UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE TUCURUÍ
FACULDADE DE ENGENHARIA ELÉTRICA

OTIMIZAÇÃO EVOLUTIVA DE CONTROLADORES PID PARA BANCADAS MOTOR-GERADOR UTILIZANDO ALGORITMOS GENÉTICOS E PYGAD

DIEGO ANTONIO SILVA DE JESUS

Tucuruí-PA
2023

UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE TUCURUÍ
FACULDADE DE ENGENHARIA ELÉTRICA

DIEGO ANTONIO SILVA DE JESUS

OTIMIZAÇÃO EVOLUTIVA DE CONTROLADORES PID PARA BANCADAS MOTOR-GERADOR UTILIZANDO ALGORITMOS GENÉTICOS E PYGAD

Trabalho de conclusão de curso apresentado ao colegiado da Faculdade de Engenharia Elétrica, do Campus Universitário de Tucuruí, da Universidade Federal do Pará, como requisito necessário para obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Raphael Barros Teixeira

Tucuruí-PA
2023

UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE TUCURUÍ
FACULDADE DE ENGENHARIA ELÉTRICA

OTIMIZAÇÃO EVOLUTIVA DE CONTROLADORES PID PARA BANCADAS MOTOR-GERADOR UTILIZANDO ALGORITMOS GENÉTICOS

AUTOR: **DIEGO ANTONIO SILVA DE JESUS**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À BANCA EXAMINADORA APROVADA PELO COLEGIADO DA FACULDADE DE ENGENHARIA ELÉTRICA, SENDO JULGADO APROVADO COM CONCEITO EXCELENTE.

BANCA EXAMINADORA:

Documento assinado digitalmente
gov.br
RAPHAEL BARROS TEIXEIRA
Data: 07/03/2024 14:25:04-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Raphael Barros Teixeira
Orientador / UFPA-CAMTUC-FEE

Documento assinado digitalmente
gov.br
CLEISON DANIEL SILVA
Data: 07/03/2024 09:58:30-0300
Verifique em https://validar.iti.gov.br

Membro 1 / UFPA-CAMTUC-FEE

Documento assinado digitalmente
gov.br
VITOR DA SILVA JORGE
Data: 07/03/2024 09:30:45-0300
Verifique em https://validar.iti.gov.br

Prof. Vitor da Silva Jorge
Membro 2 / UFPA-CAMTUC-FEE

Dedico este trabalho a todos os profissionais da área de Engenharia Elétrica que se dedicam diariamente a encontrar soluções para os mais diversos desafios da nossa sociedade.
Seja na geração de energia, na automação de processos, no desenvolvimento de novas tecnologias ou na pesquisa científica, cada um de vocês contribui de forma significativa para o progresso da humanidade.
Que este trabalho possa ser mais uma pequena contribuição para o avanço da engenharia elétrica e inspire novas ideias e projetos para o bem comum.

AGRADECIMENTOS

Primeiramente, gostaria de expressar minha profunda gratidão a Deus por me conceder saúde, força e sabedoria durante todo o processo de elaboração deste trabalho acadêmico.

Agradeço imensamente a minha esposa Carolina, que sempre esteve ao meu lado, me apoiando e me incentivando em cada etapa deste projeto. Sem sua paciência, amor e dedicação, este trabalho não teria sido possível.

Quero agradecer também à minha mãe Ivanilde e à minha irmã Daisirée pelo constante incentivo, apoio e amor incondicional que me deram. Sem dúvida, são as pessoas mais importantes da minha vida.

Não posso deixar de agradecer aos meus colegas Telson e Kayse, pelo incentivo constante, por compartilhar conhecimento e experiências durante todo o processo de aprendizada nesta universidade.

Gostaria de agradecer, em especial, ao meu Orientador Prof. Dr. Raphael Barros Teixeira pela sua orientação, paciência e dedicação. Seu conhecimento, conselhos e sugestões foram essenciais para o sucesso deste trabalho.

Por fim, agradeço à Universidade Federal do Pará (UFPA) pela oportunidade de realizar este trabalho acadêmico e pela excelente formação acadêmica que recebi ao longo de todos estes anos.

"Não há inovação sem engenharia elétrica.
Ela é a base que sustenta a evolução tecnológica
e nos permite transformar ideias em realidade."
(Bill Gates)

Resumo

Este trabalho de conclusão de curso propõe uma abordagem inovadora para o controle de uma bancada motor-gerador, integrando técnicas avançadas de otimização, modelagem e controle. O estudo utiliza algoritmos genéticos (AGs) desenvolvidos com a biblioteca PyGad para otimização offline na sintonia dos controladores Proporcional-Integral (PI) e Proporcional-Integral-Derivativo (PID). A obtenção de dados da bancada é efetuada mediante a execução de um código Python dedicado, ao passo que um segundo código é elaborado para a criação do modelo, empregando a abordagem SINDy (Identificação Esparsa de Dinâmicas Não Lineares). Esse modelo serve como base para o desenvolvimento dos controladores. Dois conjuntos de códigos são implementados para os controladores PI e PID. O primeiro conjunto, de caráter offline, utiliza AG com PyGad para otimização dos parâmetros dos controladores. O segundo conjunto é de natureza online e tem como função a transmissão do controlador obtido com a otimização para a bancada motor-gerador em tempo real. O estudo aborda aspectos teóricos e práticos, fornecendo uma análise aprofundada dos resultados obtidos com a implementação dos controladores PI e PID, comparando o desempenho dos dois sintonizadores. Além disso, são apresentadas contribuições significativas no contexto de controle de sistemas dinâmicos, explorando a eficácia da integração de técnicas modernas para otimização e modelagem.

Palavras Chave: Controles PI e PID, Algoritmos Genéticos PyGad, Bancada Motor-Gerador, Otimização, Modelagem de Sistemas Dinâmicos SINDy.

Abstract

This undergraduate thesis proposes an innovative approach to the control of a motor-generator test bench, integrating advanced optimization, modeling, and control techniques. The study utilizes genetic algorithms (GAs) developed with the PyGad library for offline optimization in tuning Proportional-Integral (PI) and Proportional-Integral-Derivative (PID) controllers. Data collection from the test bench is performed through a specific code, while another code is developed for building the model using the Sparse Identification of Nonlinear Dynamics (SINDy) technique. This model serves as a foundation for the development of controllers. Two sets of codes are implemented for PI and PID controllers. The first set, of an offline nature, employs GAs with PyGad for optimizing controller parameters. The second set is online and functions to transmit the controller obtained through optimization to the motor-generator test bench in real-time. The study addresses theoretical and practical aspects, providing an in-depth analysis of the results obtained with the implementation of PI and PID controllers, comparing the performance of the two tuning methods. Additionally, significant contributions are presented in the context of dynamic systems control, exploring the effectiveness of integrating modern techniques for optimization and modeling.

Keywords: PI and PID controllers, PyGad genetic algorithms, Motor-generator bench, Optimization, SINDy dynamic systems modeling.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

1.1 Motivação e Contexto

A implementação de controle automático em sistemas dinâmicos representa um avanço significativo na automação de processos industriais, sistemas de navegação e diversas outras áreas. No entanto, essa transição para sistemas automáticos não está isenta de desafios, que vão desde questões técnicas até considerações práticas. Neste contexto, é fundamental analisar e compreender esses desafios para garantir a eficácia e a segurança desses sistemas.

Um dos desafios fundamentais está relacionado à complexidade dos sistemas controlados. Conforme destaca (OGATA, 2010), sistemas dinâmicos mais complexos apresentam não linearidades e interações não triviais, o que pode comprometer a estabilidade e a resposta dinâmica dos controladores automáticos. A necessidade de modelagem precisa e a consideração de não idealidades tornam-se essenciais para lidar com essa complexidade.

Outro desafio crítico diz respeito à robustez dos controladores automáticos em face de variações no ambiente e nos parâmetros do sistema. Como ressaltado por (FRANKLIN; POWELL; EMAMI-NAEINI, 2015), incertezas e mudanças nos parâmetros do sistema podem comprometer o desempenho do controle automático, exigindo estratégias robustas de projeto que possam lidar com essas variações sem comprometer a estabilidade e a performance do sistema.

A implementação de controles automáticos também enfrenta desafios específicos em ambientes industriais. Segundo (DORF; BISHOP, 2016), a presença de ruídos, a necessidade de resposta rápida a perturbações e a coexistência com sistemas manuais são aspectos a serem considerados. A integração eficiente entre sistemas automáticos e manuais, garantindo a segurança e a colaboração harmoniosa, é uma tarefa não trivial.

Em síntese, a implementação de controles automáticos em sistemas de controle enfrenta desafios que vão desde a complexidade intrínseca dos sistemas dinâmicos até questões práticas como robustez e integração industrial. A abordagem desses desafios requer uma combinação de modelagem avançada, estratégias robustas de projeto e medidas eficazes de segurança, visando

garantir a eficácia e a confiabilidade desses sistemas em ambientes diversos e dinâmicos.

1.2 Justificativa

Em face desses desafios, a implementação bem-sucedida de controles automáticos requer uma abordagem holística, envolvendo expertise em modelagem, algoritmos de controle, tecnologias de sensoriamento e atenção às condições operacionais do sistema. A pesquisa contínua e o desenvolvimento de novas técnicas são essenciais para superar os desafios emergentes à medida que os sistemas se tornam mais complexos e interconectados.

Apesar dos desafios práticos e teóricos, estabelecer uma teoria de controle aplicável universalmente enfrenta dificuldades. A presença simultânea de dinâmicas não modeladas e a necessidade de robustez cria obstáculos, tornando inviável uma solução válida dentro da estrutura de um controlador baseado em modelo convencional. Na prática, quanto mais preciso for o modelo, maior será o esforço do sistema de controle. Controladores baseados em modelos possuem como fundamento o princípio da certeza de equivalência, logo se o modelo em que for baseado não possuir semelhança a planta, o controlador não terá um bom funcionamento (HOU; WANG, 2013).

O desenvolvimento de modelos eficientes e precisos para sistemas dinâmicos é um desafio reconhecido na área de controle, como apontado por (CALDEIRA, 2020). O autor destaca a falta de um procedimento capaz de lidar efetivamente com complexidades, especialmente aquelas de ordem elevada, para as quais soluções ainda não foram definidas. Nesse contexto, a utilização dessas complexidades como referência para sistemas de controle é considerada inviável. Além disso, (CALDEIRA, 2020) ressalta a importância de considerar a persistência de excitação na entrada, pois a ausência de impulsos persistentemente excitantes impede a construção de modelos precisos, levando a resultados teóricos baseados em modelos que resultam em sistemas de malha fechada cuja estabilidade e convergência não podem ser asseguradas.

Relacionando-se a essa discussão sobre controle, o controle ótimo surge como uma técnica valiosa para otimizar sistemas dinâmicos. Conforme (BRYSON; HO, 1999), uma abordagem comum para implementar o controle ótimo é por meio do AG (algoritmo genético). Este método demonstra eficiência ao encontrar soluções ótimas para problemas complexos, contribuindo para a otimização da dinâmica do sistema, aspecto fundamental no campo do controle de sistemas dinâmicos. Assim, ao considerar os desafios ressaltados por (CALDEIRA, 2020) na construção de modelos precisos, o uso do controle ótimo, especialmente por meio de AGs, destaca-se como uma abordagem promissora na busca por soluções eficientes e ótimas em sistemas dinâmicos complexos.

Para implementação do sistema de controle, a definição de um modelo matemático do sistema dinâmico a ser controlado se faz necessário. Isso se tornará possível através da criação do `Fitness`, neste trabalho chamado de função custo, que avaliará o desempenho do sistema e fornecerá feedback para o AG (KIRK, 2004).

A função de avaliação é concebida para minimizar um custo específico, como, por exemplo, reduzir a discrepância entre a saída do sistema controlado e uma referência desejada. O AG é então empregado para otimizar os parâmetros do controlador com o objetivo de minimizar essa função custo, conforme destacado por (GOLDBERG, 2010).

1.3 Objetivos da Pesquisa

1.3.1 Objetivo Geral

Desenvolver uma estratégia de controle eficiente para a bancada motor-gerador, combinando técnicas de controle PI e PID otimizadas por AG, bem como a utilização de modelos `SINDy` para representação e predição do comportamento dinâmico do sistema.

1.3.2 Objetivos Específicos

1. Implementar um código para a coleta de dados da bancada motor-gerador, garantindo uma base sólida para análise e modelagem do sistema.
2. Desenvolver algoritmos para construção de modelos através da técnica `SINDy`, utilizando os dados coletados para representação precisa do comportamento dinâmico do sistema.
3. Implementar um código offline para otimização dos parâmetros dos controladores PI e PID através de AG com a biblioteca `PyGad`.
4. Desenvolver um código online para transmissão dos controladores otimizados pela AG para a bancada motor-gerador, permitindo um controle em tempo real.

1.4 Metodologia

A implementação do sistema de controle é desdobrada em três etapas primordiais, conforme delineado por (SUTTON; BARTO, 2018): a formulação do modelo matemático do sistema, a integração do AG e a conexão com a plataforma Arduino.

Na fase inicial, o código implementado (ver Apêndice **??**) emite um sinal `PRBS` para a bancada motor-gerador e registra o comportamento do sistema dinâmico em tempo real. Os dados coletados são então transferidos para um segundo código (ver Apêndice **??**), responsável por gerar um modelo descritivo do sistema.

Na segunda etapa, após a identificação do modelo, é definida a topologia para o controlador. Após isto, o AG é instaurado utilizando a biblioteca `PyGad` (*Python Genetic Algorithm Library*). Essa etapa exige a definição da população inicial, das funções de avaliação e seleção dos indivíduos, dos operadores de mutação e crossover, e do critério de parada, conforme indicado por (HOLLAND, 1992). Depois do ajuste dos parâmetros o AG é executado e

visando a minimização do custo. O controlador que atingir o melhor desempenho é selecionado para ser implementado na malha de controle do sistema real.

Na terceira e última fase, realiza-se a integração do AG com a plataforma Arduino. Para tanto, é imperativo empregar a biblioteca `Serial` para gerenciar as entradas e saídas da placa Arduino. A saída do controlador é empregada para regular um ou mais atuadores do sistema dinâmico, como um motor, conforme mencionado por (MCROBERTS, 2018).

A implementação integral do sistema de controle possibilita a avaliação do desempenho do AG na otimização do controle do sistema dinâmico. Isso viabiliza a análise da eficácia e rapidez do algoritmo genético na descoberta de soluções ótimas para o problema de controle, conforme ressaltado por (BRUNTON; KUTZ, 2019).

A escolha da plataforma Arduino para as operações de controle é justificada pelo seu baixo custo e facilidade de uso. Além disso, a Arduino oferece uma ampla gama de sensores e atuadores que podem ser empregados no controle de sistemas dinâmicos, como destacado por (MCROBERTS, 2018).

A implementação de um controle ótimo baseado em algoritmo genético na linguagem Python usando a biblioteca `PyGad` e a plataforma Arduino é uma excelente opção para controlar sistemas dinâmicos em tempo real de forma eficiente e de baixo custo (BACK; FOGEL; MICHALEWICZ, 1997). Com a implementação desse sistema, é possível otimizar o desempenho de sistemas dinâmicos em diversas áreas da engenharia, como automação industrial, robótica e controle de processos (MITCHELL, 1998).

1.5 Apresentação do Trabalho

O trabalho foi organizado em cinco capítulos. No Capítulo dois, serão apresentados tópicos sobre os fundamentos da sintonia de controladores PI e PID, destacando as abordagens analíticas, empíricas e adaptativas, assim como os princípios dos AGs, apresentando como ocorre a emulação de um AG computacional. No Capítulo três será apresentada a metodologia utilizada para realizar a identificação do modelo da planta pelo método `SINDy`, assim como da obtenção dos controladores PI e PID através do `PyGad` proposto neste trabalho. No Capítulo quatro serão apresentados os resultados obtidos. No Capítulo cinco será realizada a discussão dos resultados. Por fim, o Capítulo seis será dedicado às considerações gerais a cerca do trabalho realizado e dos resultados obtidos.

CAPÍTULO

2

FUNDAMENTOS

2.1 Fundamentos e Sintonia de Controladores PI e PID

Os controladores Proporcional Integral (PI) e Proporcional Integral Derivativo (PID) são amplamente utilizados em sistemas de controle automático para garantir a estabilidade e o desempenho desejado. Vamos explorar os parâmetros desses controladores:

1. **Proporcional (K_p):** Este é o parâmetro proporcional que determina a magnitude da resposta do sistema proporcional à diferença entre a referência desejada e a saída real. Um valor maior de K_p aumenta a ação proporcional, proporcionando uma resposta mais rápida, mas pode resultar em oscilações se for muito alto.

2. **Integral (K_i):** O parâmetro integral está relacionado com a acumulação ao longo do tempo do erro entre a saída real e a referência desejada. Ele ajuda a eliminar o erro em estado estacionário, melhorando a precisão do sistema. No entanto, um valor excessivamente alto de K_i pode levar a respostas lentas e a oscilações.

3. **Derivativo (K_d):** Este parâmetro considera a taxa de variação do erro. Introduzindo uma ação proporcional à velocidade de mudança do erro, o termo derivativo ajuda a prevenir oscilações excessivas e aprimora a resposta transitória. Um valor muito alto de K_d pode resultar em amplificação de ruído, prejudicando a estabilidade.

4. **PI (K_p, K_i):** O controlador PI é formulado pela combinação de duas ações essenciais: a proporcional, vinculada ao erro instantâneo, e a integral, que se relaciona à soma acumulada dos erros ao longo do tempo. Ao adaptarmos os parâmetros K_p e K_i com base nos dados do sistema motor-gerador, estabelecemos o controle para os três pontos de operação.

   A equação do controlador PI é expressa por:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)\, d\tau \tag{2.1}$$

5. **PID (K_p, K_i, K_d):** Ao combinar os três termos, o controlador PID busca otimizar o desempenho do sistema. O parâmetro K_p domina a resposta inicial, K_i elimina o erro em estado estacionário, e K_d melhora a estabilidade. A sintonia adequada desses parâmetros é crucial e depende das características específicas do sistema controlado.

   A equação geral de um controlador PID é expressa como:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)\, d\tau + K_d \cdot \frac{de(t)}{dt} \tag{2.2}$$

A sintonia eficaz desses parâmetros é uma tarefa importante para alcançar um desempenho ideal do controlador em diferentes condições de operação. Diversas técnicas, como métodos de ajuste manual, sintonia automática e algoritmos avançados, são empregadas para otimizar esses valores em aplicações práticas.

A sintonia eficiente desses controladores é crucial para garantir que o sistema atenda aos requisitos de desempenho desejados. Tanto abordagens analíticas quanto empíricas desempenham um papel vital na obtenção desses sintonizadores, levando a um campo vasto e complexo de pesquisa e aplicação prática. Entre as estratégias de sintonia destacam-se as ditas analíticas, empíricas e a integração destas.

As abordagens analíticas no campo do controle de sistemas dinâmicos baseiam-se em modelos matemáticos do sistema, proporcionando uma sintonia mais precisa e teoricamente fundamentada, como enfatizado por diversos pesquisadores na área (por exemplo, (OGATA, 2010; KLUEVER, 2020)). Essa metodologia é crucial para o design de controladores eficientes e robustos, uma vez que permite uma compreensão profunda do comportamento dinâmico do sistema.

Entre os métodos comuns nas abordagens analíticas, destaca-se o uso de critérios de desempenho, tais como o critério de otimização baseado em Lugar das Raízes (LGR) (FRANKLIN; POWELL; EMAMI-NAEINI, 2010). O LGR, ao analisar a variação de parâmetros, oferece uma representação gráfica das raízes do polinômio característico, proporcionando insights valiosos sobre a estabilidade e resposta dinâmica do sistema.

Outra técnica frequentemente empregada é a utilização de técnicas de alocação de pólos, conforme discutido por (DAVISON; WANG, 1975). Nesse contexto, o controle é projetado de modo a alocar os pólos do sistema controlado em posições específicas no plano complexo. Esta estratégia sistemática influencia diretamente a resposta transitória e a estabilidade do sistema, permitindo uma sintonia precisa e adaptada às características particulares do sistema.

Dessa forma, ao adotar abordagens analíticas, os engenheiros de controle podem contar com métodos robustos e teoricamente embasados, utilizando critérios de desempenho como ferramentas valiosas na busca por soluções eficientes e bem ajustadas para sistemas dinâmicos complexos.

As abordagens empíricas representam uma estratégia valiosa no campo do controle, fundamentadas na experimentação prática e ajustes iterativos. Essa metodologia ganha

destaque, especialmente em sistemas complexos e não-lineares, nos quais a obtenção de modelos matemáticos pode se mostrar desafiadora.

Conforme enfatizado por (ASTROM; HAGGLUND, 2008) em "Advanced PID Control,"a sintonia empírica se desenrola por meio da coleta sistemática de dados experimentais e do ajuste manual dos parâmetros do controlador. Esse processo é repetido iterativamente até que o desempenho desejado seja atingido. O cerne dessa abordagem reside na flexibilidade oferecida para se adaptar a variações não modeladas e incertezas no sistema, uma vez que as nuances do sistema podem ser incorporadas à medida que surgem durante a experimentação. Essa adaptabilidade é especialmente crucial em situações onde a complexidade do sistema dificulta a formulação precisa de modelos matemáticos. Portanto, a sintonia empírica proporciona uma resposta dinâmica e ajustada às condições do mundo real, contribuindo para a eficácia e robustez do controle em sistemas complexos e dinâmicos.

Recentemente, pesquisadores têm explorado a integração de abordagens analíticas e empíricas. O uso de algoritmos de otimização, como o algoritmo genético, tem permitido a sintonia automática de controladores PID com base em dados experimentais, superando algumas limitações das abordagens puramente analíticas ou empíricas (BRUNTON; KUTZ, 2019).

Como destaca (ASTROM; MURRAY, 2010) em "Feedback Systems,"a combinação dessas abordagens pode levar a uma sintonia mais robusta e adaptativa, capaz de lidar com uma ampla gama de condições de operação.

Em suma, a sintonia de controladores PID envolve uma cuidadosa consideração de abordagens analíticas e empíricas. Enquanto os métodos analíticos fornecem uma base teórica sólida, as abordagens empíricas oferecem flexibilidade em ambientes dinâmicos e complexos. A busca contínua por métodos integrados visa aprimorar a eficiência e a adaptabilidade desses controladores em aplicações práticas.

2.2 Sintonia Adaptativa

A Sintonia Adaptativa é um conceito crucial em sistemas dinâmicos, sendo aplicada em uma variedade de campos, desde controle de processos industriais até otimização de algoritmos. Ela se destaca pela capacidade de ajustar automaticamente os parâmetros de um sistema para otimizar seu desempenho diante de mudanças nas condições operacionais.

2.2.1 Abordagens na Sintonia Adaptativa

1. **Identificação de Modelo:** Esta metodologia, fundamentada na teoria de controle, utiliza diversas técnicas para extrair informações sobre o comportamento dinâmico do sistema, possibilitando ajustes precisos de seus parâmetros.

