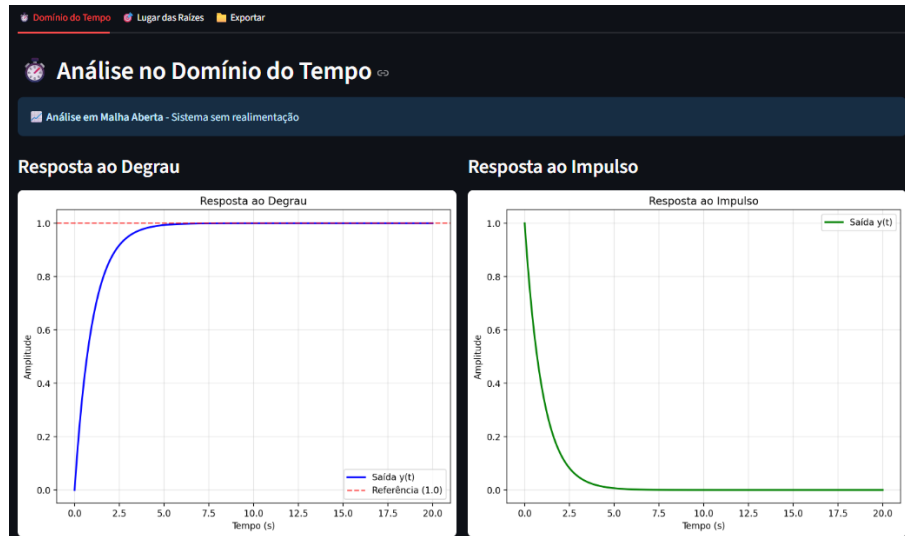


Figura 12 – Análise gráfica do domínio do tempo. Fonte: Autor.

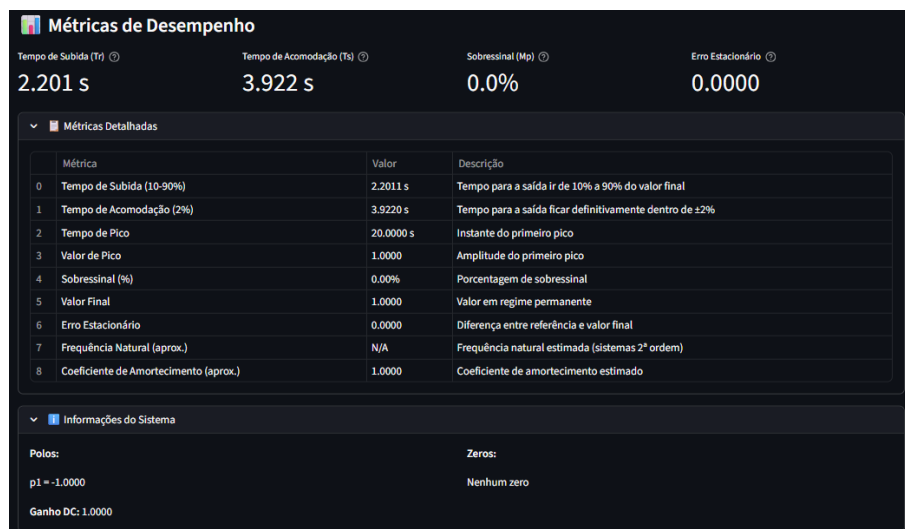


Figura 13 – Métricas de desempenho e informações do sistema. Fonte: Autor.