   Uma abordagem comum nesse contexto é a identificação do modelo do sistema em tempo real. Esta técnica procura obter uma representação matemática do sistema com

base nos dados observados durante a operação. Através da análise desses dados, é possível ajustar os parâmetros do modelo, permitindo uma melhor correspondência entre o comportamento simulado e o comportamento real do sistema.

Conforme ressaltado por (GOODWIN, 2001), a identificação de modelo desempenha um papel essencial na obtenção de uma sintonia adaptativa robusta. A capacidade de estimar com precisão a dinâmica do sistema não apenas facilita a compreensão de seu comportamento, mas também possibilita a adaptação eficiente dos parâmetros do controlador em resposta a variações no ambiente ou no próprio sistema. Essa abordagem contribui para a robustez do sistema de controle, permitindo uma resposta eficaz a condições dinâmicas variáveis.

Dessa forma, a identificação de modelo emerge como uma ferramenta valiosa no arsenal do engenheiro de controle, oferecendo a base necessária para o desenvolvimento de estratégias de controle adaptativas e eficazes em ambientes dinâmicos e sujeitos a mudanças.

2. **Controle Preditivo:** As Abordagens de Controle Preditivo representam uma estratégia inovadora ao empregar modelos preditivos para antecipar o comportamento futuro de um sistema dinâmico. Diferentemente de abordagens tradicionais que operam exclusivamente com retroalimentação, o Controle Preditivo incorpora a capacidade de realizar previsões e ajustar proativamente os parâmetros do controlador para otimizar o desempenho do sistema.

   Como destacado por CAMACHO; BORDONS em sua obra referencial de 2004, "o controle preditivo adapta os parâmetros do controlador com base nas previsões, proporcionando respostas rápidas a mudanças no ambiente". Essa abordagem inovadora permite ao controlador antecipar as variações esperadas no sistema, utilizando modelos preditivos para estimar futuros estados e comportamentos. Essas previsões são então utilizadas para otimizar as ações de controle antes que as mudanças efetivamente ocorram.

   A capacidade de adaptação do Controle Preditivo, derivada de sua capacidade preditiva, torna-o particularmente eficaz em situações dinâmicas e sujeitas a alterações. Ao antecipar as condições futuras, o controlador pode ajustar os parâmetros de controle de forma a minimizar desvios em relação aos objetivos desejados, resultando em respostas rápidas e eficientes às mudanças ambientais.

   Em resumo, as Abordagens de Controle Preditivo oferecem uma perspectiva avançada e proativa para o controle de sistemas dinâmicos, capacitando os controladores a se adaptarem continuamente às condições futuras previstas, conforme destacado por Camacho e Bordons. Essa capacidade de prever e reagir antecipadamente às mudanças ambientais confere ao Controle Preditivo um papel crucial na busca por respostas de controle mais rápidas e eficientes.

3. **Algoritmos Genéticos:** Os AGs surgem como uma poderosa ferramenta, frequentemente empregada para a sintonia adaptativa em processos de otimização. De acordo com

(DEB, 2001), "a evolução genética simula o processo de seleção natural para encontrar conjuntos ideais de parâmetros, demonstrando eficácia em ambientes dinâmicos". Essa abordagem inovadora baseia-se no princípio da seleção natural, imitando o processo evolutivo observado na natureza.

Os AGs operam por meio de uma população de soluções candidatas representadas como indivíduos, cada um composto por um conjunto de parâmetros. Esses indivíduos passam por um processo de seleção, recombinação e mutação ao longo de várias gerações, com o objetivo de buscar soluções ótimas para o problema em questão. Essa imitação do processo evolutivo permite que os AGs explorem amplamente o espaço de busca, adaptando-se dinamicamente a mudanças nas condições do ambiente.

No contexto da sintonia adaptativa, os AGs destacam-se por sua capacidade de ajustar os parâmetros do sistema de forma autônoma e contínua. Isso é especialmente vantajoso em ambientes dinâmicos, nos quais as condições podem variar ao longo do tempo. A aplicação de AGs na sintonia adaptativa demonstra eficácia em encontrar conjuntos ideais de parâmetros, maximizando ou minimizando a função objetivo, de acordo com os requisitos do sistema.

A simulação do processo de seleção natural pelos AGs possibilita uma exploração eficiente do espaço de busca, permitindo identificar soluções robustas e adaptáveis a mudanças no ambiente. Assim, os Algoritmos Genéticos surgem como uma abordagem versátil e eficaz para a sintonia adaptativa em otimização, proporcionando uma resposta dinâmica e evolutiva para sistemas que operam em ambientes complexos e mutáveis.

Ao analisar as diversas abordagens exploradas para a Sintonia Adaptativa, fica evidente a riqueza e a complexidade dessas técnicas disponíveis. A essência da Sintonia Adaptativa reside na capacidade de ajustar dinamicamente os parâmetros do sistema em resposta às mudanças nas condições operacionais. Essa adaptabilidade é vital em ambientes dinâmicos, nos quais a resposta inteligente a variações contínuas é essencial para garantir um desempenho otimizado ao longo do tempo.

Como expresso por (FRIEDMAN, 2001), "a sintonia adaptativa representa a resposta inteligente a um ambiente em constante mudança, permitindo a eficácia contínua dos sistemas complexos". Essa afirmação ressalta a natureza dinâmica e a importância fundamental da sintonia adaptativa na manutenção da eficácia operacional de sistemas complexos.

Em última análise, a variedade de abordagens discutidas aqui reflete a evolução contínua do campo da sintonia adaptativa, buscando atender às demandas de sistemas que operam em ambientes desafiadores e sujeitos a mudanças constantes. A inteligência inerente à adaptação desses métodos ressalta a capacidade da engenharia de controle em desenvolver estratégias robustas e flexíveis, capazes de enfrentar os desafios de sistemas dinâmicos e em constante evolução.

2.3 Algoritmo Genético

Os AGs são técnicas de otimização inspiradas na evolução biológica, que buscam soluções eficientes para problemas complexos através da simulação, por emulação, de processos genéticos. Desenvolvidos por HOLLAND na década de 1960, esses algoritmos encontraram aplicação em diversas áreas devido à sua capacidade de explorar espaços de busca extensos e encontrar soluções próximas ao ótimo global. Em que situações esses algoritmos são mais comumente aplicados?

1. **Otimização de Funções Complexas:** Os AGs destacam-se na otimização de funções complexas, onde a relação entre variáveis é não linear e a paisagem de busca é irregular. Segundo (GOLDBERG, 2010), "os AGs são particularmente eficazes em problemas de otimização global, nos quais o espaço de busca é vasto e mal-comportado".

2. **Design de Sistemas Complexos:** Em engenharia e projeto de sistemas complexos, os AGs são aplicados para encontrar soluções eficientes em problemas que envolvem múltiplos parâmetros e restrições. (HOLLAND, 1992) destaca que "os AGs são capazes de encontrar soluções ótimas ou subótimas em espaços de busca complexos e multidimensionais".

3. **Aprendizado de Máquina e Otimização de Redes Neurais:** Na área de aprendizado de máquina, os AGs são empregados na otimização de hiperparâmetros de modelos, contribuindo para a melhoria do desempenho. (DEB, 2001) ressalta que "os AGs têm sido eficazes na busca de conjuntos de parâmetros que otimizam o desempenho de algoritmos de aprendizado de máquina".

4. **Otimização Combinatória:** Problemas combinatórios, como o Problema do Caixeiro Viajante, são notórios pela sua complexidade. Os AGs mostram-se eficazes na busca por soluções aproximadas para esses problemas, sendo capazes de lidar com um grande número de combinações possíveis (MITCHELL, 1998).

5. **Otimização de Portfólios Financeiros:** Na área financeira, os AGs são utilizados na otimização de portfólios de investimentos. (EIBEN; SMITH, 2015) destacam que "os AGs são particularmente adequados para lidar com a natureza estocástica e dinâmica dos mercados financeiros, buscando a alocação de ativos que maximize o retorno e minimize o risco".

Em suma, os AGs emergem como ferramentas poderosas em situações onde métodos convencionais de otimização enfrentam desafios. Sua capacidade de explorar espaços de busca complexos e encontrar soluções robustas faz deles uma escolha valiosa em uma variedade de domínios, contribuindo para a resolução de problemas complexos e aprimoramento de sistemas e algoritmos.

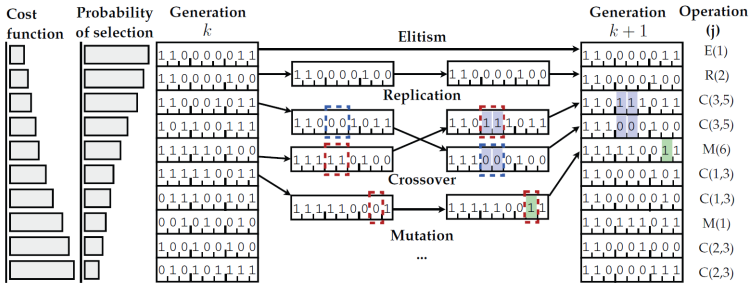
Diante das inúmeras aplicações e benefícios apresentados pelos algoritmos genéticos, é evidente a sua importância como ferramenta de otimização em diversas áreas, tanto na

indústria quanto na pesquisa científica. Este trabalho é motivado pela aplicação de AGs na sintonia de controladores de sistemas dinâmicos, uma abordagem que tem ganhado destaque em diversas aplicações recentes (HOLLAND, 1992; DEB, 2001).

2.3.1 Princípios dos Algoritmos Genéticos

A técnica de AG parte de um conjunto inicial de possíveis soluções do problema de otimização e busca a partir desta população criar novas soluções. Este processo de criação se baseia em eventos verificados na natureza, como por exemplo: o elitismo, a replicação, o cruzamento e a mutação. Esses componentes estão devidamente representados na figura 1.

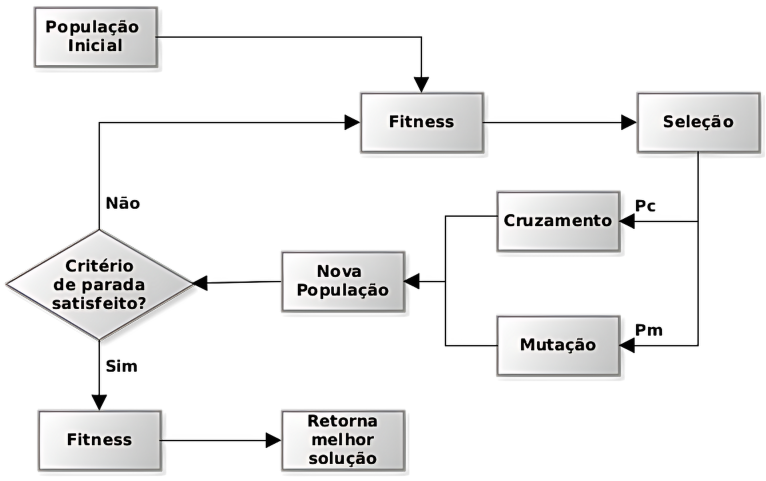
Figura 1 – Ilustração da evolução de um algoritmo genético.



Fonte: (BRUNTON; KUTZ, 2019)

Este processo é replicado em algumas etapas, cada uma da qual busca gerar uma nova população, das quais se espera que novos indivíduos, que ofereçam melhores soluções para o problema de otimização, sejam gerados. A estrutura de um Algoritmo Genético pode ser representada através do fluxograma da figura 2.

Figura 2 – Fluxograma Algoritmo Genético.



Fonte: (IZIDORO; MELO-MINARDI; PAPPA, 2014)

2.3.1.1 Elitismo

O elitismo em AGs é uma estratégia fundamental que visa preservar os melhores indivíduos de uma população ao longo das gerações. Essa abordagem consiste em selecionar os melhores cromossomos, ou seja, aqueles que têm as características mais desejáveis ou que atendem melhor ao critério de otimização, e garantir que eles sejam preservados na próxima geração, sem sofrer alterações.

A implementação do elitismo oferece uma série de vantagens em problemas de otimização. Aqui estão algumas ideias desenvolvidas com base no conceito de elitismo em Algoritmos Genéticos:

1. **Preservação de Boas Soluções:** O elitismo assegura que as melhores soluções encontradas até o momento sejam mantidas na população, evitando sua perda devido a operadores genéticos como crossover e mutação. Isso é crucial para evitar a perda prematura de soluções promissoras.
2. **Exploração e Convergência:** Ao preservar os melhores indivíduos, o elitismo permite uma exploração mais eficiente do espaço de busca. Essa exploração eficaz contribui para a convergência do algoritmo em direção a soluções ótimas, acelerando o processo de otimização.
3. **Estabilidade do Processo Evolutivo:** O elitismo confere estabilidade ao processo evolutivo, reduzindo a probabilidade de perda de diversidade genética. Ao manter os melhores indivíduos, a população mantém uma base sólida de soluções de alta qualidade.
4. **Adaptação a Ambientes Dinâmicos:** Em ambientes dinâmicos, onde as condições podem mudar ao longo do tempo, o elitismo pode ser crucial. Ao preservar as melhores soluções, o algoritmo se adapta mais rapidamente a mudanças, fornecendo uma resposta mais eficaz a novas condições do problema.
5. **Controle sobre o Nível de Exploração e Exploração:** O elitismo permite um controle mais preciso sobre o equilíbrio entre exploração e exploração. Manter os melhores indivíduos favorece a exploração, enquanto os operadores genéticos promovem a exploração, contribuindo para uma estratégia de busca equilibrada.

O elitismo em Algoritmos Genéticos é uma estratégia fundamental que fortalece o desempenho do algoritmo de otimização, assegurando a preservação das melhores soluções ao longo das gerações. Essa abordagem desempenha um papel essencial em diferentes contextos e contribui para a eficácia e estabilidade do processo evolutivo.

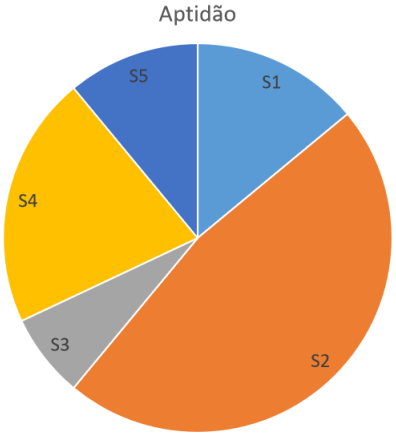
2.3.1.2 Replicação

A replicação em AGs refere-se ao processo de seleção de indivíduos com base em sua aptidão para reprodução. Esse componente, muitas vezes associado à roleta viciada ((MITCHELL, 1998)),

favorece a escolha de soluções melhores, aumentando a probabilidade de suas características serem transmitidas para as gerações subsequentes. A replicação desempenha um papel fundamental na exploração do espaço de busca, permitindo que soluções mais promissoras tenham uma maior contribuição para a próxima geração. A figura 3 representa como a replicação funciona nos AGs.

Figura 3 – Indivíduos de uma população e a sua correspondente roleta de seleção.

	Indivíduo	Aptidão	Aptidão
	S_i	$f(S_i)$	Relativa
S_1	10110	2.23	0.14
S_2	11000	7.27	0.47
S_3	11110	1.05	0.07
S_4	01001	3.35	0.21
S_5	00110	1.69	0.11



Adaptado de: (ANDRÉ, 2023)

O AG simula a replicação de Algoritmos Genéticos, explorando diferentes regiões do espaço de busca. Os resultados das diferentes réplicas são armazenados e, ao final do processo, as melhores soluções obtidas entre todas as réplicas são apresentadas. A replicação de AGs oferece vantagens em termos de diversidade e paralelismo, permitindo uma busca mais abrangente e eficiente por soluções ótimas ou próximas ao ótimo em problemas complexos.

2.3.1.3 Cruzamento (ou Crossover)

O cruzamento, no contexto de AGs, representa o processo essencial pelo qual dois ou mais indivíduos trocam partes de seu material genético, dando origem a descendentes. Este componente fundamental desempenha um papel crucial na combinação de características positivas provenientes de diferentes soluções, resultando em uma diversificação genética dentro da população. A eficácia desse processo é vital para o sucesso do AG, uma vez que influencia diretamente na busca por soluções ótimas.

A escolha apropriada da técnica de cruzamento é de grande importância, pois uma abordagem inadequada pode levar à convergência prematura do algoritmo, onde a população se estabiliza em soluções subótimas antes de explorar completamente o espaço de busca. Por outro lado, uma escolha equivocada também pode resultar na perda de informações cruciais presentes em soluções promissoras.

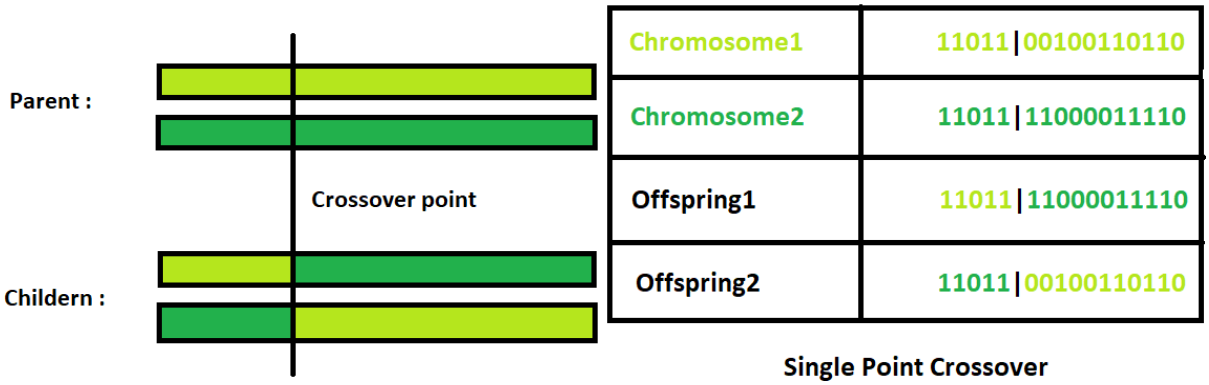
A eficiente utilização do cruzamento é um tópico discutido de forma abrangente em (GOLDBERG, 2010), um trabalho seminal que aborda princípios fundamentais em algoritmos genéticos. Nesse contexto, Goldberg explora estratégias e considerações para otimizar o

processo de cruzamento, destacando a importância de ajustar adequadamente os parâmetros relacionados a essa operação.

Portanto, o entendimento e a implementação criteriosa do cruzamento são aspectos críticos para o desempenho bem-sucedido de algoritmos genéticos. A busca por uma combinação eficiente de características genéticas, guiada pela escolha apropriada de técnicas de cruzamento, contribui significativamente para a capacidade do algoritmo em explorar e convergir para soluções de alta qualidade em espaços de busca complexos e dinâmicos.

1. Cruzamento de Um Ponto: O cruzamento de um ponto é uma técnica amplamente utilizada em AGs, desempenhando um papel crucial na geração de descendentes por meio da troca de material genético entre dois pais em um único ponto de corte. Essa abordagem é caracterizada pela simplicidade e eficiência, sendo frequentemente empregada devido à sua capacidade de promover uma diversificação genética na população.

Figura 4 – Cruzamento de um ponto.



Fonte: (GEEKSFORGEEKS, 2023)

O processo de cruzamento de um ponto é executado da seguinte maneira: um ponto de corte é escolhido aleatoriamente ao longo dos cromossomos dos pais. Os genes localizados antes desse ponto de corte em um dos pais são herdados pelos descendentes, enquanto os genes após o ponto de corte são herdados pelo outro pai. Essa troca de material genético cria descendentes que carregam uma combinação única das características genéticas dos pais.

A aleatoriedade na escolha do ponto de corte é um elemento-chave, introduzindo variabilidade nas descendências geradas e contribuindo para a exploração eficaz do espaço de busca. Essa técnica é particularmente útil em situações em que as características desejáveis podem estar localizadas em diferentes partes dos cromossomos dos pais.

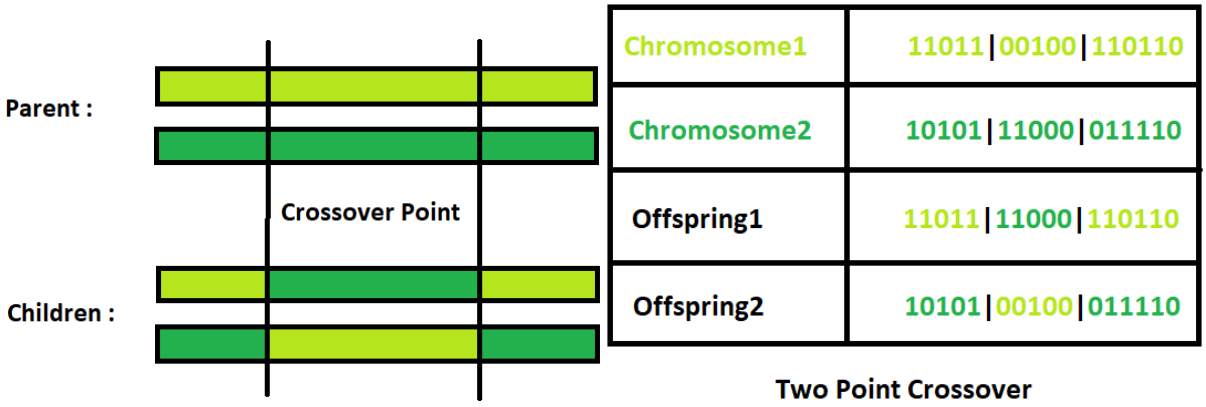
A discussão sobre a eficácia e a aplicação do cruzamento de um ponto pode ser encontrada em (MITCHELL, 1998), uma obra que aborda princípios fundamentais em algoritmos genéticos. Esta abordagem é relativamente simples de implementar e eficiente em termos computacionais. No entanto, o cruzamento de um ponto pode ser limitado em sua capa-

cidade de explorar eficientemente soluções no espaço de busca, especialmente quando as características relevantes para o problema estão distribuídas em várias regiões.

2. Cruzamento de Dois Pontos: O cruzamento de dois pontos é uma extensão do cruzamento de um ponto, onde dois pontos de corte são escolhidos aleatoriamente. Os genes entre esses dois pontos de corte são trocados entre os pais, resultando em uma seção intermediária herdada de um dos pais e seções externas herdadas do outro pai.

   Essa técnica aumenta a diversidade genética introduzida pelos cruzamentos, permitindo uma combinação mais complexa de características de ambos os pais. O cruzamento de dois pontos pode ser particularmente útil em problemas em que certas características importantes podem estar localizadas em regiões específicas do genoma.

Figura 5 – Cruzamento de dois pontos.



Fonte: (GEEKSFORGEEKS, 2023)

Segundo (MITCHELL, 1998), a escolha entre o cruzamento de um ponto e o cruzamento de dois pontos depende da natureza do problema a ser resolvido. O cruzamento de um ponto é mais simples e eficiente computacionalmente, mas pode ser limitado em sua capacidade de explorar de maneira abrangente o espaço de busca. O cruzamento de dois pontos é mais complexo, mas pode oferecer uma exploração mais diversificada e refinada do espaço de busca, especialmente em problemas com características distribuídas.