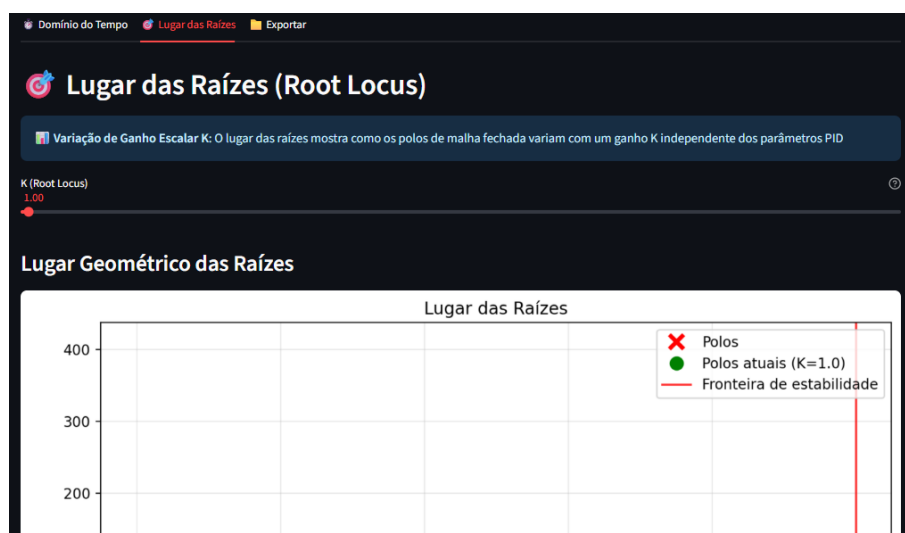


Figura 14 – Gráfico do Lugar Geométrico das Raízes. Fonte: Autor.

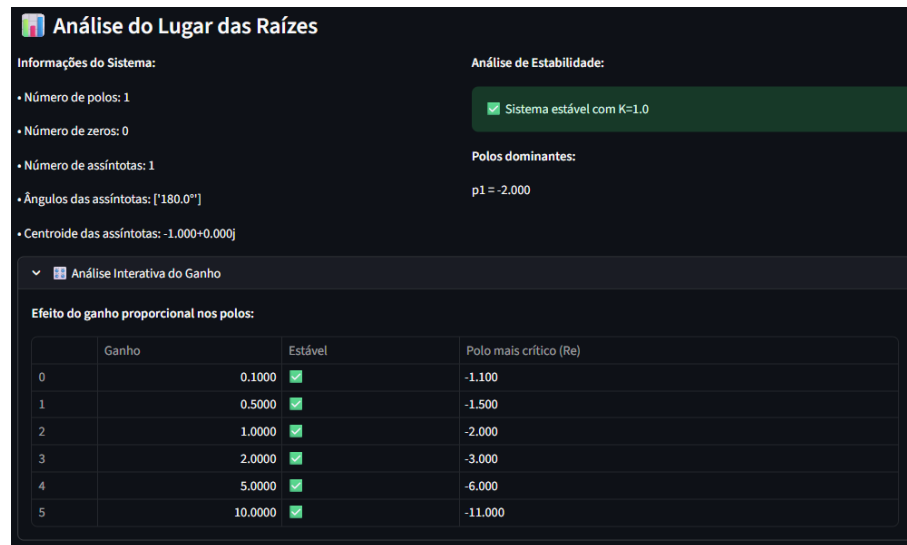


Figura 15 – Análise do Lugar das Raízes. Fonte: Autor.



Figura 16 – Exportação de resultados. Fonte: Autor.

3.2.5 Testes e Validação

Na terceira etapa, foram realizados quatro testes e suas validações com o objetivo de assegurar a precisão e a confiabilidade da ferramenta desenvolvida.

3.2.6 Documentação e Disponibilização da Ferramenta

Por fim, a quarta etapa consistiu na documentação e disponibilização da ferramenta. Foi elaborado um guia de uso em formato digital, detalhando as funcionalidades, os requisitos de instalação e exemplos de aplicação. O código-fonte foi disponibilizado via [GitHub](#), visando garantir a transparência científica, a reprodutibilidade dos resultados e a possibilidade de expansão da ferramenta em trabalhos futuros.