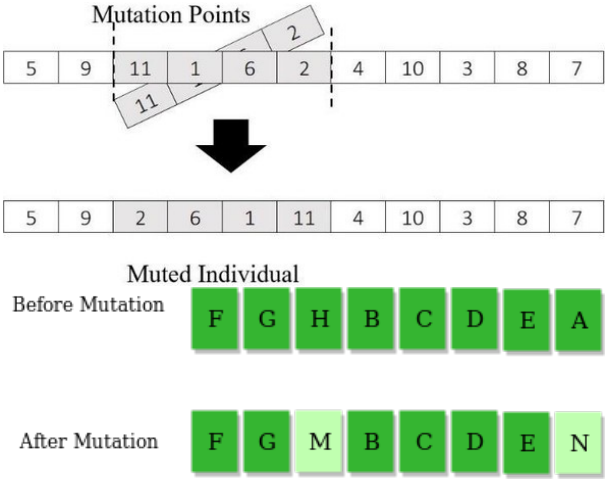
Em muitos casos, a escolha entre essas técnicas é feita empiricamente, considerando o desempenho em problemas específicos. Em problemas nos quais a interação entre genes distantes é crucial, o cruzamento de dois pontos pode fornecer uma vantagem. Entretanto, em problemas mais simples, o cruzamento de um ponto pode ser suficiente e mais eficiente em termos computacionais.

Tanto o cruzamento de um ponto quanto o cruzamento de dois pontos são estratégias eficazes em Algoritmos Genéticos, cada uma com suas vantagens e limitações. A escolha entre essas técnicas deve ser orientada pela natureza do problema específico que está sendo abordado, visando otimizar a exploração do espaço de busca e a convergência para soluções de alta qualidade. Experimentação e ajustes empíricos são frequentemente necessários para determinar a técnica mais adequada para um dado problema.

2.3.1.4 Mutação

A mutação desempenha um papel crucial em AGs, envolvendo a introdução aleatória de mudanças no material genético de um indivíduo. Essa operação é projetada para manter a diversidade genética na população, explorar novas regiões do espaço de busca e evitar a estagnação em ótimos locais. Ao introduzir variações aleatórias nos genes, a mutação permite a introdução de novas características e a adaptação contínua da população ao longo do tempo.

Figura 6 – Mutação em indivíduos do AG.



Fonte: (GEEKSFORGEEKS, 2023)

A taxa de mutação é uma consideração crítica nesse processo, determinando a probabilidade de um gene ser alterado em um determinado indivíduo. Uma taxa de mutação adequada é essencial para equilibrar a exploração do espaço de busca e a exploração de soluções já encontradas. Uma taxa muito alta pode resultar em uma busca caótica, onde a população pode perder boas soluções devido a mutações excessivas. Por outro lado, uma taxa muito baixa pode levar à convergência prematura, onde a população fica estagnada em uma região do espaço de busca antes de explorar completamente suas possibilidades.

A obra de (HOLLAND, 1992) discute a importância da mutação na adaptação contínua. Holland enfatiza que a mutação é crucial para introduzir inovações na população, permitindo que o algoritmo genético escape de ótimos locais subótimos e continue explorando soluções potenciais. A combinação adequada de cruzamento e mutação é vital para a eficácia global do algoritmo, garantindo um equilíbrio entre exploração e exploração em um ambiente dinâmico.

Dessa forma, a mutação, quando utilizada com equilíbrio, contribui para a capacidade adaptativa dos AGs, promovendo uma busca robusta e eficiente por soluções ótimas em espaços de busca complexos e desafiadores.

Elitismo, replicação, cruzamento e mutação emergem como pilares fundamentais nos AGs, desempenhando papéis vitais na exploração eficiente e eficaz do espaço de busca. Esses componentes, interagindo harmoniosamente, contribuem para a robustez e adaptabilidade

desses algoritmos, permitindo a resolução de uma ampla gama de problemas em diversas aplicações.

O elitismo, ao preservar os indivíduos mais aptos na população, atua como um mecanismo de preservação das características desejáveis ao longo das gerações. A replicação, por sua vez, introduz diversidade e explorabilidade, possibilitando a busca em várias regiões do espaço de busca simultaneamente. O cruzamento combina características genéticas positivas de diferentes soluções, promovendo a criação de descendentes mais adaptáveis. Enquanto isso, a mutação, ao introduzir aleatoriedade, impede a estagnação da população e incentiva a descoberta de soluções inovadoras.

A compreensão profunda desses elementos e de suas interações é crítica para o projeto e implementação bem-sucedidos de AGs em uma variedade de aplicações. A escolha cuidadosa de parâmetros, como taxas de cruzamento e mutação, bem como estratégias de elitismo e replicação, influencia diretamente no desempenho e na convergência do algoritmo. A capacidade de ajustar esses componentes de acordo com as características do problema específico é essencial para otimizar a eficácia dos AGs.

2.3.2 Estrutura Básica dos AGs

A estrutura fundamental de um Algoritmo Genético (AG) é delineada por cinco componentes-chave, cada um desempenhando um papel essencial no processo evolutivo. Esses elementos colaboram de maneira sinérgica para explorar e otimizar o espaço de busca de maneira eficiente. Os cinco principais componentes são:

1. **População Inicial:** A população inicial consiste no conjunto inicial de indivíduos, representados por soluções candidatas ao problema em questão. Esses indivíduos são gerados aleatoriamente ou com base em conhecimento prévio do domínio do problema.

2. **Função de Custo (Fitness):** A função de avaliação, também conhecida como função de fitness, quantifica a adequação ou qualidade de cada indivíduo na população. Ela atribui um valor numérico que reflete o quão bem uma solução atende aos requisitos do problema. A função de avaliação é crucial para orientar o processo evolutivo, privilegiando indivíduos mais aptos. "A função custo é crucial, pois é ela que guia a busca ao longo do espaço de soluções possíveis." (GOLDBERG, 2010)

3. **Seleção:** O processo de seleção determina quais indivíduos serão escolhidos para reprodução e formação da próxima geração. Estratégias de seleção, como a roleta viciada ou torneios, são empregadas para dar preferência aos indivíduos mais aptos, aumentando suas chances de serem pais na próxima geração.

4. **Operadores Genéticos (Crossover e Mutação):** O crossover (cruzamento) e a mutação são os operadores genéticos responsáveis pela criação de novos indivíduos. O crossover envolve a troca de material genético entre dois pais para gerar descendentes. A mutação introduz pequenas alterações aleatórias nos genes dos indivíduos, adicionando

diversidade à população. Esses operadores são fundamentais para explorar e ampliar o espaço de busca. "A combinação eficiente de crossover e mutação desempenha um papel crucial na convergência e diversidade da população." (MITCHELL, 1998)

5. **Critério de Parada:** O critério de parada define as condições que encerram a execução do algoritmo. Pode ser baseado em um número fixo de gerações, atingir um valor mínimo de fitness ou outras condições específicas do problema. O critério de parada garante que o algoritmo alcance um ponto de convergência ou atenda aos objetivos predefinidos.

A integração eficaz desses cinco componentes é crucial para o sucesso e desempenho de um AG. A estrutura modular e adaptativa permite a aplicação desses algoritmos a uma ampla variedade de problemas, tornando-os uma ferramenta poderosa em otimização e busca heurística.

2.4 Desempenho das AGs em sintonizadores

Vários estudos têm sido realizados para avaliar o desempenho de AGs para a sintonia de controladores. Em geral, os resultados desses estudos indicam que AGs podem ser uma abordagem eficaz para a sintonia de controladores. Em particular, AGs têm mostrado ser capazes de encontrar parâmetros de controle que proporcionam um desempenho superior ao de abordagens tradicionais, como o método de Ziegler-Nichols.

Em um estudo (GANI; ISLAM; ULLAH, 2019) propõem um método baseado em AG para ajustar os parâmetros do controlador PID visando controlar a temperatura de uma fornalha elétrica. O método proposto de ajuste de PID baseado em AG foi avaliado por meio de simulações e comparado com o método de Ziegler-Nichols. Os resultados demonstraram que o controlador PID ajustado pelo AG alcançou desempenho superior na minimização de `IAE`, overshoot, tempo de subida e tempo de estabilização. O controlador ajustado pelo AG também apresentou melhor robustez contra perturbações no sistema. Em particular, os parâmetros de controle obtidos por AGs resultaram em uma resposta de controle mais rápida e menos oscilatória.

Em outro estudo, (JAYACHITRA; VINODHA, 2015) compararam o desempenho de AGs com o método de Ziegler-Nichols em um sistema de controle PID para o controle de reatores de tanque agitado contínuo. Os resultados do estudo mostraram que o método proposto é capaz de encontrar um conjunto de parâmetros de controlador PID que garantem um desempenho de controle superiores do que os parâmetros de controle obtidos pelo método de Ziegler-Nichols. Em particular, os parâmetros de controle obtidos por AGs resultaram em um erro menor pois é capaz de encontrar um conjunto de parâmetros de controlador PID em um tempo razoável e para diferentes condições operacionais do reator.

Entretanto, é importante notar que o desempenho das AGs na sintonização de sistemas pode ser sensível à escolha adequada de operadores genéticos e parâmetros de controle. Conforme salientado por (MITCHELL, 1998), o sucesso das AGs está intrinsecamente ligado

à seleção criteriosa desses elementos, de modo a equilibrar a exploração e a explotação do espaço de busca.

Em resumo, as pesquisas existentes sugerem que os Algoritmos Genéticos apresentam um desempenho promissor na sintonização de sistemas, especialmente em contextos onde a complexidade e a não linearidade das relações entre parâmetros representam desafios. No entanto, é fundamental abordar cuidadosamente a configuração dos AGs para garantir resultados ótimos e evitar convergência prematura. A contínua pesquisa nesse campo é crucial para aprimorar ainda mais a compreensão e o aproveitamento dessas poderosas ferramentas na otimização de sistemas complexos.

2.5 Implementação com `PyGad`

A implementação da sintonia dos controladores PI e PID foi realizada por meio da biblioteca `PyGad`. Essa biblioteca oferece suporte para a aplicação de algoritmos genéticos em problemas de otimização, permitindo a busca eficiente de valores ótimos para os ganhos dos controladores. No contexto específico deste trabalho, o algoritmo genético foi configurado para otimizar os parâmetros dos controladores, considerando a função de aptidão previamente definida.

Ao empregar o AG, foram ajustados diversos parâmetros, incluindo a população inicial, o número de gerações, taxas de mutação, e outros fatores relevantes. Essas configurações foram cuidadosamente ajustadas para garantir uma exploração eficiente do espaço de solução, visando encontrar os valores ideais dos ganhos dos controladores.

Desta forma há a configuração e execução do AG utilizando a biblioteca `PyGad` para otimizar os parâmetros de um controlador. A seguir, apresenta-se uma descrição detalhada dos principais parâmetros e funcionalidades do código:

1. **`num_generations`**: Número total de gerações que o algoritmo genético irá evoluir.
2. **`num_parents_mating`**: Número de pais selecionados em cada geração para produzir descendentes.
3. **`fitness_func`**: Função custo (*fitness*) que avalia o desempenho do controlador para uma dada configuração de parâmetros.
4. **`sol_per_pop`**: Número de soluções (indivíduos) em cada população.
5. **`num_genes`**: Número de genes (variáveis) na codificação genética de cada solução.
6. **`init_range_low` e `init_range_high`**: Intervalo inicial para a geração de valores aleatórios dos genes.
7. **`parent_selection_type`**: Método de seleção de pais, neste caso, **`sss`** (*Steady-State Selection*).
8. **`keep_parents`**: Número de pais que são mantidos na próxima geração.

9. `crossover_type`: Tipo de *crossover* utilizado para recombinação genética, aqui **two_points** (dois pontos).
10. `mutation_type`: Tipo de mutação, neste caso, **random** (mutação aleatória).
11. `mutation_percent_genes`: Percentual de genes que sofrerão mutação em cada geração.
12. `save_best_solutions` **e** `save_solutions`: Salvamento das melhores soluções e de todas as soluções para análise posterior.
13. `mutation_num_genes`: Número de genes afetados pela mutação.
14. `allow_duplicate_genes`: Permite ou não genes duplicados em uma solução.
15. `stop_criteria`: Critério de parada, aqui "saturate_50"indica que o AG deve parar se não houver melhoria significativa nas últimas 50 gerações.
16. `on_generation`: Função chamada a cada geração, neste caso, "salvar_populacoes"é utilizada para salvar a população a cada geração.
17. `ga_instance.run()`: Execução do algoritmo genético.
18. `ga_instance.pop_fitness=True`: Habilita o cálculo da aptidão (*fitness*) para a população.

Essa abordagem baseada em algoritmos genéticos oferece uma maneira sistemática e automatizada de sintonizar os controladores, proporcionando uma solução robusta e eficiente para otimizar o desempenho do sistema de controle. A utilização da biblioteca `PyGad` simplifica o processo de implementação do algoritmo genético, tornando-o acessível e eficaz para a sintonia de controladores em sistemas dinâmicos.

2.6 Configuração dos Algoritmos Genéticos

No contexto do AG empregado, houve uma adaptação cuidadosa de certos parâmetros para atender às demandas particulares associadas à otimização dos controladores PI e PID. Na tabela abaixo é apresentada uma descrição mais pormenorizada desses parâmetros:

Esses parâmetros foram definidos por meio de experimentação preliminar, buscando uma combinação que equilibre eficientemente a exploração do espaço de soluções e a convergência para ótimos locais. A seleção cuidadosa desses parâmetros desempenha um papel fundamental assegurando que o AG alcance soluções de alta qualidade para o problema específico de sintonia de controladores PI e PID.

Tabela 1 – Parâmetros do Algoritmo Genético.

Parâmetro	**Valor**
Número de Gerações	1000
Número de Pais para Cruzamento	6
Função de Aptidão	AjusteControlador
População por Geração	25
Número de Genes (Parâmetros)	3
Intervalo Inicial (Baixo)	0.1
Intervalo Inicial (Alto)	5
Tipo de Seleção de Pais	Steady-State Selection (sss)
Número de Pais Mantidos	1
Tipo de Crossover	Two Points
Tipo de Mutação	Random
Taxa de Mutação (por Gene)	0.01
Salvar Melhores Soluções	Verdadeiro
Salvar Todas as Soluções	Verdadeiro
Número de Genes Afetados pela Mutação	1
Permitir Genes Duplicados	Falso
Critério de Parada	Saturar após 50 gerações sem melhoria

Fonte: O Autor

2.7 Ambiente Computacional `Python`

O Python é um ecossistema dinâmico que oferece uma plataforma robusta para o desenvolvimento de software. Com uma sintaxe limpa e legível, Python é amplamente reconhecido por sua versatilidade e facilidade de aprendizado (GAD, 2021).

A simplicidade e clareza sintática do Python facilitam a implementação de algoritmos genéticos, permitindo aos desenvolvedores expressar conceitos complexos de maneira concisa. A vasta gama de bibliotecas, como NumPy, que oferece suporte para operações numéricas eficientes, e a biblioteca PyGAD, específica para algoritmos genéticos, simplificam a implementação e aceleram o processo de desenvolvimento.

Além disso, a natureza de código aberto do Python fomenta uma comunidade ativa e colaborativa. Isso resulta em uma abundância de recursos, documentação rica e suporte constante. O Python também é intercompatível com diversas plataformas, proporcionando flexibilidade no desenvolvimento e garantindo a portabilidade dos algoritmos genéticos.

A visualização de resultados e análise de dados são facilitadas por bibliotecas como Matplotlib e Seaborn, proporcionando uma compreensão mais profunda do comportamento e desempenho dos algoritmos genéticos. Além disso, a integração com Jupyter Notebooks permite a criação de documentos interativos, tornando a exploração e apresentação dos resultados mais envolventes.

2.8 Descrição das Bibliotecas Python

As bibliotecas Python são conjuntos de módulos e funções pré-implementados, criados para resolver problemas específicos ou realizar tarefas comuns de maneira eficiente. Essas ferramentas pré-existentes abstraem complexidades, permitindo que os desenvolvedores se concentrem na lógica específica de seus projetos, em vez de reinventar a roda a cada novo desafio.

A diversidade das bibliotecas é notável, abrangendo áreas como processamento de dados, aprendizado de máquina, visualização gráfica, manipulação de arquivos, comunicação de rede, entre outras. Ao incorporar essas bibliotecas em seus códigos, os desenvolvedores podem aproveitar anos de experiência e otimização coletiva, economizando tempo e recursos.

Abaixo estão descritas as principais bibliotecas utilizadas no desenvolvimento deste trabalho.

- **Classe:** Biblioteca personalizada chamada "classe", cujo propósito e funcionalidades específicas são o armazenamento dos melhores indivíduos de todas as gerações e a obtenção da função de transferência discreta através do método Tustin.
- **Control:** Biblioteca para análise e design de sistemas de controle em Python. Oferece ferramentas para modelagem, simulação e análise de sistemas dinâmicos.
- **Matplotlib.pyplot:** Biblioteca para a criação de gráficos e visualizações. `pyplot` é uma interface do Matplotlib que fornece funções similares às do MATLAB para facilitar a geração de gráficos.
- **Numpy:** Biblioteca fundamental para computação numérica em Python. Fornece suporte para arrays e matrizes, bem como funções matemáticas para operações eficientes.
- **Pandas:** Biblioteca que oferece estruturas de dados e ferramentas de análise de dados. É particularmente útil para manipulação e análise de conjuntos de dados tabulares.
- **Pikle:** Módulo para serialização e desserialização de objetos Python. Permite salvar e carregar objetos em formato binário.
- **PySindy:** Biblioteca para a identificação de sistemas dinâmicos esparsos (SINDy). É utilizada para modelagem e análise de sistemas dinâmicos a partir de dados observados.
- **PyGad:** Biblioteca para algoritmos genéticos em Python. Utilizada para otimização e resolução de problemas complexos.
- **SciPy.signal:** Parte do SciPy, esta biblioteca fornece funções relacionadas ao processamento de sinais, incluindo filtragem, transformadas e geração de sinais.
- **Seaborn:** Biblioteca para a melhoria da estética dos gráficos gerados pelo Matplotlib. Oferece estilos visuais adicionais e funções de plotagem de alto nível.
- **Serial:** Biblioteca utilizada para comunicação serial em Python, permitindo a comunicação com dispositivos externos via portas seriais.

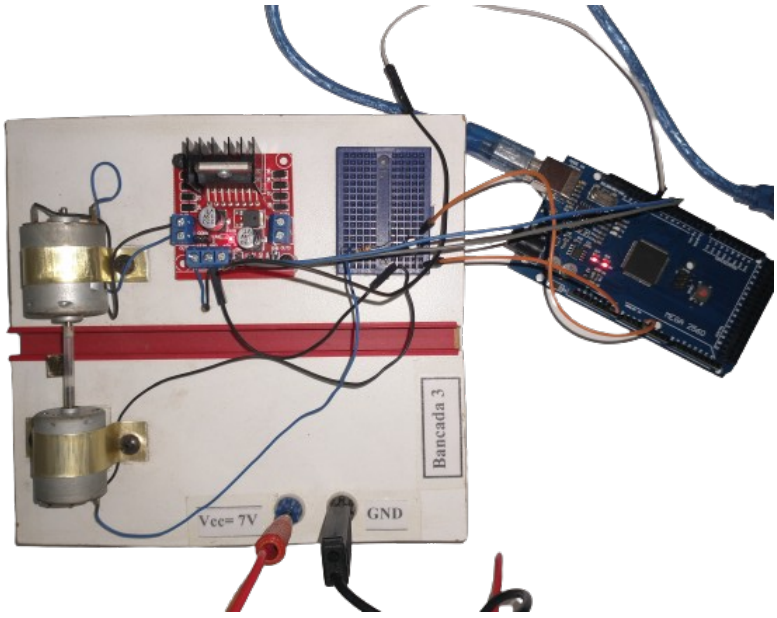
- **Sys:** Módulo que fornece acesso a algumas variáveis utilizadas ou mantidas pelo interpretador Python, bem como a funções que interagem fortemente com o interpretador.
- **Time:** Módulo para manipulação de tempo. Fornece funções relacionadas a medição de tempo e controle de temporização em programas Python.

Estas bibliotecas abrangem uma variedade de funcionalidades, desde visualização de dados até controle de sistemas dinâmicos, processamento de sinais e manipulação de arquivos. A escolha das bibliotecas depende das necessidades específicas do projeto em que estão sendo aplicadas.

2.9 Bancada Motor-Gerador

A bancada, conforme apresentado na Figura 7, é constituída por elementos essenciais para testes e avaliações de desempenho de motores, oferecendo uma plataforma funcional e versátil. Composta por uma mesa branca, uma placa Arduino Mega 2560 e um drive L298N, a configuração é identificada como "Bancada 3". Dois motores, conectados como motor-gerador (taco-gerador), compõem o conjunto, com a placa Arduino assumindo o controle do motor por meio do drive.

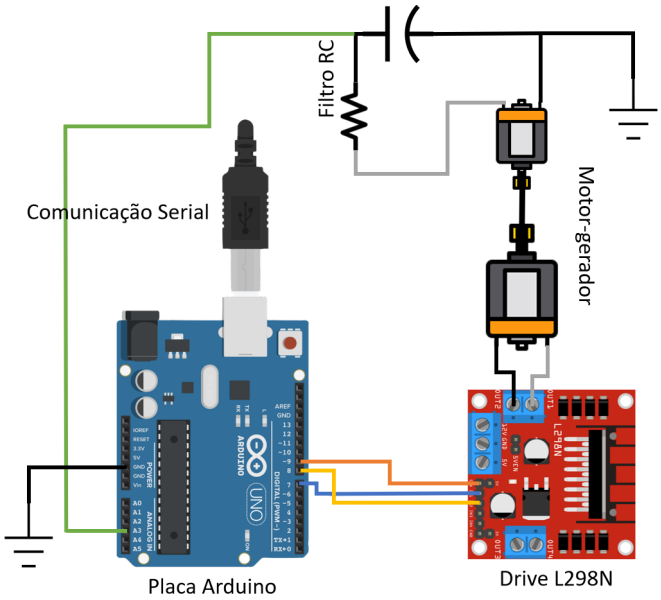
Figura 7 – Bancada Motor-Gerador 7V.



Fonte: O Autor

O esquema elétrico da bancada, detalhadamente representado na Figura 8, oferece uma visão abrangente da interconexão entre os componentes fundamentais. Nesse esquema, a relação entre a placa Arduino, o drive L298N e os motores torna-se evidente, destacando a rota dos fios e as conexões vitais para o funcionamento integrado do sistema.

Figura 8 – Esquema Elétrico da Bancada.



Fonte: O Autor

O drive L298N desempenha um papel crucial no esquema, convertendo a tensão proveniente da fonte de alimentação de `7V` em uma voltagem apropriada para o motor. Destaca-se que a bancada é especialmente projetada para motores de corrente contínua (DC) de ímã permanente. Além disso, o motor-gerador, responsável por converter energia mecânica em energia elétrica, oferece uma abordagem prática para a análise de motores em diferentes condições operacionais.

Para garantir medições precisas, a tensão gerada nos terminais do gerador é direcionada para um filtro `RC`, cujo propósito é minimizar o ruído na leitura, assegurando resultados confiáveis e livres de interferências. Essa atenção aos detalhes na configuração da bancada contribui para a obtenção de dados de alta qualidade durante os experimentos.

CAPÍTULO

3

PROPOSTA METODOLÓGICA

A pesquisa adotada neste trabalho é classificada como exploratória e descritiva, uma vez que visa investigar e descrever a eficiência de estratégias de controle para uma bancada motor-gerador. Essa abordagem possibilita uma compreensão mais profunda do comportamento dinâmico do sistema. O método utilizado é dedutivo, seguindo a lógica de desenvolver estratégias de controle eficientes com base em teorias existentes sobre controladores PI, PID, AGs e Modelos `SINDy`. A abordagem não apenas explora as vantagens dos controladores clássicos, mas também incorpora técnicas avançadas de otimização e modelagem, alinhadas com as demandas de sistemas dinâmicos complexos. A expectativa é que essa abordagem promova um desempenho otimizado, adaptando-se de forma inteligente a diferentes condições e variações do sistema em questão.

3.1 Metodologia

3.1.1 Implementação do Código de Comunicação com a Placa Arduino

A implementação do código de comunicação com a placa Arduino é necessária para viabilizar a troca eficiente de dados entre o computador e o microcontrolador. O processo geralmente envolve a criação de um script em uma linguagem de programação, como Python, que estabelece uma comunicação serial bidirecional com a placa Arduino.

O código facilita a transmissão de dados, permitindo que o computador envie comandos para a placa Arduino e receba informações em tempo real. Essa comunicação bidirecional é essencial para o controle dinâmico de dispositivos conectados à placa, como motores, sensores, ou qualquer outro componente que possa interagir com o Arduino. Aqui está uma visão geral do processo:

1. **Seleção da Linguagem de Programação:** A linguagem de programação Python será

utilizada devido à sua simplicidade e suporte a bibliotecas úteis.

2. **Configuração da Comunicação Serial:** A comunicação serial em Python é definida através da biblioteca `serial`.

3. **Estabelecimento da Conexão:** A porta serial será aberta afim de estabelecer a conexão entre o computador e a placa Arduino. Um tempo de espera é necessário para garantir que a comunicação seja iniciada corretamente.

4. **Envio de Comandos:** A função utilizada é apropriada para enviar comandos ou dados do computador para a placa Arduino. Os dados representam instruções para controle de dispositivos ou qualquer informação necessária.

5. **Recepção de Dados:** A leitura dos dados enviados pela placa Arduino envolve a utilização de funções como `read` ou `readline`.

6. **Encerramento da Conexão:** Ao finalizar a comunicação, a porta serial é fechada para liberar recursos e encerrar a conexão.

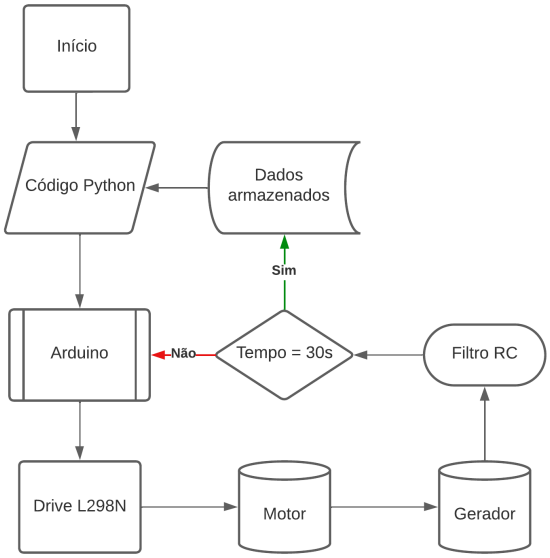
7. **Tratamento de Exceções:** A implementação de tratamento de exceções são realizadas para lidar com possíveis problemas de comunicação, como perda de conexão ou timeouts.

A comunicação serial é uma base fundamental para muitos projetos envolvendo Arduino, proporcionando uma interface eficiente entre o hardware e o software. Certifique-se de adaptar o código conforme necessário para atender aos requisitos específicos do seu projeto.

3.1.2 Coleta de Dados da Bancada Motor-Gerador

A implementação de um código para a coleta de dados da bancada motor-gerador é a etapa primordial neste trabalho. Este código será projetado para realizar a aquisição de dados em diferentes condições de operação da bancada, abrangendo uma variedade de cenários e estados do sistema. A forma como a coleta de dados funcionará está representado no fluxograma abaixo:

Figura 9 – Fluxograma da coleta de dados da bancada motor-gerador



Fonte: O Autor

A coleta de dados desempenha um papel fundamental na construção e análise dos modelos SINDy. Durante a operação da bancada motor-gerador, o código irá registrar informações relevantes, como sinais de entrada e saída, características operacionais e respostas dinâmicas do sistema. É essencial abranger uma ampla gama de condições operacionais para capturar a diversidade de comportamentos que o sistema pode apresentar.

A versatilidade do código de coleta de dados permitirá a adaptação a diferentes protocolos experimentais ou testes específicos da bancada. Isso inclui a variação de parâmetros, a introdução de perturbações controladas e a observação de respostas em diferentes pontos de operação. A coleta abrangente de dados proporcionará uma base sólida para a construção dos modelos SINDy.

Em resumo, a implementação do código de coleta de dados é crucial para a qualidade e abrangência do conjunto de dados disponível. Este conjunto de dados será essencial para a análise e construção dos modelos SINDy, permitindo uma representação robusta e eficaz da dinâmica do sistema motor-gerador em diversas condições operacionais (SOUZA, 2023).

3.1.3 Desenvolvimento do modelo SINDy com a Biblioteca PySINDy

A utilização dos dados coletados é determinante na criação de modelos SINDy, empregando técnicas avançadas de identificação de sistemas dinâmicos. Esses modelos são cruciais para aprofundar a compreensão do comportamento intrínseco do sistema motor-gerador e serão a base para a implementação de controladores eficientes. A ferramenta escolhida para implementar o modelo SINDy será o PySINDy.

Os dados coletados da bancada motor-gerador serão explorados para identificar rela-

ções não lineares entre as variáveis do sistema. A técnica `SINDy` é particularmente adequada para este propósito, pois é capaz de identificar padrões complexos e não lineares nos dados, proporcionando uma representação compacta e interpretação compreensível das dinâmicas do sistema. Segundo (SOUZA, 2023) A estratégia `SINDy` busca obter uma aproximação para cada função de estado a partir de medições reais das entradas, dos estados e possivelmente de suas derivadas.

A implementação do modelo `SINDy` com `PySINDy` oferece uma abordagem eficaz e acessível. O `PySINDy` é uma biblioteca em Python projetada especificamente para realizar a identificação de sistemas dinâmicos esparsos. Ele utiliza métodos de aprendizado de máquina para descobrir as equações que governam o sistema a partir dos dados, possibilitando a geração automática de modelos matemáticos representativos.

Com o modelo `SINDy` em mãos, será possível obter uma representação matemática concisa das interações e comportamentos dinâmicos do sistema motor-gerador. Essa representação matemática será vital para a concepção e implementação de controladores PI e PID eficientes, pois fornecerá insights valiosos sobre como as variáveis do sistema respondem a diferentes entradas e condições operacionais (SOUZA, 2023).

Em resumo, a utilização dos dados coletados para desenvolver modelos `SINDy`, combinada com a implementação prática desses modelos utilizando o `PySINDy`, abrirá caminho para uma compreensão aprofundada do sistema motor-gerador. Esses modelos não apenas auxiliarão na análise do comportamento dinâmico, mas também fornecerão as bases necessárias para a implementação de estratégias de controle avançadas e eficientes.

3.1.4 Desenvolvimento de códigos offline para a otimização dos parâmetros dos controladores

A implementação de um código offline para a otimização dos parâmetros dos controladores PI e PID representa uma etapa vital na busca por um desempenho otimizado do sistema. Este código será desenvolvido para operar de forma independente, permitindo a busca eficiente dos melhores conjuntos de parâmetros sem a necessidade de intervenção em tempo real.

A escolha da biblioteca `PyGad` para esta tarefa é estratégica, dada sua capacidade de realizar otimizações eficientes em problemas complexos. O `PyGad`, baseado em algoritmos genéticos, proporciona uma abordagem evolutiva para encontrar conjuntos ideais de parâmetros que maximizem uma função objetivo predefinida. Essa função objetivo será formulada com base nos critérios de desempenho desejados para o sistema motor-gerador.

O código offline começará com a definição da função objetivo, que pode incluir métricas de desempenho como resposta transitória, erro estático, ou outros indicadores relevantes. A biblioteca `PyGad` será então utilizada para explorar o espaço de busca dos parâmetros do controlador, ajustando-os de forma iterativa ao longo de múltiplas gerações.

Durante o processo de otimização, o `PyGad` empregará conceitos inspirados na evolução

natural, como seleção, cruzamento (crossover) e mutação, para iterativamente ajustar os parâmetros dos controladores. Isso permitirá que o algoritmo descubra conjuntos de parâmetros que levem a um desempenho otimizado do sistema motor-gerador, de acordo com os critérios estabelecidos.

A implementação offline tem a vantagem de poder realizar otimizações de forma mais abrangente, explorando um espaço de parâmetros mais amplo em comparação com abordagens online. Após a conclusão da otimização, os parâmetros ideais identificados podem ser aplicados diretamente no sistema motor-gerador para melhorar seu desempenho em tempo real.

A implementação de um código offline para otimização dos parâmetros dos controladores PI e PID, utilizando a biblioteca `PyGad`, é uma estratégia eficaz para maximizar o desempenho do sistema motor-gerador. Essa abordagem oferece flexibilidade, eficiência e capacidade de adaptação aos requisitos específicos de controle, contribuindo para a obtenção de resultados otimizados.

3.1.5 Desenvolvimento de códigos online para transmissão dos controladores otimizados para a bancada motor-gerador

O desenvolvimento de um código online é essencial para efetuar a transmissão dos controladores otimizados pela Algoritmo Genético (AG) para a bancada motor-gerador em tempo real. Essa implementação, através da plataforma Arduino, proporcionará a execução dinâmica e contínua do controle, permitindo que o sistema se adapte dinamicamente a variações operacionais.

O código online será projetado para operar em conjunto com o Arduino, uma plataforma versátil e acessível, com ampla compatibilidade de hardware e recursos de entrada/saída. A comunicação entre o código e o Arduino será estabelecida por meio da biblioteca Serial, que possibilita a transmissão eficiente de dados entre o computador e o microcontrolador.

A função principal do código online será receber os parâmetros do controlador otimizados pela AG, transmiti-los para o Arduino e garantir a aplicação contínua desses parâmetros no sistema motor-gerador. Este processo ocorrerá em tempo real, permitindo que o sistema se ajuste instantaneamente a mudanças nas condições operacionais.

A implementação online oferece diversas vantagens, especialmente quando se trata de sistemas dinâmicos sujeitos a variações e perturbações. A capacidade de adaptação em tempo real permite que o controlador otimizado pela AG responda dinamicamente a alterações nas características do sistema, garantindo um desempenho robusto e eficaz.

A utilização do Arduino como plataforma de controle proporciona uma solução eficiente e de baixo custo. Sua capacidade de interface com uma variedade de sensores e atuadores torna-o ideal para aplicações de controle em tempo real, como é o caso da bancada motor-gerador.

O desenvolvimento do código online para transmitir os controladores otimizados pela AG para a bancada motor-gerador, via Arduino, é uma etapa crucial para garantir a implementação eficaz do controle em tempo real. Essa abordagem assegura que o sistema se adapte dinamicamente às variações operacionais, otimizando continuamente seu desempenho em resposta às condições em evolução.

3.2 Análise quantitativa dos resultados

A abordagem quantitativa será adotada para analisar os resultados. Espera-se que a combinação entre as técnicas de controle PI e PID, otimizadas por AG, em conjunto com os modelos `SINDy`, culminará em um sistema de controle robusto e eficiente para a bancada motor-gerador. Os resultados derivados dessa abordagem serão submetidos a uma avaliação abrangente, abordando aspectos como desempenho, estabilidade e a capacidade de adaptação a diferentes pontos de operação.

A combinação de técnicas tradicionais de controle, como PI e PID, com a otimização proporcionada por AG promete fornecer conjuntos de parâmetros que maximizam o desempenho do sistema. Essa otimização é crucial, especialmente quando se considera a complexidade dinâmica inerente a sistemas como a bancada motor-gerador. A capacidade dos Algoritmos Genéticos em explorar de forma abrangente o espaço de parâmetros do controlador oferece uma vantagem significativa na busca por soluções que atendam aos critérios de desempenho estabelecidos.

A inclusão dos modelos `SINDy` nesse contexto acrescenta uma camada de compreensão adicional ao sistema. Esses modelos, derivados da identificação de padrões não lineares nos dados, fornecem uma representação matemática compacta e interpretável das dinâmicas subjacentes do sistema. Isso não apenas contribui para uma compreensão mais profunda do comportamento do motor-gerador, mas também serve como uma base sólida para a sintonia dos controladores PI e PID.

Os resultados obtidos serão avaliados em termos de desempenho, considerando métricas como tempo de resposta, erro estático e dinâmico, e a capacidade de seguir referências desejadas. A estabilidade do sistema será um critério fundamental, garantindo que o controle permaneça eficaz mesmo diante de perturbações ou variações nas condições operacionais. A capacidade de adaptação do sistema a diferentes cenários operacionais também será examinada, assegurando uma resposta consistente em ambientes dinâmicos.

A escolha dessa metodologia visa proporcionar uma análise aprofundada e abrangente das estratégias de controle propostas, utilizando técnicas avançadas como AGs e Modelos `SINDy`, contribuindo para o avanço do conhecimento na área de controle de sistemas dinâmicos.

Ao seguir essa metodologia, espera-se atingir os objetivos propostos e obter resultados relevantes para a eficiência do controle na bancada motor-gerador, além de contribuir para a aplicação prática de técnicas avançadas de controle em sistemas dinâmicos complexos.

CAPÍTULO

4

RESULTADOS

4.1 Introdução aos Resultados

A seção de resultados deste trabalho oferece uma visão aprofundada sobre a implementação bem-sucedida de um sistema de controle ótimo baseado em AG, utilizando a linguagem de programação Python com a biblioteca `PyGad` e integrando-se à plataforma Arduino. Como estratégia para otimizar o desempenho de sistemas dinâmicos em tempo real, o emprego do AG demonstra eficácia na busca por soluções ótimas para problemas complexos de controle.

A abordagem adotada neste estudo compreende três fases fundamentais: a criação do modelo matemático do sistema dinâmico, a implementação do algoritmo genético e a integração eficaz com a plataforma Arduino para a realização das ações de controle. O modelo matemático é essencial para a avaliação do desempenho em tempo real, enquanto o AG atua na otimização dos parâmetros do controlador, visando à minimização de uma função objetiva específica. A integração com o Arduino, uma plataforma de baixo custo e amplamente acessível, proporciona a execução prática do controle em sistemas físicos, ampliando as aplicações potenciais.

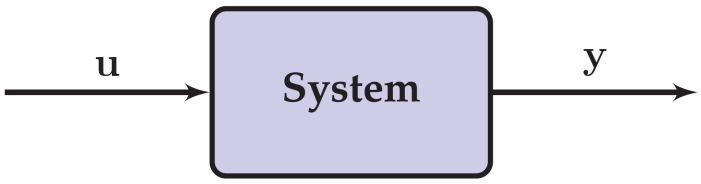
Ao explorar esta seção, os resultados detalhados revelarão a eficiência e rapidez do algoritmo genético na otimização do controle de sistemas dinâmicos. Serão apresentados gráficos, métricas de desempenho e análises que evidenciam a robustez do sistema implementado. A diversidade de sensores e atuadores disponíveis na plataforma Arduino possibilita a aplicação deste controle ótimo em diferentes contextos da engenharia, desde automação industrial até robótica e controle de processos.

Dessa forma, a seção de resultados é central para validar a viabilidade e utilidade prática desta abordagem, fornecendo insights valiosos sobre a eficácia do AG na otimização do desempenho de sistemas dinâmicos em tempo real.

4.2 Coleta de Dados da Bancada Motor-Gerador

A execução da coleta de dados foi realizada em um sistema de malha aberta, figura 10, indicando que o sistema em questão não possui um mecanismo de realimentação para ajustar ou corrigir a saída com base nas condições atuais. Nesse contexto, a ausência de um controle ativo ou feedback durante a coleta de dados sugere que as variáveis do sistema estão sendo observadas sem intervenção direta ou ajuste em tempo real. Essa abordagem é comumente utilizada para adquirir informações iniciais sobre o comportamento natural do sistema antes da implementação de estratégias de controle mais avançadas.

Figura 10 – Sistema em malha aberta



Fonte: (BRUNTON; KUTZ, 2019)

Na condução da coleta de dados, foi adotada uma abordagem sistemática ao fixar a tensão de alimentação (V_{cc}) em $7V$. Para cada coleta, o setpoint (Spu), chamado neste trabalho de ponto de operação, foi cuidadosamente estabelecido, definindo condições específicas de operação. Esse procedimento visa avaliar o desempenho do sistema em diferentes níveis de tensão de entrada, proporcionando uma análise abrangente do comportamento dinâmico do motor-gerador. Esses parâmetros foram registrados em um arquivo CSV denominado `ci.csv`.

Na primeira coleta, o setpoint foi fixado em $3.5V$, representando uma condição inicial que ultrapassa a zona morta e fornece dados sobre a resposta do sistema em uma região onde o motor-gerador está ativo.

Para a segunda coleta, o setpoint foi ajustado para $4V$, situando-se em uma região intermediária. Essa configuração permite observar como o sistema reage em uma faixa próxima à zona morta, oferecendo informações cruciais sobre o comportamento do motor-gerador em condições moderadas de tensão de entrada.

Na terceira coleta, o setpoint foi estabelecido em $5V$, ultrapassando a tensão de saturação. Essa condição representa uma operação mais extrema, possibilitando a avaliação do desempenho do sistema em situações próximas ao limite de sua capacidade.

Ao adotar essa abordagem multiponto, a fixação da tensão de alimentação em 7V, juntamente com a variação dos setpoints em cada coleta, proporciona uma visão abrangente do sistema em diferentes cenários operacionais. Essa estratégia é fundamental para identificar padrões de resposta, analisar a estabilidade do sistema e otimizar o controle, contribuindo para a compreensão completa do comportamento dinâmico do motor-gerador em toda a faixa

de operação.

Há ainda a definição de parâmetros de coleta que desempenha um papel crucial ao estabelecer configurações essenciais para a coleta de dados do sistema de controle.

Esses parâmetros desempenham um papel central na configuração do experimento, influenciando a dinâmica do sistema, a quantidade de dados coletados e a interpretação dos resultados. A escolha cuidadosa desses valores é fundamental para assegurar uma coleta de dados eficaz e representativa do comportamento do sistema em questão.

O estabelecimento da conexão serial é fundamental para estabelecer a comunicação bidirecional entre o programa em Python e a placa Arduino. Neste trabalho estabeleceu-se a conexão serial com o dispositivo externo conectado à porta COM3, com uma taxa de transmissão de 9600 *bauds*.

O código implementado gerou uma onda quadrada com características definidas, sendo esta uma entrada de referência (r) para a bancada motor-gerador. A resposta do sistema (y) foi continuamente monitorada e registrada ao longo do tempo.

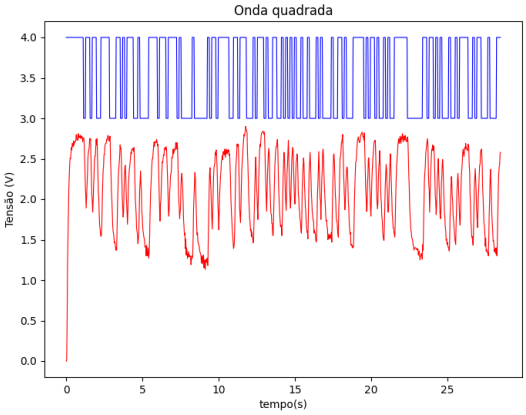
É necessário o encerramento da comunicação serial com o dispositivo externo (microcontrolador) após a conclusão da coleta de dados, marcando o término da coleta de dados e garantindo a integridade do sistema.

Adicionalmente, a coleta de dados incluiu a criação de um arquivo CSV ($'Dados.csv'$) que armazenou as variáveis de entrada (u) e saída (y).

A obtenção da resposta do sistema em tempo real e a visualização por meio de um gráfico são etapas cruciais na análise do comportamento do sistema. No contexto do código fornecido, as ações realizadas permitem uma avaliação dinâmica do sistema, incluindo a representação gráfica do sinal de entrada `(PRBS)` em azul e a resposta do sistema ao longo do tempo em vermelho.

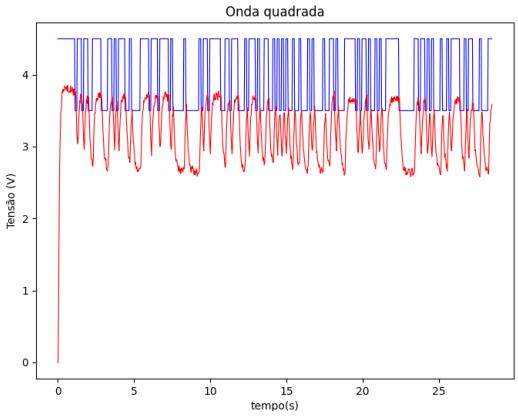
Ao obter dados de três pontos de operação distintos, o código oferece uma perspectiva abrangente do desempenho do sistema em condições variadas. A geração de gráficos, figuras 11, 12 e 13, proporciona uma representação visual dessas respostas, facilitando a interpretação e análise dos resultados.

Figura 11 – Dados coletados 3.5V.



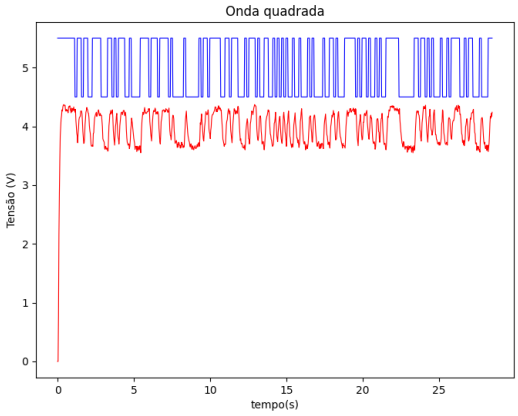
Fonte: O Autor

Figura 12 – Dados coletados 4V.



Fonte: O Autor

Figura 13 – Dados coletados 5V.