4 Resultados

A escolha dos quatro sistemas utilizados para a validação da ferramenta teve como objetivo abranger um conjunto de desafios clássicos e didáticos da Engenharia de Controle e Automação. Cada sistema foi selecionado para testar uma funcionalidade específica da plataforma em um cenário representativo, conforme detalhado a seguir:

- Sistema 1 – Desempenho de Resposta Lenta: Representa uma planta estável, porém superamortecida e com um tempo de resposta elevado. O objetivo neste caso é demonstrar a capacidade da ferramenta em projetar controladores que melhorem a velocidade e a precisão do sistema.
- Sistema 2 – Comportamento Oscilatório: Trata-se de um sistema marginalmente estável que, sem controle, oscila continuamente. Foi escolhido para validar a aplicação de ações de controle que introduzem amortecimento e estabilizam a resposta.
- Sistema 3 – Erro em Regime Estacionário: Este sistema, embora estável, não é capaz de seguir uma referência constante sem um erro significativo. Sua finalidade é testar a eficácia do controle, especialmente da ação integral, na eliminação do erro em regime permanente.
- Sistema 4 – Sistema Instável: Corresponde a uma planta com comportamento divergente em malha aberta. Foi selecionado como o desafio final para a ferramenta, demonstrando sua capacidade de projetar um controlador que estabiliza completamente um sistema intrinsecamente instável.

Em conjunto, esses quatro casos de estudo fornecem uma validação robusta da ferramenta, cobrindo os principais objetivos do projeto de sistemas de controle.

4.1 Sistema 1 - Desempenho de Resposta Lenta.

O sistema representado pela função de transferência

$$G(s) = \frac{1}{s^2 + 3s + 2} \quad (4.1)$$

apresenta, em malha aberta, um comportamento estável, porém lento e superamortecido. Pela figura 17, a resposta ao degrau unitário mostra que o sistema leva um tempo considerável para se aproximar da referência, não atingindo exatamente o valor de 1, o que caracteriza a presença de erro em regime permanente. Além disso, o tempo de acomodação

é elevado, o que demonstra a dificuldade da planta em acompanhar sinais de referência de maneira satisfatória (ver figura 18).

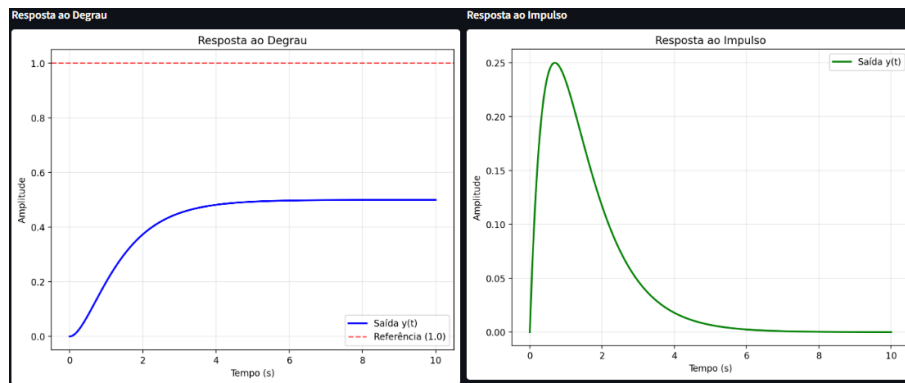


Figura 17 – Respostas ao Degrau e ao Impulso do Sistema 1. Fonte: Autor.



Figura 18 – Métricas de desempenho do Sistema 1. Fonte: Autor.

Ao aplicar-se o controlador PID ajustado na equação 4.1 com $K_p = 4$, $K_i = 5$, $K_d = 1$ e $N = 20$ temos:

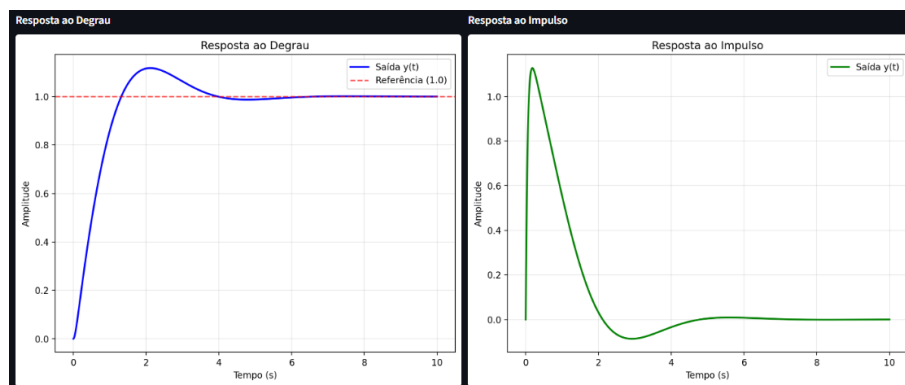


Figura 19 – Respostas ao Degrau e ao Impulso do Sistema 1 PID. Fonte: Autor.

Métricas de Desempenho			
Tempo de Subida (Tr) ⓘ	Tempo de Acomodação (Ts) ⓘ	Sobressinal (Mp) ⓘ	Erro Estacionário ⓘ
0.951 s	3.574 s	11.8%	0.0001
▼ Métricas Detalhadas			
	Métrica	Valor	Descrição
0	Tempo de Subida (10-90%)	0.9510 s	Tempo para a saída ir de 10% a 90% do valor final
1	Tempo de Acomodação (2%)	3.5736 s	Tempo para a saída ficar definitivamente dentro de ±2%
2	Tempo de Pico	2.1121 s	Instante do primeiro pico
3	Valor de Pico	1.1177	Amplitude do primeiro pico
4	Sobressinal (%)	11.79%	Porcentagem de sobressinal
5	Valor Final	0.9999	Valor em regime permanente
6	Erro Estacionário	0.0001	Diferença entre referência e valor final
7	Frequência Natural (aprox.)	1.7992 rad/s	Frequência natural estimada (sistemas 2ª ordem)
8	Coefficiente de Amortecimento (aprox.)	0.5627	Coefficiente de amortecimento estimado

Figura 20 – Métricas de desempenho do Sistema 1 PID. Fonte: Autor.

Como mostrado na figura 19, o desempenho do sistema foi significativamente aprimorado. A resposta passou a ser muito mais rápida, com tempos de subida e acomodação reduzidos em comparação à planta original. O erro em regime permanente foi praticamente eliminado, garantindo que a saída atingisse o valor de referência com precisão. Embora tenha surgido um sobressinal, este se manteve em valores controlados, sem comprometer a estabilidade (ver figura 20). Dessa forma, o controlador transformou um sistema inicialmente lento e impreciso em uma dinâmica adequada para fins de controle.

4.2 Sistema 2 - Comportamento Oscilatório.

O segundo sistema, descrito por

$$G(s) = \frac{1}{s^2 + 9} \quad (4.2)$$

é marginalmente estável. Pela figura 21, a resposta ao degrau evidencia oscilações indefinidas, que não se atenuam com o tempo, impossibilitando o rastreamento de referências constantes. Em malha aberta, portanto, o sistema não possui estabilidade prática (ver figura 22) e não pode ser utilizado em aplicações reais.

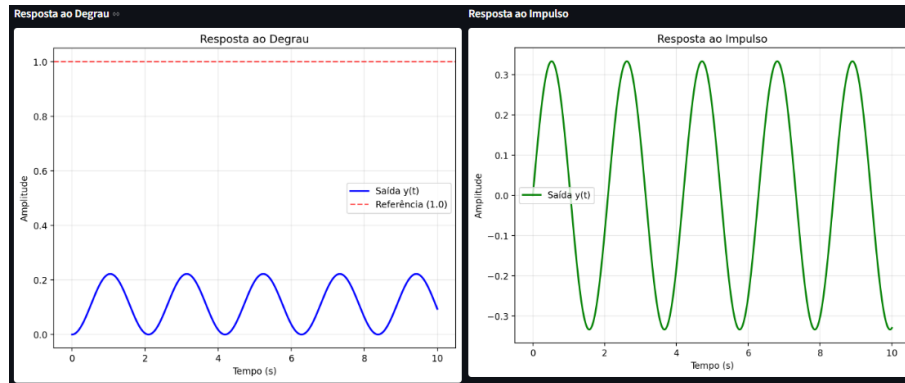


Figura 21 – Respostas ao Degrau e ao Impulso do Sistema 2. Fonte: Autor.