Fonte: O Autor

O processo de coleta de dados foi efetuado de forma eficiente, com a comunicação serial estabelecida e a resposta do sistema sendo registrada com sucesso. Os dados coletados

são fundamentais para as fases subsequentes do trabalho, incluindo a criação do modelo matemático do sistema e a implementação do algoritmo genético para otimização do controle. O arquivo 'setpointspu.csv' foi gerado para armazenar o valor médio da resposta do sistema, que servirá como setpoint para as etapas futuras do projeto.

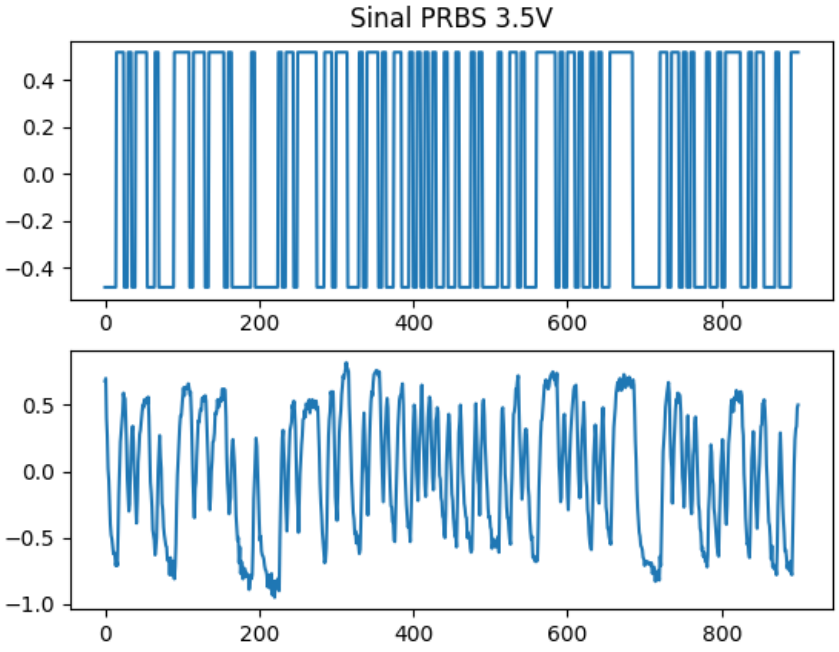
4.3 Validação do Modelo `SINDy` com Dados da Bancada Motor-Gerador

A validação do modelo `SINDy` com os dados provenientes da bancada motor-gerador é uma etapa crucial para avaliar a capacidade do modelo em representar fielmente o comportamento dinâmico do sistema. Inicialmente, foram coletados dados da resposta do sistema à entrada, utilizando a configuração e comunicação previamente estabelecidas. Estes dados foram registrados e tratados para posterior utilização no processo de identificação e validação do modelo.

Os dados coletados foram carregados e visualizados, proporcionando uma compreensão inicial do comportamento do sistema. Um número de amostras (N_t) foi eliminado do transiente inicial, e as séries temporais de entrada (u) e saída (y) foram plotadas para uma análise exploratória inicial. Posteriormente, foi realizado o tratamento dos dados para remoção da média, visando normalizar o comportamento do sistema.

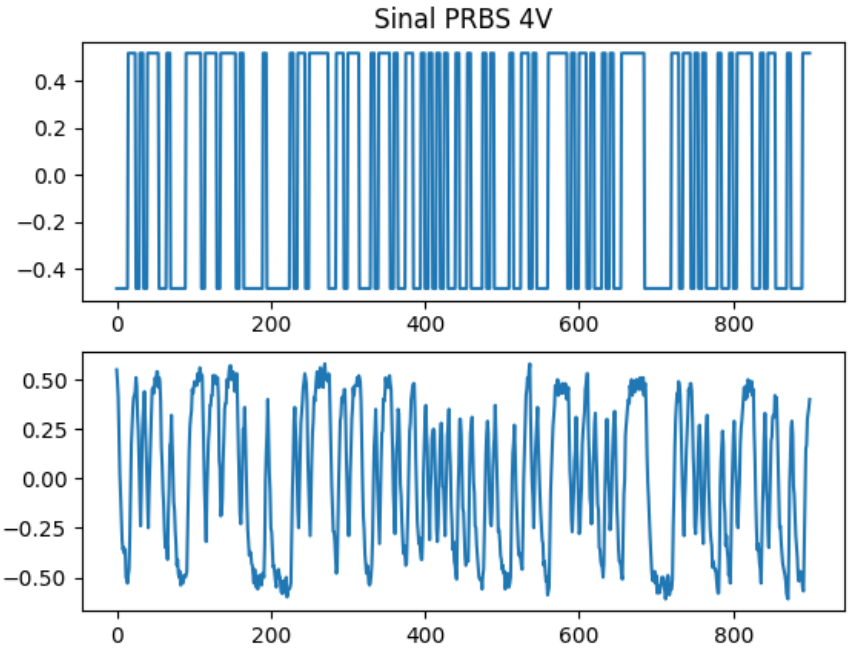
As séries temporais foram plotadas após esse processo, evidenciando a remoção da componente constante. O resultado deste processo é mostrado nas figuras 14, 15 e 16.

Figura 14 – Séries temporais de entrada (u) e saída (y) para o ponto de operação 3.5V.



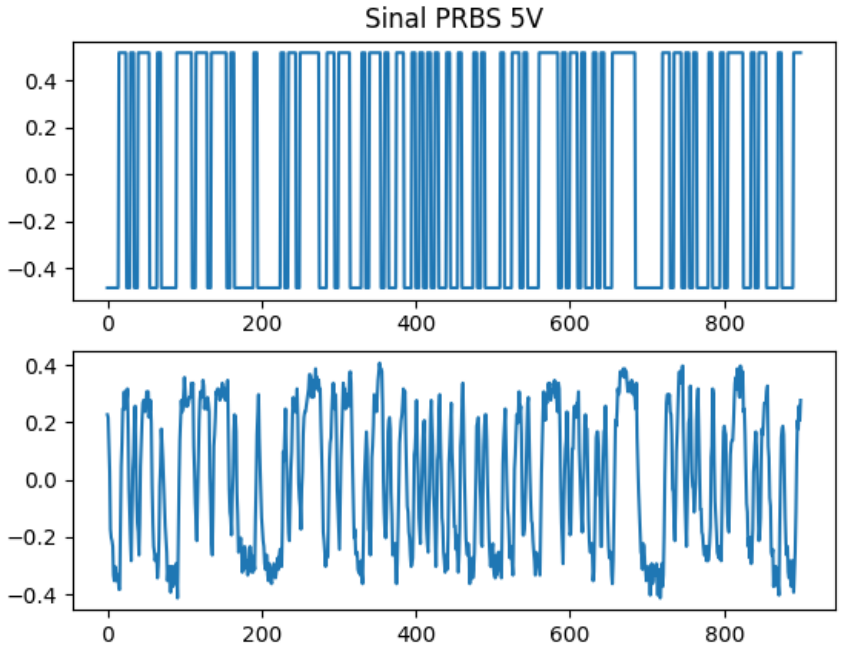
Fonte: O Autor

Figura 15 – Séries temporais de entrada (u) e saída (y) para o ponto de operação 4V.



Fonte: O Autor

Figura 16 – Séries temporais de entrada (u) e saída (y) para o ponto de operação 5V.



Fonte: O Autor

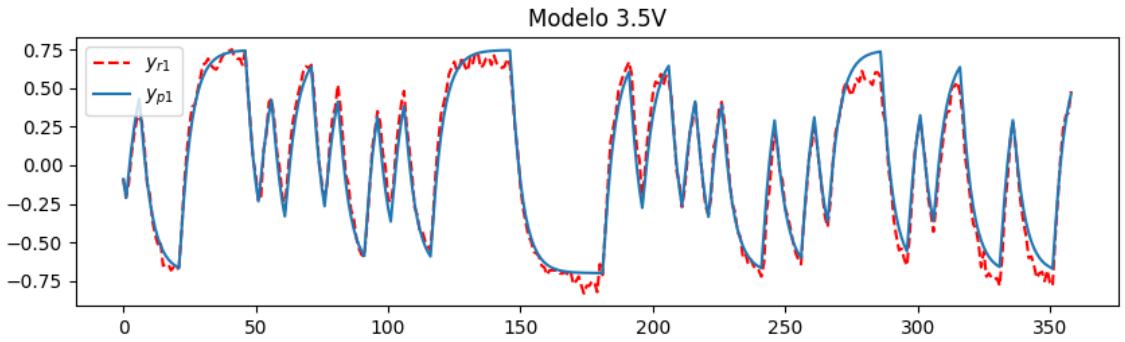
O código divide o conjunto total de dados experimentais em dois subconjuntos distintos: um conjunto usado para identificar o modelo e outro para validar o modelo identificado. O conjunto de identificação é usado para treinar o modelo, enquanto o conjunto de validação

é mantido de lado e utilizado posteriormente para avaliar a capacidade do modelo de fazer previsões precisas em dados não vistos anteriormente.

A aplicação do Método `SINDy` para identificação do modelo é responsável por utilizar o método `SINDy` para identificar um modelo matemático que descreve o sistema dinâmico a partir dos dados experimentais coletados. O método `SINDy` procura automaticamente relações esparsas entre as variáveis, simplificando o modelo enquanto preserva a informação essencial para descrever o sistema dinâmico. O modelo resultante pode ser utilizado para simulações, análises e controle do sistema.

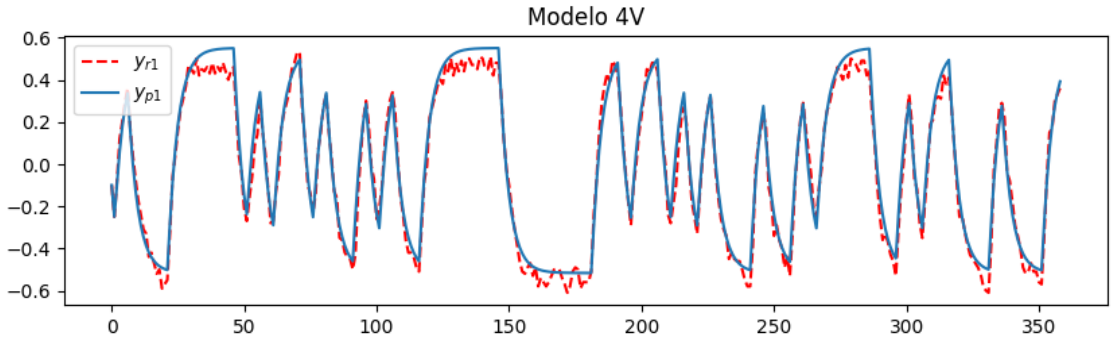
O modelo `SINDy` foi então utilizado para simular a saída do sistema durante o período de validação, sendo comparado com os dados reais para avaliação do ajuste do modelo. As figuras 17, 18 e 19 apresentam a comparação entre a saída simulada e os dados reais durante o período de validação.

Figura 17 – Comparação entre a saída simulada e os dados reais durante o período de validação para o ponto de operação 3.5V.



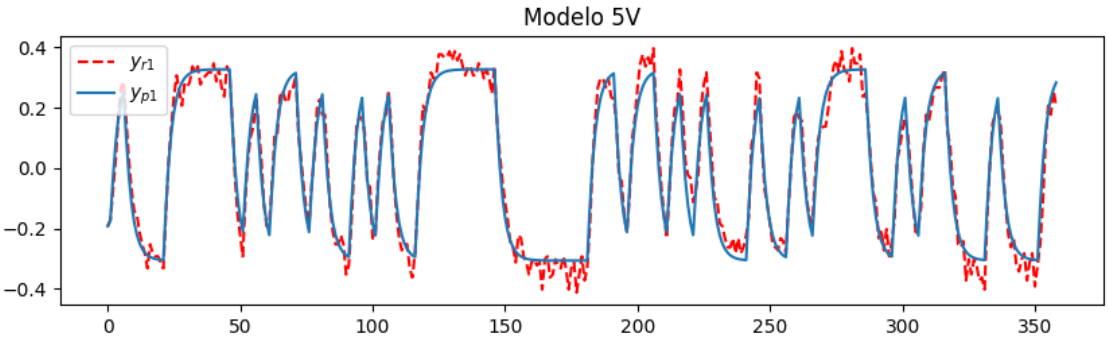
Fonte: O Autor

Figura 18 – Comparação entre a saída simulada e os dados reais durante o período de validação para o ponto de operação 4V.



Fonte: O Autor

Figura 19 – Comparação entre a saída simulada e os dados reais durante o período de validação para o ponto de operação 5V.



Fonte: O Autor

A qualidade do ajuste foi quantificada utilizando o critério de ajuste $NRMSE$ (*Normalized Root Mean Square Error*), que é apresentado como um indicador de adequação do modelo aos dados reais. O valor do $NRMSE$ foi calculado e registrado para avaliação.

Este processo de validação é essencial para assegurar a confiabilidade do modelo `SINDy` na representação do comportamento do sistema motor-gerador. Os resultados obtidos são fundamentais para a continuidade do trabalho, fornecendo insights cruciais para a implementação bem-sucedida do controle ótimo baseado em algoritmo genético.

4.4 Otimização Eficiente com `PyGad`

A aplicação do `PyGad` na sintonia de controladores PI e PID envolve a definição de uma função custo (*fitness*) que represente a performance do controlador para um conjunto específico de parâmetros. Apesar do `PyGad` ser uma função de maximização, a função custo pode ser minimizada pelo algoritmo genético ao indicarmos que desejamos uma solução negativa ao AG, desta forma, ele gera sucessivas populações de soluções candidatas (conjuntos de parâmetros) até encontrar uma configuração que minimize a função de avaliação.

A abordagem genética do `PyGad` simula o processo de evolução natural, utilizando conceitos de seleção natural, recombinação genética e mutação para explorar eficientemente o espaço de busca de parâmetros (GAD, 2021). Essa característica é particularmente valiosa para a sintonia de controladores, onde a interação complexa entre diferentes parâmetros requer uma abordagem sistemática e adaptativa.

Ao adotar o `PyGad` para a sintonia de controladores PI e PID, os seguintes passos gerais são seguidos:

1. **Definição da Função Custo:** - A função custo (*fitness*) é projetada para quantificar o desempenho do controlador para um determinado conjunto de parâmetros. Isso pode incluir critérios como a minimização do erro, resposta transitória desejada e estabilidade do sistema.

2. **Configuração dos Parâmetros do AG:** - Parâmetros do AG, como tamanho da população, número de gerações, taxa de *crossover* e taxa de mutação, são configurados para otimizar o desempenho do `PyGad` na busca dos parâmetros ideais.
3. **Implementação da Função Custo no `PyGad`:** - A função custo é incorporada no `PyGad`, permitindo que o algoritmo genético avalie a aptidão de cada solução candidata com base em sua performance no sistema dinâmico.
4. **Execução do AG:** - O `PyGad` é executado, gerando sucessivas gerações de soluções candidatas até convergir para um conjunto de parâmetros que minimize a função de avaliação.
5. **Análise dos Resultados:** - Os resultados obtidos pelo `PyGad` são analisados para determinar a eficácia do controlador otimizado. Isso inclui a avaliação da estabilidade, resposta transitória e capacidade de adaptação a diferentes condições de operação.

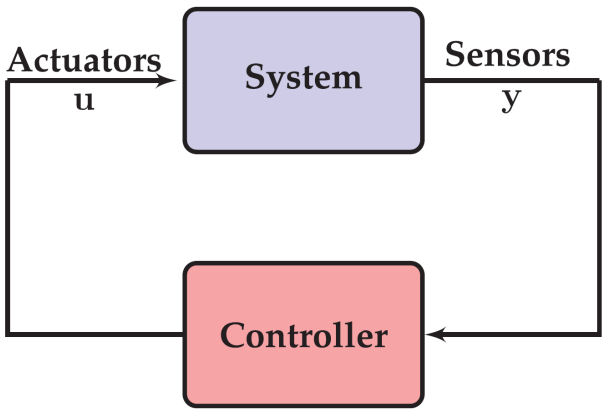
A abordagem genética oferecida pelo `PyGad` proporciona uma solução eficaz para encontrar configurações de controladores que atendam aos requisitos específicos de desempenho em uma variedade de situações operacionais.

4.5 Sintonia de Controladores PI e PID com `PyGad`

Neste estudo, serão exploradas e avaliadas duas topologias distintas de controladores lineares, especificamente, uma do tipo PI e outra PID. Cada uma dessas topologias apresenta características únicas em termos de desempenho e complexidade.

A malha de controle, figura 20, será cuidadosamente delineada para cada uma das topologias consideradas. Isso incluirá a identificação clara dos elementos da malha, como o sistema dinâmico, o sensor de feedback, o controlador e o atuador. A interação entre esses componentes é fundamental para o entendimento do comportamento global do sistema de controle.

Figura 20 – Sistema em malha fechada



Fonte: (BRUNTON; KUTZ, 2019)

Além disso, é definida uma função custo associada à otimização dos parâmetros dos controladores por meio do AG. A função custo assume uma função essencial na avaliação e ajuste dos controladores, sendo meticulosamente elaborada para espelhar os objetivos almejados em termos de desempenho do sistema.

A função custo J geralmente é usada em problemas de otimização para avaliar quão bem um determinado conjunto de parâmetros (nesse caso, os valores em u) atende aos objetivos desejados. A expressão $Q \cdot \sum (r - y)^2$ penaliza a diferença entre o vetor de referência r e a saída do sistema y, enquanto $R1 \cdot \sum u^2$ penaliza a magnitude do vetor de entrada u.

A função `penalidade(u)`, é uma função de penalização que avalia se algum valor no vetor de entrada `u` está fora de certos limites especificados. Se algum dos valores de `u` for maior que 7 ou menor que 2.5, a função retorna um valor alto de 1000 como penalidade. Isso indica que a otimização deve evitar fortemente valores de entrada que excedam esses limites.

A função é usada para impor penalidades significativas quando as entradas ultrapassam faixas desejadas. Essas penalidades podem ser úteis em problemas de otimização nos quais é importante manter as variáveis de entrada dentro de certos limites para garantir um comportamento seguro ou adequado do sistema.

Em resumo, a função custo esta sendo calculada para refletir o desempenho do sistema em relação aos objetivos desejados, levando em consideração a diferença entre a saída real e a referência, a magnitude da entrada e possíveis penalidades adicionais.

Ao considerar ambas as topologias (PI e PID) e a aplicação do AG, este estudo visa não apenas comparar o desempenho dos controladores lineares, mas também destacar a eficácia do uso de algoritmos evolutivos na sintonia adaptativa desses controladores. Isso proporcionará insights valiosos para a escolha e otimização de controladores em sistemas dinâmicos.

4.5.1 Discretização de Controladores: Método `Tustin`

O código implementa a discretização de controladores PI e PID por meio do método `Tustin`. Esse método é uma técnica comum para converter controladores contínuos em suas formas discretas, permitindo a aplicação eficaz em sistemas de controle digital. A classe **PI** e **PID** apresentam métodos específicos, como `discretizar_controlador`, que utilizam a biblioteca `control` para realizar essa transformação, garantindo a aplicabilidade prática desses controladores em ambientes digitais.

4.5.2 Salvando Melhores Ganhos para Análise Futura

A classe **MelhoresGanhos** é uma adição valiosa, especialmente quando se utiliza algoritmos genéticos para otimização de parâmetros. O método `salvar_populacoes` é responsável por capturar os melhores ganhos de cada iteração da otimização. Esses ganhos são armazenados na lista `self.ganhos`. Posteriormente, o método `salvar` possibilita a persistência desses dados em arquivos pickle, tornando-os prontamente disponíveis para análises futuras.

Essa abordagem permite a documentação e avaliação dos conjuntos de ganhos que demonstraram melhor desempenho durante o processo de otimização. Analisar esses dados pode oferecer insights valiosos sobre as características do sistema e contribuir para decisões informadas na escolha dos parâmetros de controle.

4.5.3 Controle PI Offline

Ao adaptarmos os parâmetros K_p e K_i com base nos dados do sistema motor-gerador, estabelecemos o controle para os três pontos de operação. Os coeficientes do controlador foram otimizados para minimizar a função custo, considerando o erro entre a saída desejada e a saída real, juntamente com a magnitude do sinal de controle. O AG explorou o espaço de busca dos ganhos do controlador PI.

O código desenvolvido implementa uma função denominada **AjusteControlador** que serve como a função custo (fitness). Aqui está uma descrição do que cada parte do código realiza:

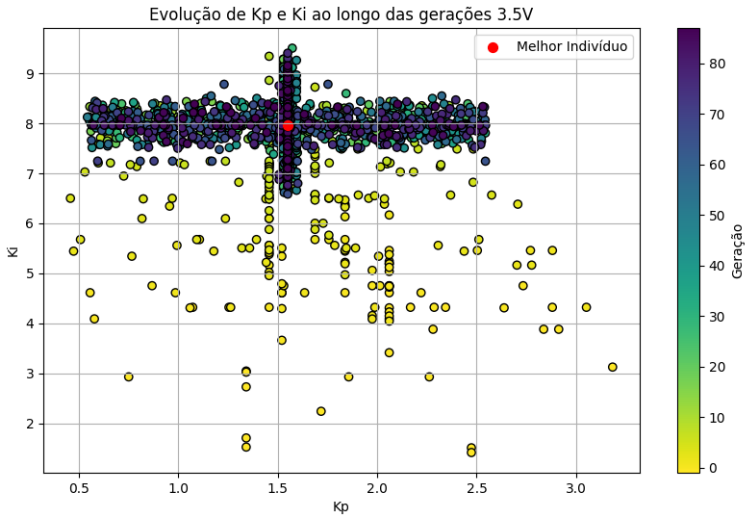
1. **Definição dos Parâmetros do Controlador:**
   - `K1` e `K2`: São os parâmetros do controlador PI, que serão otimizados pelo AG.
   - `Ts`: É o período de amostragem.
2. **Criação do Controlador PI:**
   - `controlador`: Instância da classe `PI`, com os parâmetros K_1 e K_2 e o período de amostragem Ts.
   - `C`: É o controlador PI discretizado, presumivelmente implementado no método `discretizar_controlador`.
3. **Resposta da Malha Entrada-Saída e Entrada-Controle:**
   - `Hy`: Malha de realimentação para a resposta entrada-saída.
   - `Hu`: Malha de realimentação para a resposta entrada-controle.
   - `y`: Resposta do sistema à entrada r (presumivelmente um sinal de referência) considerando a malha entrada-saída.
   - `u`: Resposta do sistema à entrada r considerando a malha entrada-controle.
4. **Cálculo do Custo (Função de Aptidão):**
   - `Q` e `R1`: São parâmetros de ponderação na função de custo.
   - `J`: Função custo que incorpora a soma ponderada dos quadrados da diferença entre a referência r e a saída do sistema y, a magnitude do sinal de controle u, e penalidades adicionais definidas pela função `penalidade(u)`.

- O valor retornado `-J` é usado porque muitos algoritmos de otimização tentam minimizar a função de custo, e aqui o objetivo é maximizar a função de aptidão, que é equivalente a minimizar o negativo da função de custo.