Figura 22 – Métricas de desempenho do Sistema 2. Fonte: Autor.

Ao aplicar-se o controlador PID ajustado na equação 4.2 com $K_p = 10$, $K_i = 4$, $K_d = 2$ e $N = 20$ temos:

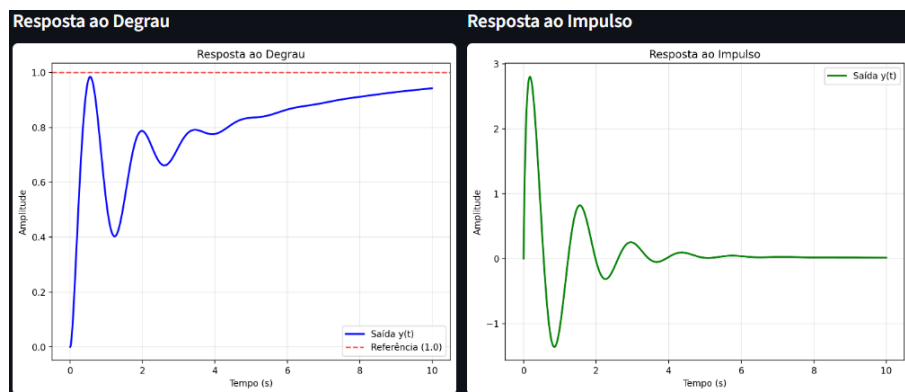


Figura 23 – Respostas ao Degrau e ao Impulso do Sistema 2 PID. Fonte: Autor.



Figura 24 – Métricas de desempenho do Sistema 2 PID. Fonte: Autor.

Observa-se que com a aplicação do controlador PID, pela figura 23, o comportamento da planta foi estabilizado. As oscilações permanentes desapareceram, dando lugar a uma resposta estável, que converge para a referência em tempo finito. Além disso, o erro estacionário foi eliminado e o tempo de resposta tornou-se mais rápido. Houve a presença de sobressinal, mas em valores aceitáveis (ver figura 24). O controlador foi, portanto, responsável por conferir estabilidade e precisão a uma planta que, em sua forma original, era incapaz de atingir a referência.

4.3 Sistema 3 - Erro em Regime Estacionário.

O terceiro sistema testado é representado por

$$G(s) = \frac{1}{s(s+2)} \quad (4.3)$$

Em malha aberta, como demonstra a figura 25, a resposta ao degrau não se estabiliza, mas cresce de forma indefinida, apresentando comportamento de rampa. Esse fenômeno indica a presença de erro estacionário infinito, o que inviabiliza o uso do sistema em aplicações que demandem rastreamento de sinais constantes (ver figura 26).