Essa função é utilizada em um AG para ajustar os parâmetros do controlador PI, buscando minimizar a função custo, refletindo um desempenho desejado do sistema de controle.

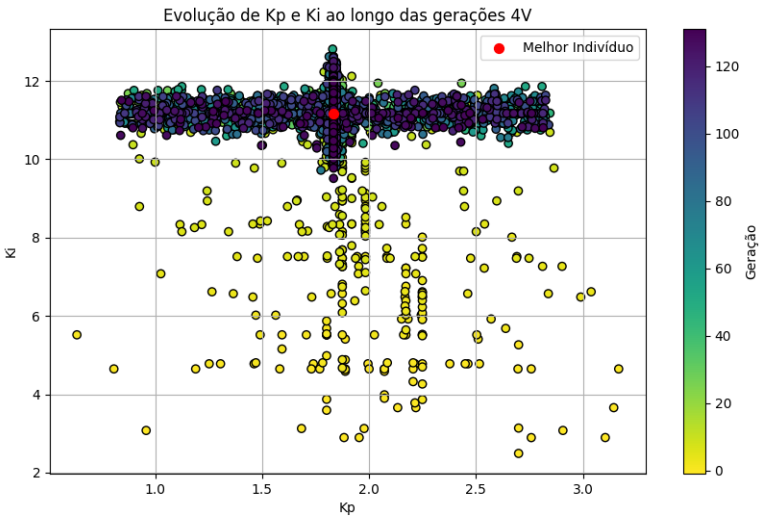
Ao observar as figuras 21, 22 e 23 é evidente a evolução dos parâmetros K_p e K_i durante as iterações do AG. Destaca-se o melhor indivíduo encontrado, indicando a convergência para uma solução otimizada.

Figura 21 – Evolução de K_p e K_i ao longo das gerações para o ponto de operação 3.5V.



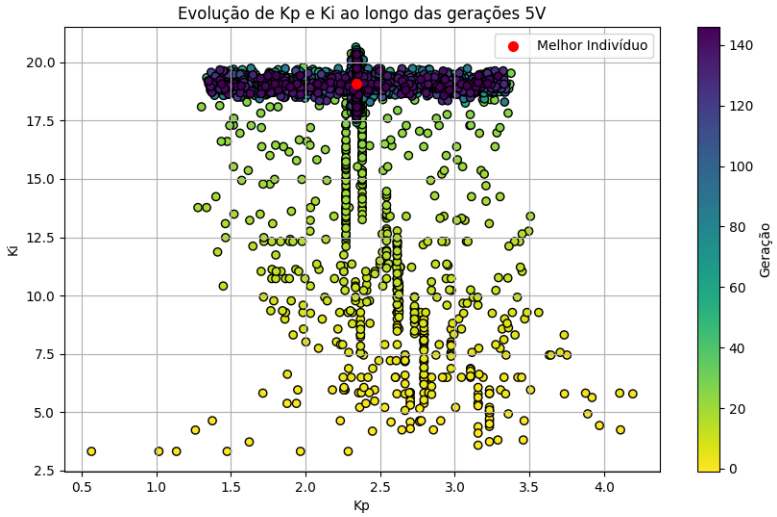
Fonte: O Autor

Figura 22 – Evolução de K_p e K_i ao longo das gerações para o ponto de operação 4V.



Fonte: O Autor

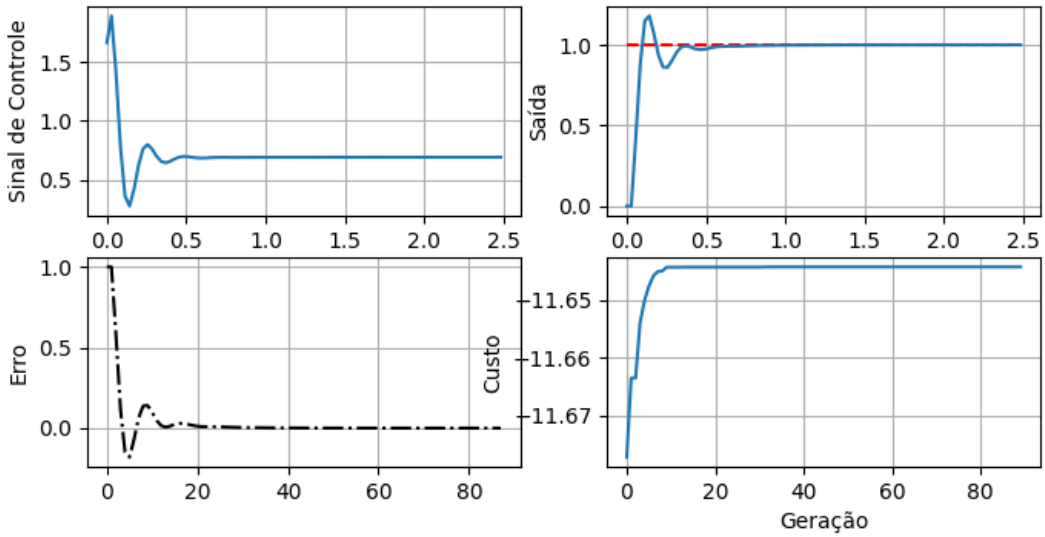
Figura 23 – Evolução de K_p e K_i ao longo das gerações para o ponto de operação 5V.



Fonte: O Autor

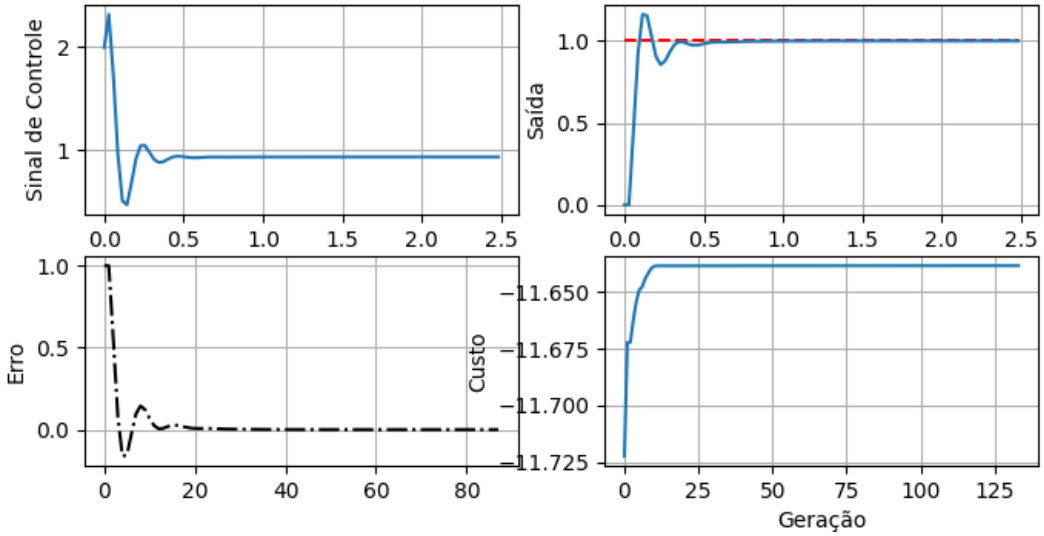
As figuras 24, 25 e 26, demonstram a resposta do sistema à entrada de referência, o sinal de controle, o erro e a função custo ao longo das iterações. Estes resultados representam a configuração final do controlador PI para os diferentes pontos de operação, refletindo em respostas distintas para cada configuração.

Figura 24 – Configuração final do controlador PI para o ponto de operação 3.5V.



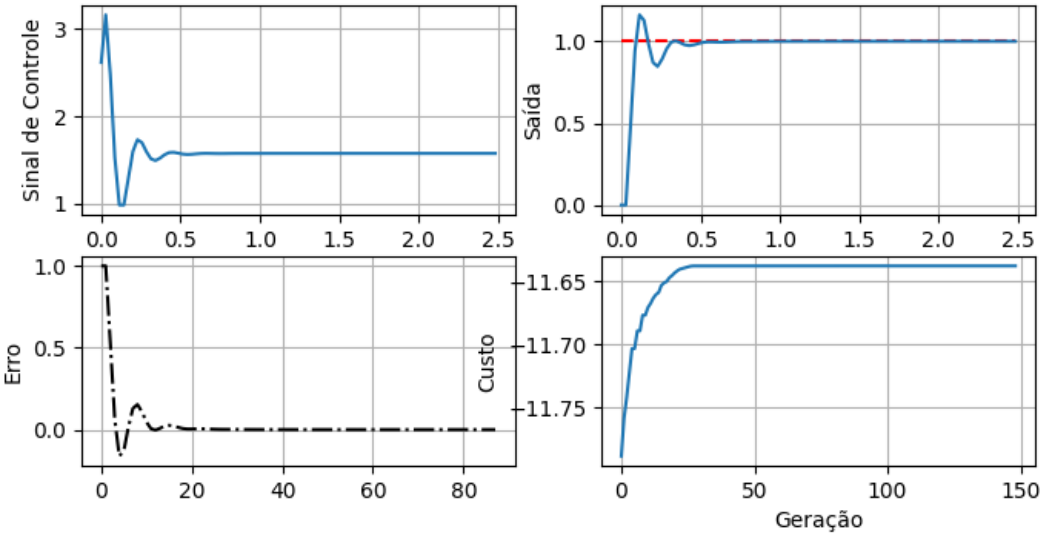
Fonte: O Autor

Figura 25 – Configuração final do controlador PI para o ponto de operação 4V.



Fonte: O Autor

Figura 26 – Configuração final do controlador PI para o ponto de operação 5V.



Fonte: O Autor

4.5.4 Controle PID Offline

O controle PID, além das ações proporcional e integral, incorpora a ação derivativa, proporcional à taxa de variação do erro. Assim como no controle PI, os ajustes dos parâmetros K_p, K_i, e K_d (ganho derivativo) foram realizados offline utilizando a abordagem com AG e a biblioteca `PyGad`.

Essa estratégia consiste em otimizar os ganhos do controlador PID por meio do AG, explorando o espaço de busca para encontrar valores que minimizem uma função de custo específica.

O AG é configurado para iterar através das gerações, ajustando sistematicamente os parâmetros K_p, K_i, e K_d para encontrar uma configuração otimizada.

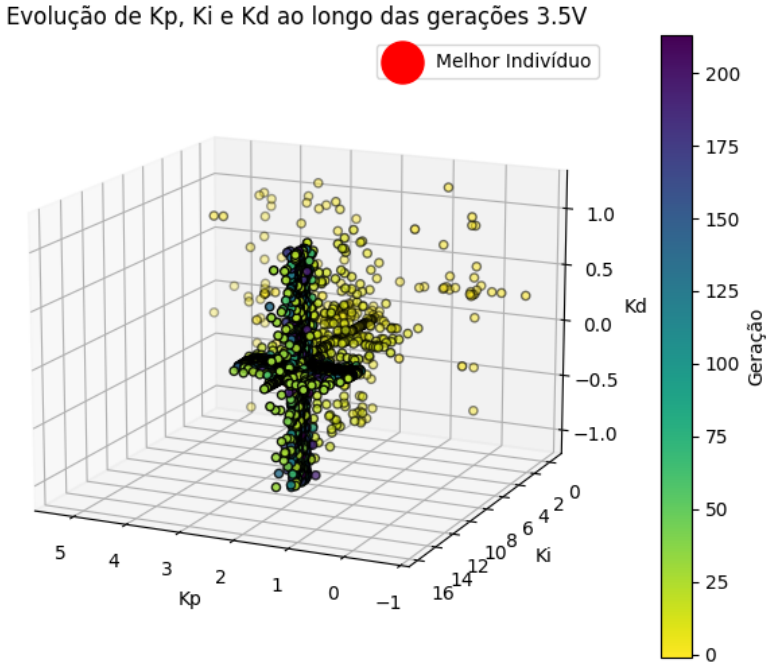
No código é definada uma função chamada `AjusteControlador` que é utilizada como a função custo (fitness). Aqui estão as principais características do código:

1. **Parâmetros de Entrada:**
   - `ga_instance`: Instância do algoritmo genético (`PyGad`).
   - `K`: Vetor contendo os parâmetros do controlador PID (ganho proporcional, ganho integral, ganho derivativo).
   - `idK`: Identificação única do indivíduo na população.
2. **Configuração do Controlador PID:**
   - Os parâmetros do controlador PID (`K1`, `K2`, `K3`) são extraídos do vetor `K`.
   - São definidos parâmetros adicionais, como `a` (um coeficiente), e criado um objeto de controle (`controlador`) da classe PID.
3. **Resposta da Malha Entrada-Saída e Entrada-Controle:**
   - Utilizando as bibliotecas de controle (`ct`), são calculadas as respostas ao degrau da malha entrada-saída e entrada-controle.
   - A resposta ao degrau da malha entrada-saída (`Hy`) é obtida por meio do feedback da série entre o controlador discretizado (`C`) e a planta (`G`).
   - A resposta ao degrau da malha entrada-controle (`Hu`) é obtida pelo feedback direto do controlador (`C`) com a planta (`G`).
4. **Cálculo do Custo (Função de Aptidão):**
   - São definidos os pesos `Q` e `R1`.
   - A função custo `J` é calculada considerando o erro entre a referência (`r`) e a saída do sistema (`y`), a magnitude do sinal de controle (`u`) e uma penalidade associada ao vetor de controle.
   - O objetivo é maximizar a função de aptidão, por isso, a função retorna `-J`.

Este código representa o núcleo da otimização do controlador PID, onde a qualidade do ajuste é avaliada pela função de custo considerando as respostas ao degrau da malha entrada-saída e entrada-controle. A otimização é conduzida pelo algoritmo genético que busca encontrar os parâmetros do controlador que minimizam a função de custo.

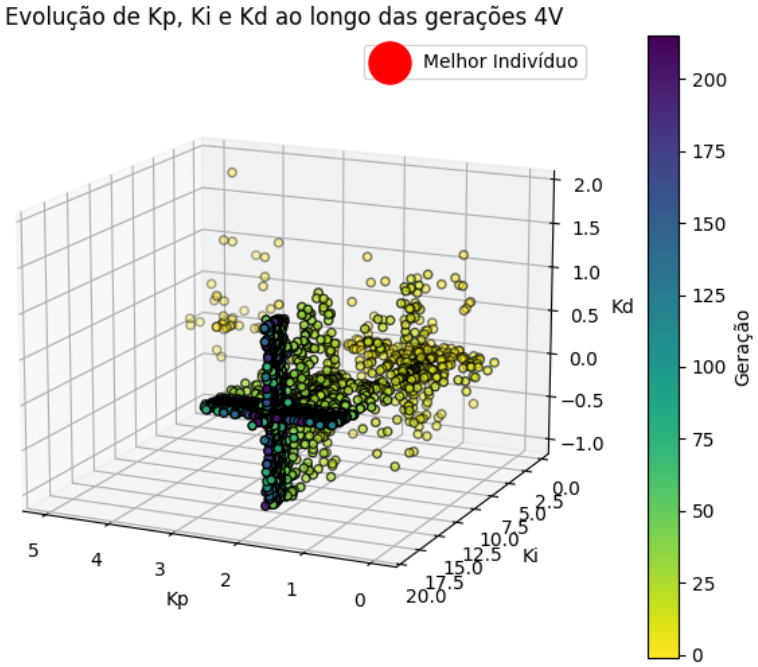
As figuras 27, 28 e 29, apresentam a evolução dos ganhos K_p, K_i, e K_d ao longo das iterações do AG, destacando o melhor indivíduo encontrado que representa a convergência para uma solução otimizada.

Figura 27 – Evolução de K_p, K_i, e K_d ao longo das gerações no controle PID para o ponto de operação 3.5V.



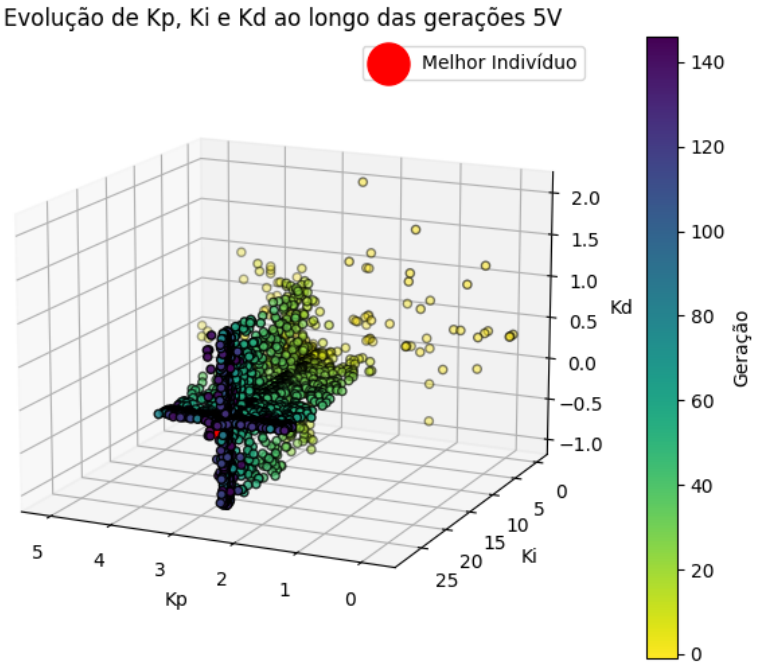
Fonte: O Autor

Figura 28 – Evolução de K_p, K_i, e K_d ao longo das gerações no controle PID para o ponto de operação 4V.



Fonte: O Autor

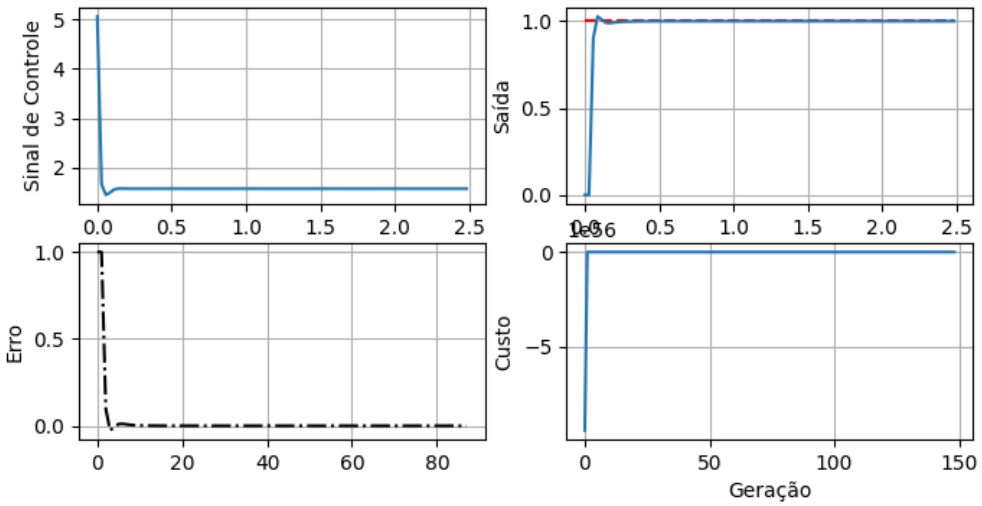
Figura 29 – Evolução de K_p, K_i, e K_d ao longo das gerações no controle PID para o ponto de operação 5V.



Fonte: O Autor

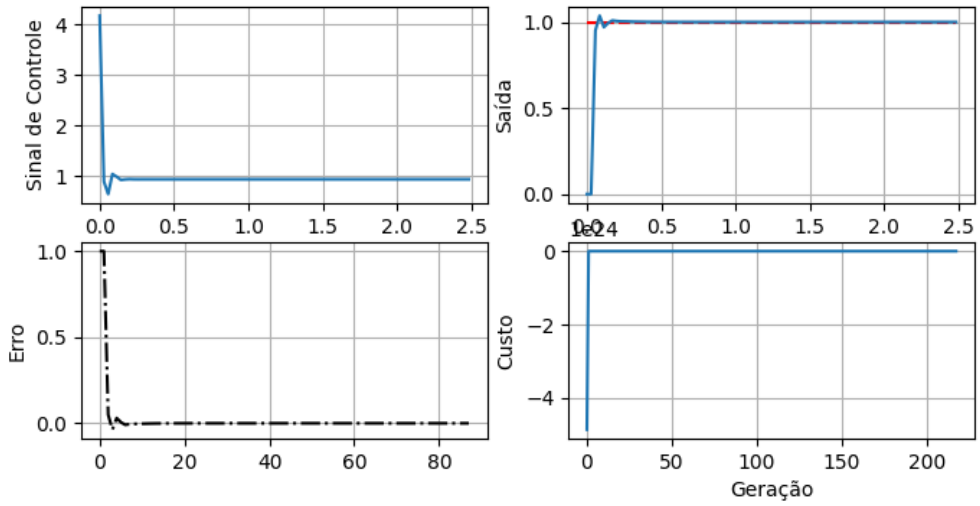
Os resultados finais do controle PID são visualizados nas figuras 30, 31 e 32, mostrando a resposta do sistema à entrada de referência, o sinal de controle, o erro, e a função custo ao longo das iterações. Esses resultados proporcionam uma configuração final do controlador PID otimizado, considerando os diferentes pontos de operação e suas respostas distintas.

Figura 30 – Configuração final do controlador PID otimizado para o ponto de operação 3.5V.



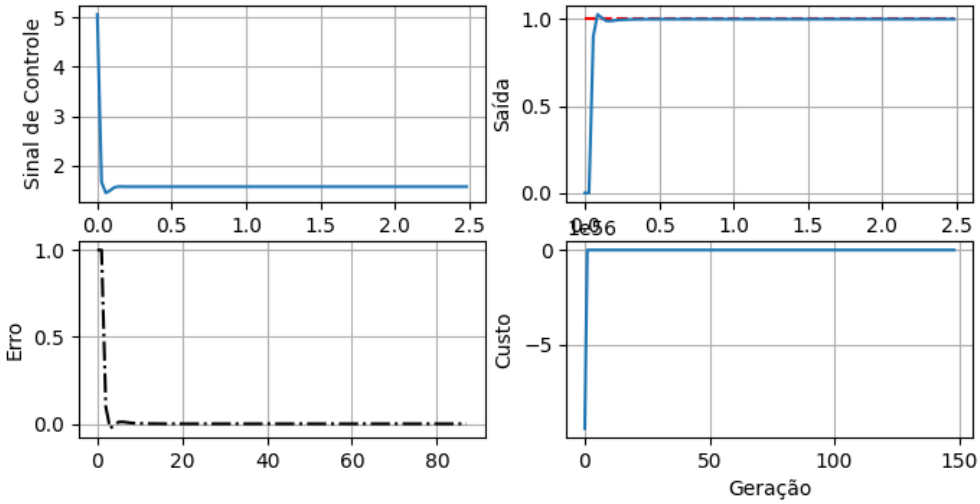
Fonte: O Autor

Figura 31 – Configuração final do controlador PID otimizado para o ponto de operação 4V.