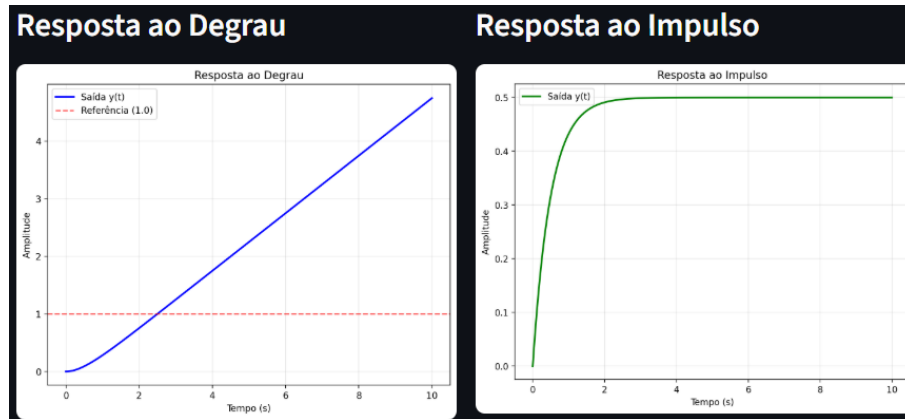


Figura 25 – Respostas ao Degrau e ao Impulso do Sistema 3. Fonte: Autor.



Figura 26 – Métricas de desempenho do Sistema 3. Fonte: Autor.

Ao aplicar-se o controlador PID ajustado na equação 4.3 com $K_p = 9$, $K_i = 0.5$, $K_d = 3$ e $N = 20$ temos:

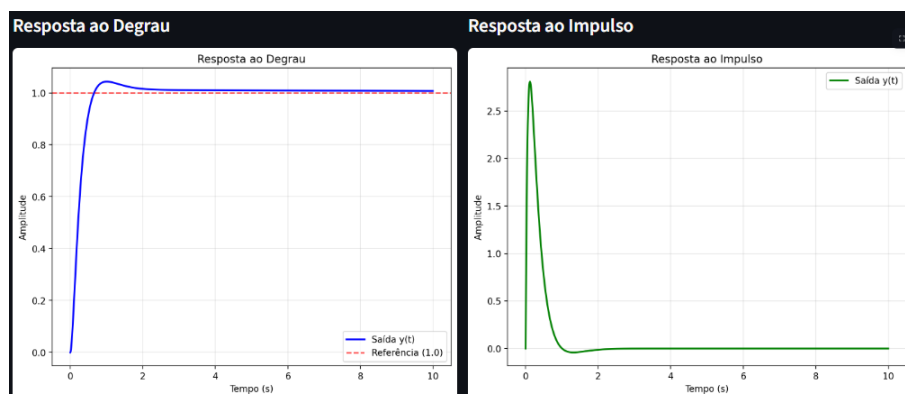


Figura 27 – Respostas ao Degrau e ao Impulso do Sistema 3 PID. Fonte: Autor.



Figura 28 – Métricas de desempenho do Sistema 3 PID. Fonte: Autor.

Observando a figura 27, nota-se que o sistema passou a acompanhar a referência de forma satisfatória. A ação integral foi capaz de eliminar o erro em regime estacionário, enquanto a ação derivativa contribuiu para melhorar a estabilidade transitória. A resposta obtida apresentou um tempo de subida reduzido, precisão elevada e apenas um pequeno sobressinal (ver figura 28), demonstrando que o controlador conseguiu transformar a planta em um sistema adequado para controle de posição e aplicações práticas.

4.4 Sistema 4 - Sistema Instável.

O quarto sistema analisado é descrito pela função de transferência

$$G(s) = \frac{1}{s - 1} \quad (4.4)$$

e possui um polo no semiplano direito do plano-s ($s = +1$), sendo, portanto, instável em malha aberta. Pela figura 29, sua resposta ao degrau diverge exponencialmente, crescendo indefinidamente, o que inviabiliza qualquer aplicação sem a presença de uma ação de controle (ver figura 30).