Fonte: O Autor

Figura 32 – Configuração final do controlador PID otimizado para o ponto de operação 5V.



Fonte: O Autor

4.5.5 Resultados da Otimização Offline

Os resultados provenientes da otimização offline evidenciaram uma convergência eficaz do AG. Isso sugere que o AG foi capaz de explorar o espaço de busca de maneira eficiente, ajustando os parâmetros dos controladores para otimizar a função custo. A convergência eficaz indica que o AG atingiu uma configuração de controlador PI e PID que atende aos critérios estabelecidos, proporcionando respostas desejadas para o sistema motor-gerador nos diferentes pontos de operação. Essa eficácia na convergência valida a robustez do processo de otimização offline empregado.

Os valores ótimos dos parâmetros dos controladores estão discriminados na tabela 2.

Tabela 2 – Ganhos dos Controladores PI e PID para os Pontos de Operação

Ponto de Operação	**Tipo de Controlador**	K_p	K_i	K_d
3.5	PI	1.55066354	7.96370323	-
4.0	PI	1.83389219	11.17146209	-
5.0	PI	2.3432469	19.06451886	-
3.5	PID	1.65692946	12.08077753	0.02932723
4.0	PID	1.9402764	16.28779479	0.03066031
5.0	PID	2.46910857	26.20334478	0.03375592

Fonte: O Autor

Em síntese, a aplicação de AGs na otimização offline dos controladores PI e PID demonstrou eficácia ao buscar soluções que minimizam a função de custo, resultando em ganhos apropriados para o sistema motor-gerador. Detalhes completos sobre os resultados e implementação podem ser encontrados nos Apêndices **??** e **??**, o qual inclui os códigos fonte completos utilizados na sintonia dos controladores PI e PID, juntamente com informações adicionais sobre as configurações dos algoritmos genéticos e o emprego da biblioteca `PyGad`.

4.6 Controladores PI e PID Online para Bancada Motor-Gerador com Arduino

A implementação em tempo real dos controladores PI e PID na bancada motor-gerador é executada por meio do Arduino. Essa abordagem proporciona um controle eficiente e dinâmico do sistema, permitindo ajustes contínuos e respostas em tempo real com base nos dados sensoriais fornecidos pela bancada. O Arduino atua como o cérebro do sistema, processando informações, calculando sinais de controle e ajustando parâmetros de acordo com as estratégias específicas do controle PI ou PID. Essa implementação em tempo real é fundamental para garantir um desempenho preciso e adaptável do sistema motor-gerador em diferentes condições operacionais.

4.6.1 Controle Dinâmico em Tempo Real com Arduino

A comunicação bidirecional com o Arduino foi estabelecida de forma eficiente por meio da biblioteca **serial**, permitindo uma troca contínua de dados entre o sistema motor-gerador e o computador. Esse processo possibilita o envio de sinais de controle do computador para o Arduino, que, por sua vez, os utiliza para realizar ajustes em tempo real na entrada do sistema.

A implementação dos controladores PI e PID em tempo real na bancada motor-gerador demonstra sua viabilidade e eficácia. A comunicação contínua com o Arduino possibilita

ajustes dinâmicos, proporcionando respostas rápidas e adaptáveis a variações nas condições operacionais do sistema.

4.6.2 Resultados da Comunicação Python-Arduino e Controle com Controlador PI na Bancada Motor-Gerador

Através da biblioteca **serial**, o Python transmitiu sinais de controle em tempo real, permitindo um ajuste preciso da entrada do sistema. Essa comunicação bidirecional possibilitou uma implementação robusta do controlador PI para regular a dinâmica da bancada.

O código implementa um controlador PI discreto para regular um sistema em tempo real, usando uma abordagem de controle de malha aberta, onde o Arduino recebe sinais de referência, calcula o sinal de controle, e controla um sistema dinâmico. O PI discreto é utilizado para ajustar o sinal de controle com base no erro entre a referência e a saída atual.

A equação do controlador PI discreto pode ser expressa como:

$$u[n] = a_1 \cdot u[n-1] + b_1 \cdot \text{erro}[n] + b_0 \cdot \text{erro}[n-1] \tag{4.1}$$

A equação 4.1 é uma forma discreta de representar um controlador PI em um sistema de controle discreto. Nesta equação:

- $u[n]$ é o sinal de controle na amostra n,
- $u[n-1]$ é o sinal de controle na amostra anterior $(n-1)$,
- $\text{erro}[n]$ é o erro na amostra n (diferença entre a referência e a saída),
- $\text{erro}[n-1]$ é o erro na amostra anterior $(n-1)$,
- a_1, b_1, e b_0 são os coeficientes do controlador.

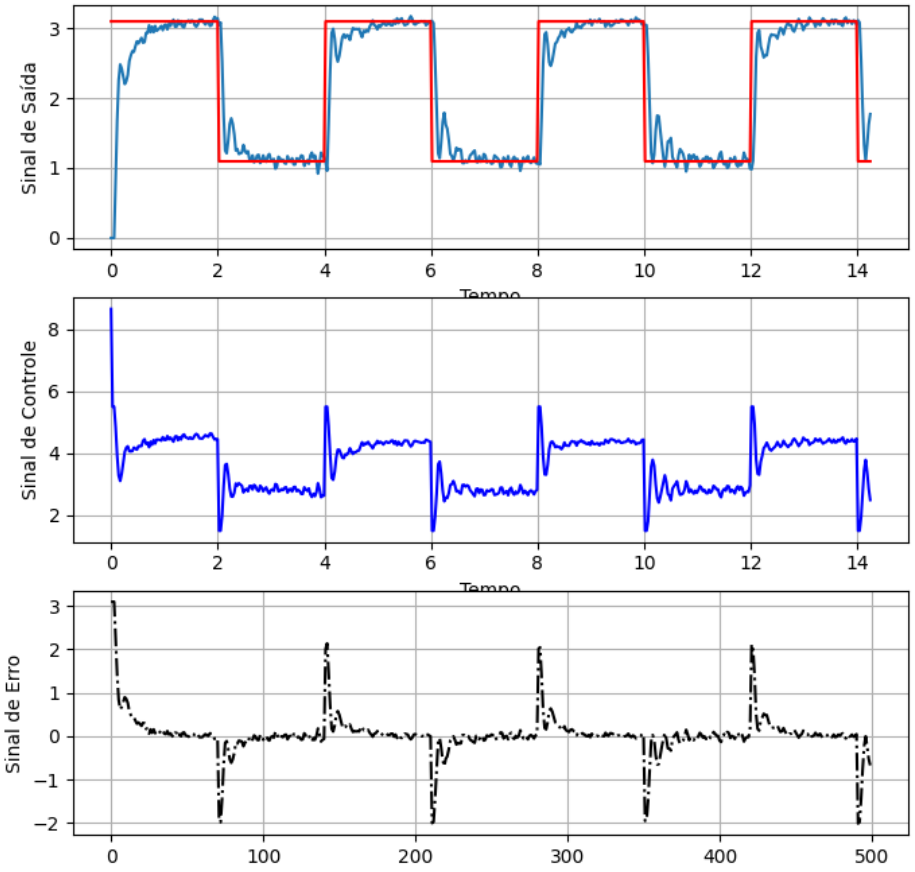
Essa equação representa um controlador PI discreto, onde:

- A parcela $b_1 \cdot \text{erro}[n]$ representa a ação proporcional, proporcional ao erro atual.
- A parcela $b_0 \cdot \text{erro}[n-1]$ representa a ação integral, proporcional ao erro acumulado ao longo do tempo.
- A parcela $a_1 \cdot u[n-1]$ representa uma realimentação do sinal de controle anterior, que pode ser parte da ação integral ou derivativa, dependendo do valor de a_1.

A combinação dessas parcelas ajusta o sinal de controle $u[n]$ com base no desempenho passado e atual do sistema em relação à referência desejada. Os coeficientes a_1, b_1, e b_0 são ajustados para atender aos requisitos específicos do sistema e aos objetivos de controle.

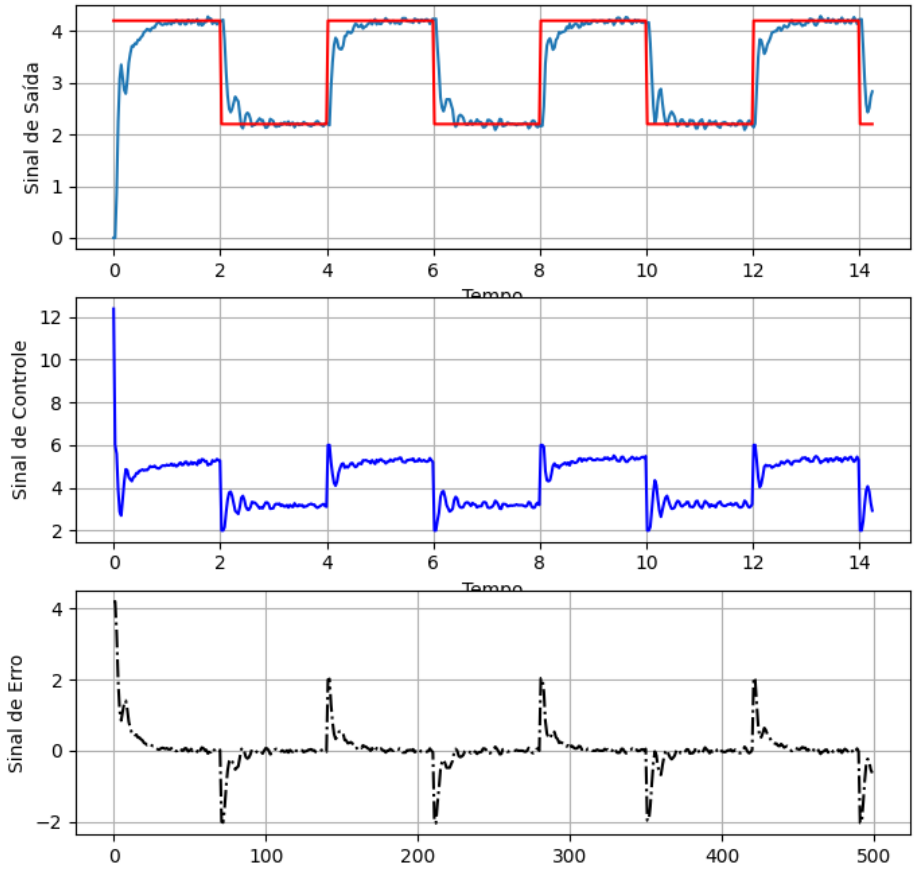
Os resultados, figuras 33, 34 e 35, do controle implementado foram extremamente promissores. O controlador PI foi capaz de responder com agilidade a variações detectadas no sistema motor-gerador, garantindo um desempenho estável e responsivo. Essa dinâmica foi evidenciada pelos ajustes na entrada do sistema em resposta aos sinais de controle enviados pelo Python, refletindo um controle ativo e eficaz.

Figura 33 – Resultado do sistema em resposta aos sinais de controle PI enviados pelo Python para o ponto de operação 3.5V.



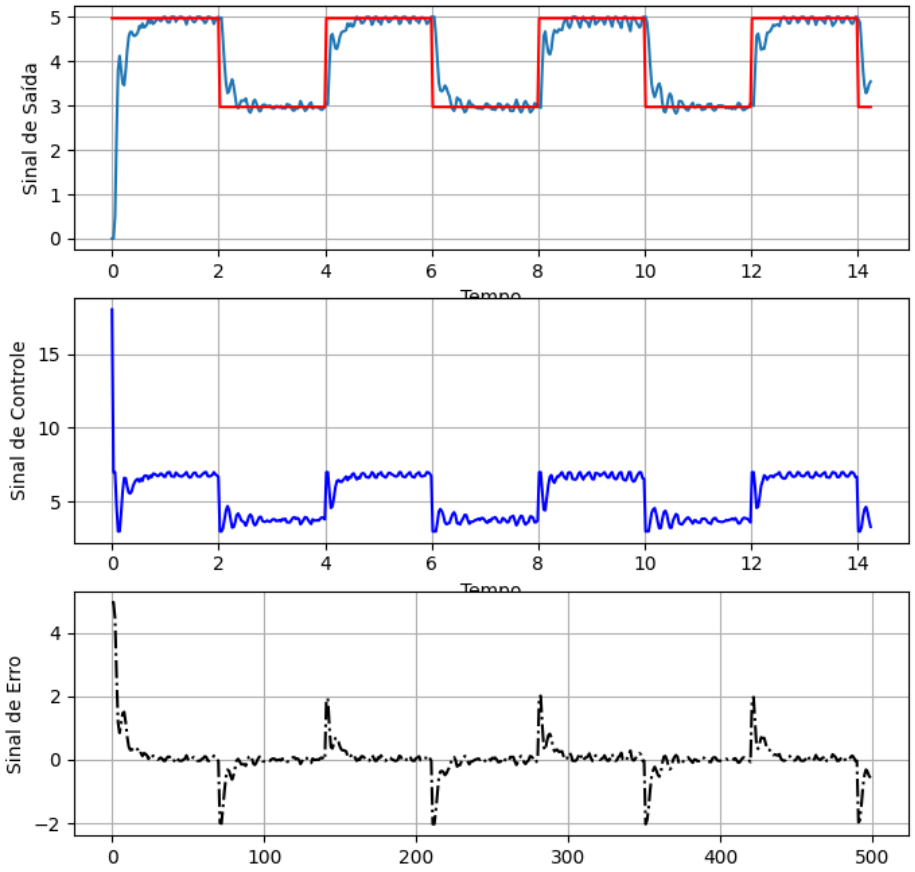
Fonte: O Autor

Figura 34 – Resultado do sistema em resposta aos sinais de controle PI enviados pelo Python para o ponto de operação 4V.



Fonte: O Autor

Figura 35 – Resultado do sistema em resposta aos sinais de controle PI enviados pelo Python para o ponto de operação 5V.



Fonte: O Autor

Os valores obtidos para os ganhos do controlador PI foram fundamentais para a obtenção desses resultados. Ajustados de acordo com os dados experimentais e otimizados com o auxílio de algoritmos genéticos, esses ganhos demonstraram-se eficientes para regular a bancada motor-gerador em diferentes condições de operação.

A precisão e a capacidade de resposta do controlador PI, quando combinadas com a comunicação contínua e confiável entre Python e Arduino, representam uma significativa conquista neste estudo.

4.6.3 Resultados da Comunicação Python-Arduino e Controle com Controlador PID na Bancada Motor-Gerador

Através da biblioteca **serial**, o Python estabeleceu uma comunicação eficaz, transmitindo sinais de controle em tempo real para o Arduino. Essa abordagem permitiu ajustes dinâmicos na entrada do sistema, viabilizando a implementação eficiente do controlador PID.

O código representa a implementação de um controlador PID discreto para controle em tempo real, envolvendo o cálculo do erro, os termos do controlador PID e a comunicação

com o Arduino.

A equação do controlador PID discreto pode ser expressa como:

$$u[n] = -a_1 \cdot u[n-1] - a_2 \cdot u[n-2] + b_0 \cdot \text{erro}[n] + b_1 \cdot \text{erro}[n-1] + b_2 \cdot \text{erro}[n-2] \quad (4.2)$$

A equação 4.2 é uma forma discreta de representar um controlador PID em um sistema de controle discreto. Nesta equação:

- $u[n]$ é o sinal de controle no instante de tempo discreto n,
- a_1 e a_2 são coeficientes associados aos termos de controle retroativo,
- b_0, b_1 e b_2 são coeficientes associados aos termos do sinal de erro,
- $\text{erro}[n]$ é o erro entre a saída desejada e a saída real no instante de tempo n.

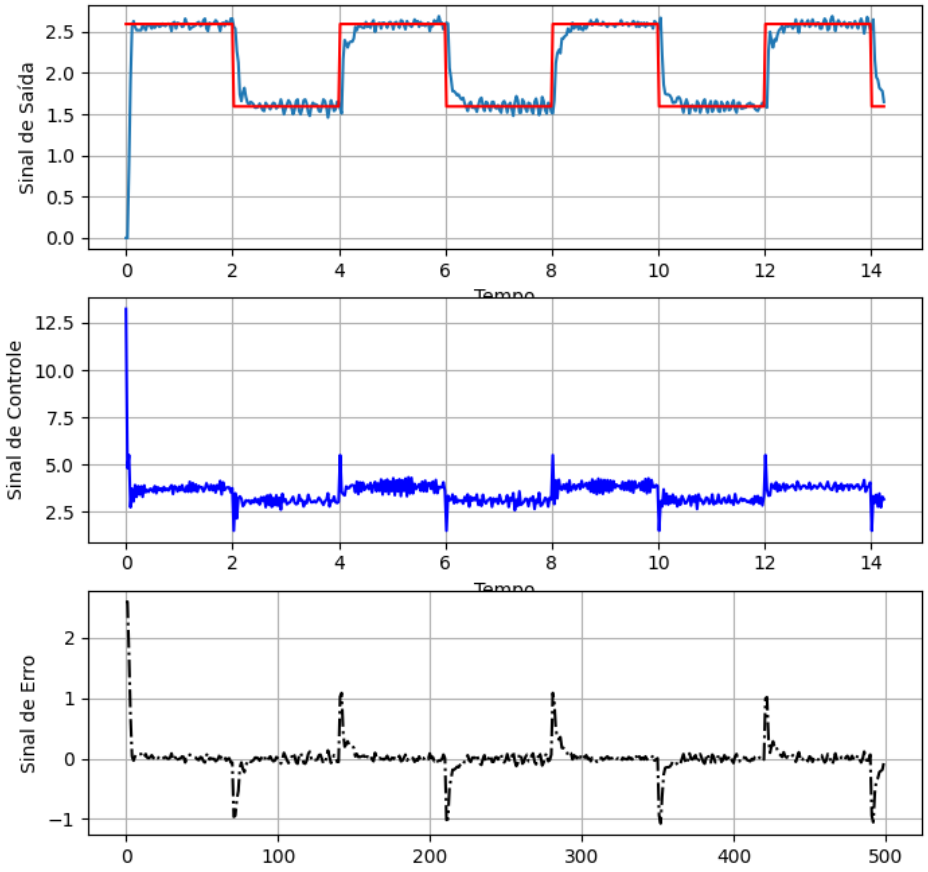
Essa equação é comumente encontrada em sistemas de controle discretos, especialmente quando se utiliza um controlador PID discreto de ordem superior. O termo $u[n]$ representa o sinal de controle calculado para o instante de tempo discreto n, considerando os valores anteriores de u e erro. O objetivo é ajustar o sinal de controle para minimizar o erro entre a saída desejada e a saída real do sistema.

Os resultados obtidos com o controle PID foram altamente satisfatórios. A inclusão da ação derivativa proporcionou uma resposta ainda mais precisa e ajustada às variações detectadas no sistema motor-gerador. O controlador PID demonstrou sua capacidade de regular a dinâmica da bancada de forma estável e responsiva, evidenciada pelos ajustes contínuos na entrada do sistema em resposta aos sinais de controle do Python.

Os ganhos do controlador PID, otimizados por meio de AG, desempenharam um papel crucial nos resultados alcançados. A adaptação desses parâmetros à dinâmica específica da bancada motor-gerador permitiu um controle eficaz em diferentes pontos de operação.

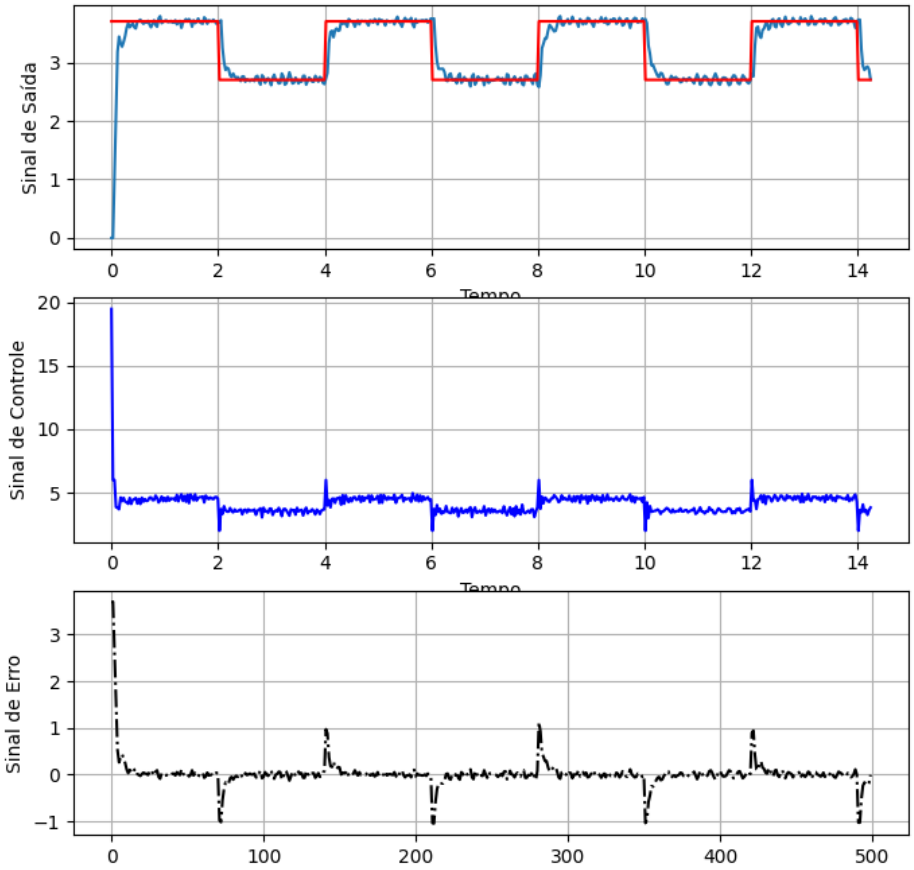
A precisão do controle PID, aliada à comunicação confiável entre Python e Arduino, representa um avanço significativo neste estudo. Os resultados observados nas figuras 36, 37 e 38, não apenas confirmam a eficácia do controle em tempo real, mas também destacam a importância da integração entre linguagens de programação para a implementação bem-sucedida de sistemas de controle dinâmico.

Figura 36 – Resultado do sistema em resposta aos sinais de controle PID enviados pelo Python para o ponto de operação 3.5V.