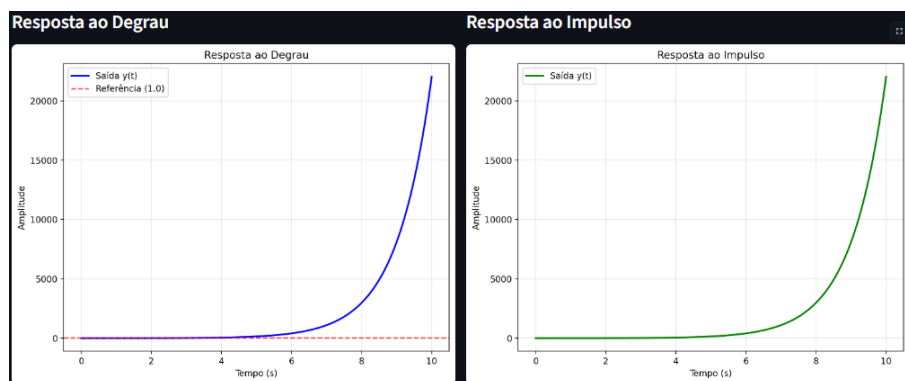


Figura 29 – Respostas ao Degrau e ao Impulso do Sistema 4. Fonte: Autor.



Figura 30 – Métricas de desempenho do Sistema 4. Fonte: Autor.

Ao aplicar-se o controlador PID ajustado na equação 4.4 com $K_p = 5$, $K_i = 8$, $K_d = 0.5$ e $N = 20$ temos:

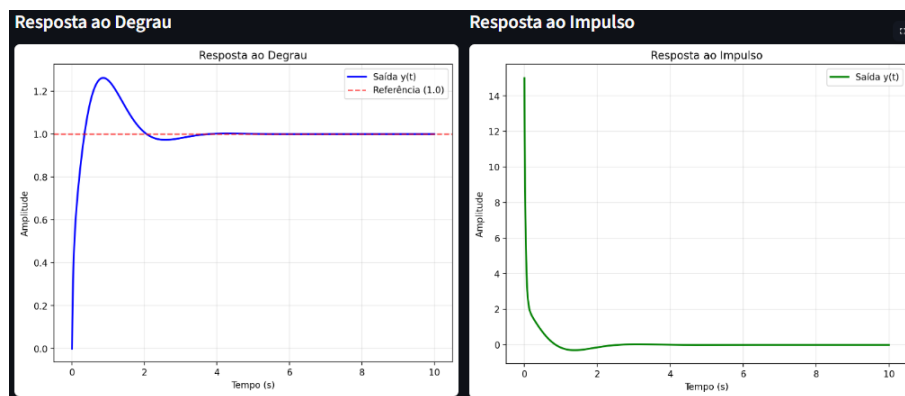


Figura 31 – Respostas ao Degrau e ao Impulso do Sistema 4 PID. Fonte: Autor.



Figura 32 – Métricas de desempenho do Sistema 4 PID. Fonte: Autor.

Após a aplicação do controlador PID ajustado, observando a figura 31 uma mudança drástica no comportamento do sistema. A planta tornou-se estável em malha fechada, apresentando resposta finita e convergente para a referência. O erro estacionário foi eliminado pela ação integral, enquanto a ação proporcional garantiu um tempo de resposta adequado. Apesar da presença de sobressinal moderado (ver figura 32), o desempenho final foi bastante satisfatório, evidenciando que o controlador foi capaz de transformar uma planta originalmente instável em um sistema estável e funcional.

5 Considerações finais

Como considerações finais, conclui-se que, este trabalho apresenta o desenvolvimento e a validação de uma ferramenta de *software* interativa, projetada para fins educacionais na área de sistemas de controle. Utilizando a linguagem de programação *Python* e um conjunto de bibliotecas consolidadas, como *Python Control*, *NumPy*, *SciPy*, *Pandas* e a estrutura *Streamlit*, foi criada uma plataforma que permite ao usuário analisar o comportamento de sistemas e projetar controladores de maneira prática. A ferramenta possibilita ainda a exportação dos resultados obtidos, consolidando-se como um recurso completo para o estudo aplicado.

O *software* desenvolvido transcende a teoria ao oferecer um ambiente de simulação que torna conceitos abstratos mais compreensíveis por meio da interatividade e da visualização de dados. Como uma solução de código aberto, acessível e didática, a ferramenta oferece suporte direto ao processo de ensino-aprendizagem, servindo como uma ponte entre o conhecimento teórico e a aplicação prática.