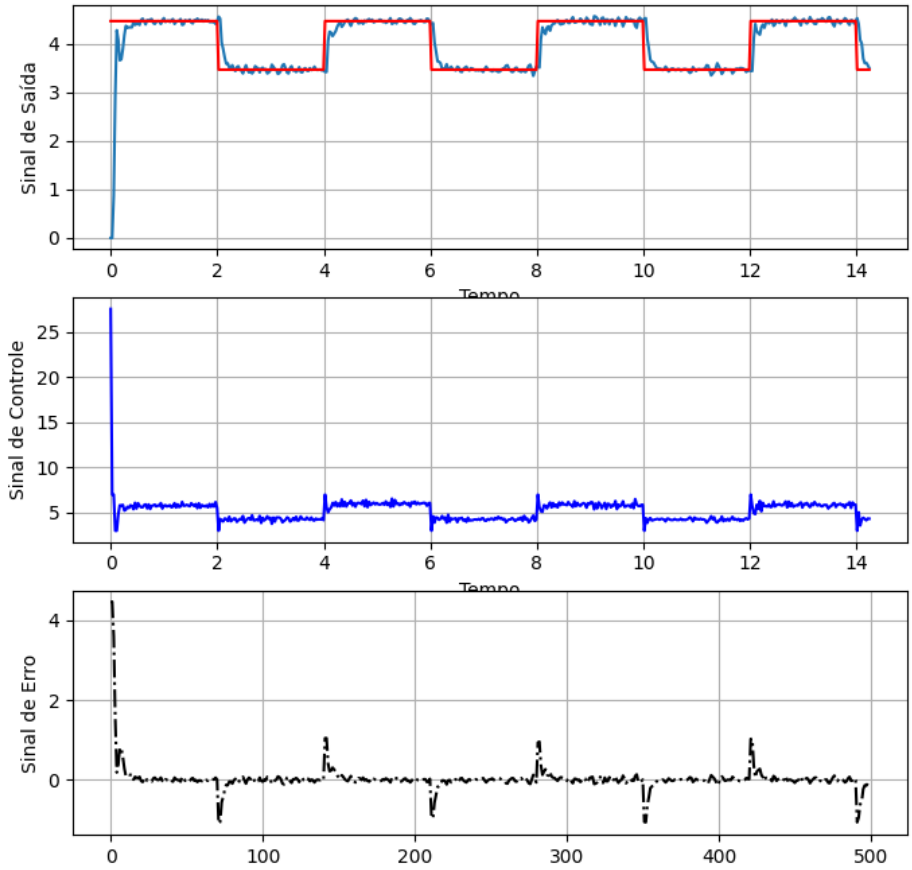
Fonte: O Autor

Figura 37 – Resultado do sistema em resposta aos sinais de controle PID enviados pelo Python para o ponto de operação 4V.



Fonte: O Autor

Figura 38 – Resultado do sistema em resposta aos sinais de controle PID enviados pelo Python para o ponto de operação 5V.



Fonte: O Autor

Os códigos completos e detalhes adicionais dos experimentos estão disponíveis no Apêndice **??** e **??**.

4.7 Análise de Transferibilidade dos Controladores em Três Pontos de Operação

O cálculo do Erro Médio Quadrático **MSE** (Mean Squared Error) desempenha um papel importante na avaliação da eficácia de estratégias de controle, fornecendo uma medida quantitativa da discrepância entre os valores preditos e observados. Ao aplicar o **MSE** a três pontos de operação distintos em uma bancada motor-gerador, podemos obter insights valiosos sobre a precisão e a robustez do sistema de controle em diferentes condições.

Para cada ponto de operação, representado pelos conjuntos de observações $\{erro_{1i}\}$, $\{erro_{2i}\}$ e $\{erro_{3i}\}$, onde i varia de 1 a N (o número total de observações).

A equação que calcula o **MSE** é representada pela fórmula:

$$E_n = \frac{1}{N}\sum_{i=1}^{N}(erro_i)^2 \tag{4.3}$$

Na expressão acima: - E_n é o erro médio quadrático, - N é o número total de observações ou amostras, - $erro_i$ é o erro associado à i-ésima observação.

A análise do **MSE** para os três pontos de operação oferece uma visão abrangente da capacidade do sistema de controle em lidar com diferentes regimes operacionais. Uma baixa pontuação de **MSE** indica uma correspondência próxima entre os valores previstos e os observados, sugerindo que o controlador está apto a manter o sistema próximo ao comportamento desejado em cada ponto de operação.

Por outro lado, um **MSE** mais elevado pode indicar áreas de melhoria na estratégia de controle para otimizar o desempenho do sistema em condições específicas. Essas descobertas podem orientar ajustes nos parâmetros do controlador ou até mesmo sugerir a necessidade de estratégias de controle alternativas em determinados pontos operacionais.

Ao considerar o **MSE** para os três pontos de operação, este trabalho proporciona uma avaliação abrangente da eficiência das estratégias de controle em diferentes cenários, permitindo ajustes precisos e aprimoramentos específicos em busca de um controle mais eficaz e adaptativo em uma variedade de condições operacionais.

A tabela 3 apresenta uma comparação dos erros associados aos controladores PID ajustados para os pontos de operação de 3.5V, 4V e 5V. Esses valores de erro refletem a diferença entre a saída desejada e a saída real do sistema para cada ponto de operação.

Tabela 3 – Comparação de Erros Associados aos Controladores PID Ajustados para os Pontos de Operação de 3.5V, 4V e 5V.

	3.5V	**4V**	**5V**
Controlador PID (3.5V)	0.0725	0.0722	0.2500
Controlador PID (4V)	0.1352	0.1114	0.1362
Controlador PID (5V)	0.4855	1.1747	0.1526

Fonte: O Autor

Ao examinar os erros associados aos diferentes controladores em três pontos de operação distintos (3.5V, 4V e 5V), observam-se padrões interessantes.

Ao considerar o controlador ajustado para operar a 3.5V, observa-se que, ao aplicá-lo aos pontos de 4V e 5V, os erros associados foram de 0.0722 e 0.25, respectivamente. Esses resultados indicam uma relativa estabilidade ao aplicar o controlador otimizado para 3.5V em outros pontos de operação, embora haja um aumento notável no erro quando usado no ponto de 5V.

Similarmente, ao utilizar o controlador otimizado para 4V nos pontos de 3.5V e 5V, os erros foram de 0.1352 e 0.1362, respectivamente. Essa consistência sugere uma certa

transferibilidade das configurações do controlador entre os pontos de 3.5V e 4V, com variações mínimas no desempenho quando aplicado ao ponto de 5V.

No entanto, ao aplicar o controlador sintonizado para operar a 5V nos outros pontos de operação, os erros resultantes foram de 0.4855 e 1.1747 para 3.5V e 4V, respectivamente. Isso revela uma maior sensibilidade e limitação na utilização desse controlador em pontos de operação diferentes daquele para o qual foi originalmente otimizado.

Esses resultados indicam que, embora haja alguma transferibilidade entre os controladores otimizados para diferentes pontos de operação, a aplicabilidade em pontos distintos pode variar consideravelmente. É importante considerar cuidadosamente a adaptabilidade e a limitação de cada controlador ao ser aplicado em regimes de operação diferentes.

CAPÍTULO

5

DISCUSSÃO DOS RESULTADOS

Nesta seção, iniciaremos a discussão dos resultados obtidos por meio da implementação dos controladores PI e PID no sistema dinâmico em questão. Os dados e análises apresentados têm o objetivo de fornecer uma compreensão aprofundada do desempenho dos controladores e suas capacidades de regulação diante das condições dinâmicas do sistema motor-gerador.

5.1 Análise Crítica do Uso do `PySINDy` e Placa Arduino

O uso do Modelo `SINDy` com a biblioteca `PySINDy` em conjunto com a placa Arduino pode ser realizada considerando diferentes aspectos, incluindo vantagens, desafios e considerações práticas. Vamos explorar esses pontos:

5.1.1 Vantagens:

1. **Identificação Eficiente de Dinâmicas Não Lineares:** O `PySINDy` é uma biblioteca eficaz para identificação de dinâmicas não lineares em sistemas. Ele utiliza métodos de regressão esparsa para inferir as equações diferenciais subjacentes a partir de dados.

2. **Simplicidade de Implementação:** O `PySINDy` oferece uma interface relativamente simples para implementar o processo de identificação do modelo. Isso facilita a aplicação mesmo para usuários com menos experiência em identificação de sistemas.

3. **Integração com Arduino:** A utilização da placa Arduino permite a aquisição de dados em tempo real, possibilitando a identificação do modelo a partir de dados experimentais. A comunicação bidirecional com o Arduino oferece uma abordagem prática para a implementação em sistemas físicos.

5.1.2 Desafios:

1. **Complexidade dos Sistemas:** A eficácia do `SINDy` depende da qualidade e quantidade dos dados disponíveis. Em sistemas muito complexos ou com dados ruidosos, a identificação pode se tornar desafiadora.

2. **Ajuste de Parâmetros:** A configuração de parâmetros no `PySINDy` pode exigir experimentação para alcançar resultados ideais. A escolha apropriada de hiperparâmetros e métodos de regressão pode impactar significativamente os resultados.

3. **Limitações da Placa Arduino:** Embora a placa Arduino seja uma solução prática para a aquisição de dados, ela pode ter limitações em termos de taxa de amostragem e precisão, dependendo do modelo específico.

5.1.3 Considerações Práticas:

1. **Validação Experimental:** A validação do modelo identificado pelo `PySINDy` com dados experimentais da placa Arduino é crucial. Garantir que o modelo seja capaz de prever o comportamento do sistema em condições não utilizadas durante a identificação é essencial.

2. **Avaliação da Generalização:** Verificar a capacidade do modelo identificado de generalizar para diferentes condições operacionais é importante. Isso ajuda a garantir que o modelo seja robusto e aplicável em uma variedade de cenários.

3. **Documentação e Suporte:** A documentação da biblioteca `PySINDy` e da placa Arduino, juntamente com a disponibilidade de comunidades de suporte online, pode impactar positivamente a experiência do usuário.

A combinação da biblioteca `PySINDy` com a placa Arduino oferece uma abordagem interessante para identificação de sistemas dinâmicos lineares. No entanto, é crucial considerar os desafios inerentes à identificação de sistemas complexos e garantir uma configuração adequada para obter resultados confiáveis. A prática de validar experimentalmente o modelo identificado é fundamental para garantir sua aplicabilidade prática em contextos do mundo real.

5.2 Análise Crítica dos Controladores PI e PID

O desempenho dos controladores PI e PID revela aspectos distintos em relação à eficácia de cada estratégia de controle na regulação do sistema dinâmico. Vamos abordar algumas considerações para ambos:

5.2.1 Controlador PI:

1.1 **Estabilidade e Resposta ao Erro:**

- O controlador PI apresenta uma resposta suave ao erro, utilizando a ação proporcional e integral.
- A integração ajuda a eliminar o erro em estado estacionário, proporcionando estabilidade ao sistema.

1.2 **Simplicidade e Implementação:**

- A simplicidade do controlador PI facilita a implementação e ajuste dos parâmetros.
- Contudo, pode ter limitações em sistemas mais complexos que exigem uma resposta mais rápida a mudanças.

5.2.2 Controlador PID:

2.1 **Resposta Rápida a Variações:**

- O controlador PID, ao incorporar a ação derivativa, responde mais rapidamente a variações no sistema.
- Isso é crucial para situações em que a resposta rápida é essencial para a estabilidade.

2.2 **Adaptabilidade Dinâmica:**

- A capacidade do PID de ajustar dinamicamente os parâmetros permite uma adaptação mais eficiente a diferentes condições do sistema.
- Essa característica é valiosa em ambientes dinâmicos e sujeitos a mudanças.

2.3 **Complexidade de Sintonia:**

- A sintonia dos parâmetros PID pode ser mais complexa devido à presença do termo derivativo.
- Um ajuste inadequado pode levar a oscilações ou respostas instáveis.

5.2.3 Comparação Geral:

3.1 **Especificidade do Sistema:**

- A escolha entre PI e PID depende da natureza específica do sistema e dos requisitos de desempenho.
- Para sistemas mais simples ou estáveis, um controlador PI pode ser suficiente. Ambientes mais dinâmicos podem exigir um PID para melhor adaptabilidade.

3.2 **Otimização com `PyGad`:**

- A otimização dos parâmetros com a biblioteca `PyGad` é uma abordagem inovadora e eficaz para ambos os controladores.
- `PyGad` permite uma sintonia automática que pode levar a resultados mais eficientes em comparação com métodos manuais.

3.3 **Feedback Contínuo:**

- Ambos os controladores se beneficiaram da comunicação contínua com o Arduino, proporcionando feedback em tempo real.
- Isso contribuiu para a estabilidade e eficácia geral do controle.

3.4 **Trade-off entre Desempenho e Complexidade:**

- A escolha entre PI e PID envolve um trade-off entre desempenho e complexidade. A adição do termo derivativo no PID pode trazer benefícios significativos em termos de resposta rápida, mas também aumenta a complexidade de sintonia.

A análise crítica sugere que ambos os controladores têm seus méritos e são aplicáveis em diferentes contextos. A escolha entre PI e PID deve considerar a natureza específica do sistema, os requisitos de desempenho e a facilidade de implementação. O uso da biblioteca `PyGad` para otimização destaca-se como uma abordagem valiosa em ambas as estratégias de controle.

5.3 Análise Crítica da Sintonia em Três Pontos de Operação

A utilização de três pontos de operação na sintonia de controladores é uma estratégia comum para sistemas dinâmicos, proporcionando ajustes específicos para diferentes regimes de operação. No entanto, é importante destacar algumas considerações críticas sobre essa prática:

1. **Especificidade dos Pontos de Operação:**
   - **Vantagem:** A sintonia em três pontos permite adaptar os controladores a diferentes condições operacionais, considerando variações nas características do sistema.
   - **Desvantagem:** Os ganhos otimizados para um ponto específico podem não ser diretamente aplicáveis a outros pontos de operação. Isso ocorre porque as dinâmicas do sistema podem variar significativamente em diferentes condições, exigindo ajustes específicos.
2. **Linearidade e Invariância do Sistema:**
   - **Vantagem:** Se o sistema for linear e invariante no tempo, os ganhos sintonizados em diferentes pontos podem ser aplicados com mais confiança.

- **Desvantagem:** Em sistemas não lineares ou variantes no tempo, os ganhos otimizados podem não garantir um desempenho ideal em pontos fora daqueles utilizados na sintonia.

3. **Complexidade da Implementação:**

   - **Vantagem:** A abordagem de múltiplos pontos de operação permite maior flexibilidade no ajuste dos controladores.
   - **Desvantagem:** A implementação pode se tornar complexa, especialmente se houver muitos pontos de operação a serem considerados. A seleção adequada desses pontos é crucial.

4. **Necessidade de Retrabalho:**

   - **Vantagem:** Ajustar os ganhos em pontos específicos pode melhorar significativamente o desempenho do controlador nessas condições.
   - **Desvantagem:** Mudanças nas condições operacionais podem exigir retrabalho na sintonia, tornando o sistema sensível a variações.

5. **Generalização para Outros Sistemas:**

   - **Vantagem:** Em alguns casos, a sintonia em múltiplos pontos pode levar a um controlador mais robusto.
   - **Desvantagem:** Os ganhos otimizados podem não ser facilmente generalizáveis para sistemas diferentes, especialmente se suas dinâmicas forem substancialmente distintas.

A aplicação de três pontos de operação na sintonia de controladores é uma estratégia eficaz quando utilizada com cuidado e compreensão das características específicas do sistema em questão. A escolha entre essa abordagem e métodos mais simplificados depende da complexidade do sistema, da variabilidade nas condições operacionais e dos requisitos de desempenho desejados.

Em resumo, os resultados obtidos até o momento indicam uma promissora eficácia dos controladores PI e PID implementados. A flexibilidade proporcionada pela discretização de controladores, aliada à capacidade de otimização automática, destaca a robustez da abordagem adotada. A análise dos conjuntos de ganhos e a sensibilidade aos diferentes pontos de operação oferecem uma compreensão valiosa para ajustes e futuras melhorias.

CAPÍTULO

6

CONCLUSÃO

6.1 Principais Conclusões

O desenvolvimento e implementação da estratégia de controle para a bancada motor-gerador representam um marco significativo nesta pesquisa. A combinação de técnicas avançadas, como controle proporcional integral (PI) e proporcional integral derivativo (PID) otimizados por Algoritmos Genéticos (AG), aliada à utilização de modelos SINDy, demonstrou ser uma abordagem eficaz para atender aos objetivos propostos.

Ao implementar um código para a coleta de dados da bancada, garantimos uma base sólida para análise e modelagem do sistema. A utilização da técnica SINDy para construção de modelos proporcionou uma representação precisa do comportamento dinâmico, permitindo uma compreensão profunda das interações no sistema motor-gerador.

A aplicação de Algoritmos Genéticos (AG) para otimização dos parâmetros dos controladores PI e PID foi essencial para alcançar um desempenho otimizado. Essa abordagem offline permitiu uma sintonia precisa, melhorando a eficiência do controle e a resposta do sistema.

Além disso, a implementação de um código online para a transmissão dos controladores otimizados pela AG para a bancada motor-gerador possibilitou um controle em tempo real, contribuindo para a eficiência e estabilidade do sistema.

Dessa forma, a pesquisa atingiu seus objetivos ao propor uma estratégia de controle eficiente, integrando múltiplas técnicas para garantir o desempenho ideal da bancada motor-gerador. As conquistas alcançadas neste estudo não apenas avançam o conhecimento na área de controle de sistemas dinâmicos, mas também abrem portas para aplicações práticas em diversos cenários industriais.

6.2 Sugestões para Trabalhos Futuros

Considerando os resultados obtidos e as nuances identificadas durante esta pesquisa, algumas sugestões para trabalhos futuros incluem:

1. Realizar análises estatísticas mais detalhadas, como testes de significância e intervalos de confiança, para validar a robustez e a confiabilidade dos resultados obtidos. Isso poderia proporcionar uma compreensão mais aprofundada da variabilidade do sistema e da eficácia dos controladores em diferentes condições.

2. Conduzir uma análise comparativa entre os resultados experimentais e modelos teóricos conhecidos. Essa comparação poderia fornecer insights sobre a fidelidade do modelo utilizado na simulação em relação ao comportamento real do sistema, contribuindo para uma validação mais sólida.

3. Investigar estratégias de controle mais avançadas e específicas para sistemas motor-gerador, como controle adaptativo, controle preditivo e redes neurais. Essas abordagens podem oferecer melhor desempenho em condições operacionais desafiadoras e complexas.

4. Considerar a integração de sensores adicionais para capturar mais informações sobre o sistema, como temperatura, corrente e velocidade. Esses dados podem enriquecer a análise do comportamento do sistema e fornecer entradas mais precisas para os controladores.

5. Estender a pesquisa para experimentação com outros dispositivos além do Arduino, explorando a aplicabilidade da abordagem em diferentes plataformas e dispositivos embarcados.

REFERÊNCIAS

ANDRÉ, L. P. *Algoritmos Genéticos*. 2023. Disponível em: <https://sites.icmc.usp.br/andre/research/genetic/>. 13

ASTROM, K. J.; HAGGLUND, T. *Advanced PID Control*. [S.l.]: ISA - The Instrumentation, Systems, and Automation Society, 2008. 7

ASTROM, K. J.; MURRAY, R. M. *Feedback Systems: An Introduction for Scientists and Engineers*. [S.l.]: Princeton University Press, 2010. 7

BACK, T.; FOGEL, D. B.; MICHALEWICZ, Z. *Evolutionary Computation 1: Basic Algorithms and Operators*. [S.l.]: Institute of Physics Publishing, 1997. 4

BRUNTON, S. L.; KUTZ, J. N. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. [S.l.]: Cambridge University Press, 2019. 4, 7, 11, 32, 39

BRYSON, A. E.; HO, Y.-C. *Applied Optimal Control: Optimization, Estimation, and Control*. [S.l.]: CRC Press, 1999. 2

CALDEIRA, P. H. E. Controle de velocidade de um motor de corrente contínua utilizando o método vrft. 2020. 2

CAMACHO, E. F.; BORDONS, C. *Model Predictive Control*. [S.l.]: Springer, 2004. 8

DAVISON, E.; WANG, S. On pole assignment in linear multivariable systems using output feedback. *IEEE Transactions on Automatic Control*, IEEE, v. 20, n. 4, p. 516–518, 1975. 6

DEB, K. *Multi-Objective Optimization Using Evolutionary Algorithms*. [S.l.]: John Wiley & Sons, 2001. 9, 10, 11

DORF, R. C.; BISHOP, R. H. *Modern Control Systems*. [S.l.]: Pearson, 2016. 1

EIBEN, A. E.; SMITH, J. E. *Introduction to Evolutionary Computing*. [S.l.]: Springer, 2015. 10

FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. *Feedback control of dynamic systems (6th ed.)*. Boston, MA: Pearson Education, 2010. 6

_____. *Feedback Control of Dynamic Systems*. [S.l.]: Pearson, 2015. 1

FRIEDMAN, J. *Greedy Function Approximation: A Gradient Boosting Machine*. [S.l.]: Annals of Statistics, 2001. 9

GAD, A. F. Pygad: An intuitive genetic algorithm python library. *arXiv preprint arXiv:2106.06158*, 2021. 21, 38

GANI, M. M.; ISLAM, M. S.; ULLAH, M. A. Optimal pid tuning for controlling the temperature of electric furnace by genetic algorithm. *SN Applied Sciences*, Springer, v. 1, p. 1–8, 2019. 18

GEEKSFORGEEKS. *Crossover in Genetic Algorithm*. 2023. Disponível em: <https://www.geeksforgeeks.org/crossover-in-genetic-algorithm/>. 14, 15, 16

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. [S.l.]: Addison-Wesley, 2010. 3, 10, 13, 17

GOODWIN, G. C. *Control System Desig*. [S.l.]: Control System Design, 2001. 8

HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: MIT press, 1992. 3, 10, 11, 16

HOU, Z.-S.; WANG, Z. From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, Elsevier, v. 235, p. 3–35, 2013. 2

IZIDORO, S. C.; MELO-MINARDI, R. C. de; PAPPA, G. L. Gass: identifying enzyme active sites with genetic algorithms. *Bioinformatics*, v. 31, n. 6, p. 864–870, 2014. 11

JAYACHITRA, A.; VINODHA, R. Genetic algorithm based pid controller tuning approach for continuous stirred tank reactor. *Advances in Artificial Intelligence*, Hindawi Limited London, UK, United Kingdom, v. 2014, p. 9–9, 2015. 18

KIRK, D. E. *Optimal control theory: an introduction*. [S.l.]: Courier Corporation, 2004. 2

KLUEVER, C. A. *Dynamic systems: modeling, simulation, and control*. [S.l.]: John Wiley & Sons, 2020. 6

MCROBERTS, M. *Arduino básico*. [S.l.]: Novatec Editora, 2018. 4

MITCHELL, M. *An introduction to genetic algorithms*. [S.l.]: MIT press, 1998. 4, 10, 12, 14, 15, 18

OGATA, K. *Engenharia de Controle Moderno*. 5th. ed. [S.l.]: Pearson Prentice Hall, 2010. 1, 6

SOUZA, K. C. d. Trabalho de Conclusão de Curso, *Identificação não-linear no espaço de estados por regressão esparsa de bancada motor-gerador*. Tucuruí: [s.n.], 2023. 53 p. <https://bdm.ufpa.br:8443/jspui/handle/prefix/5375>. Orientador: Raphael Barros Teixeira. Disponível em: <https://bdm.ufpa.br:8443/jspui/handle/prefix/5375>. 27, 28

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. [S.l.]: MIT Press, 2018. 3