5.1 Trabalhos Futuros

Para trabalhos futuros, podem ser consideradas as seguintes melhorias:

- A ampliação do tempo de simulação nos gráficos, de modo a permitir a visualização completa do processo de estabilização do sistema;
- A implementação de análises no domínio da frequência;
- A possibilidade de salvar e carregar projetos, incluindo ganhos e modelos previamente configurados;
- A inclusão de sistemas discretos, possibilitando uma análise mais abrangente;
- A adaptação da ferramenta para um aplicativo móvel, em formato *apk*;
- A publicação da ferramenta em servidores na nuvem;

Referências

- ÅSTRÖM, Karl J.; MURRAY, Richard M. *Feedback Systems: An Introduction for Scientists and Engineers*. 2nd: Princeton University Press, 2017. Citado 6 vezes nas páginas 9, 12–14, 27.
- ÅSTRÖM, Karl Johan; HÄGGLUND, Tore. *Advanced PID Control*. Research Triangle Park, NC: ISA - The Instrumentation, Systems, e Automation Society, 2006. Citado 1 vez na página 29.
- BAZANELLA, A. S.; SILVA JR., J. M. G. *Sistemas de Controle: princípios e métodos de projeto*. Porto Alegre: Editora da UFRGS, 2005. Citado 0 vez nas páginas 13, 14.
- CHEN, Chi-Tsong. *Linear System Theory and Design*. 5th. Oxford: Oxford University Press, 2021. Citado 1 vez na página 26.
- DORF, Richard C.; BISHOP, Robert H. *Sistemas de Controle Modernos*. 13. ed. Rio de Janeiro: LTC, 2018. Citado 9 vezes nas páginas 9, 12, 15, 21, 24, 26, 28.
- FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. *Feedback Control of Dynamic Systems*. 8. ed. Hoboken, NJ: Pearson, 2019. Citado 1 vez na página 16.
- GARCÍA, C.; PONSART, J. C.; THEILLIOL, D. Predictive control and steady-state tracking performance: a survey. *Annual Reviews in Control*, v. 51, p. 134–146, 2021. Citado 1 vez na página 24.
- KUO, Benjamin C.; GOLNARAGHI, Farid. *Automatic Control Systems*. 9. ed. Hoboken, NJ: Wiley, 2014. Citado 2 vezes na página 13.
- MATHWORKS INC. *MATLAB and Simulink Student Version*. 2020. Disponível em: <https://www.mathworks.com>. Acesso em: 20 ago. 2025. Citado 1 vez na página 9.
- NISE, N. S. *Engenharia de Sistemas de Controle*. 8. ed. Rio de Janeiro: LTC, 2023. Citado 15 vezes nas páginas 9, 12–16, 23–27.
- OGATA, Katsuhiko. *Engenharia de Controle Moderno*. 5. ed. São Paulo: Pearson Prentice Hall, 2010. Citado 20 vezes nas páginas 9, 12–16, 18, 19, 21–28.
- SILVA, J. R.; COSTA, E. F.; LIMA, A. M. N. Analysis of steady-state error in discrete-time control systems. *Journal of Control, Automation and Electrical Systems*, v. 30, n. 5, p. 721–732, 2019. Citado 1 vez na página 24.
- ZHOU, H.; WANG, Y. Advanced error compensation in nonlinear control systems: steady-state accuracy improvement. *IEEE Transactions on Automatic Control*, v. 67, n. 8, p. 4123–4136, 2022. Citado 1 vez na página 25.

APÊNDICE A – Código utilizado

O código utilizado no desenvolvimento deste trabalho, foi armazenado em um repositório da ferramenta *Github* e pode ser acessado através do link:

- Ferramenta interativa educacional para análise de sistemas de controle e projeto de controladores PID (<https://github.com/ViniciusMendes98/ControlInsight>)