



BRENO CAUÊ SATURNINO CARLOS E CASAIS

**SISTEMA DE CONTROLE PARA OTIMIZAÇÃO DE
LUMINOSIDADE EM PLANTA REDUZIDA PARA GALINHAS
POEDEIRAS**

LAVRAS – MG

2023

BRENO CAUÊ SATURNINO CARLOS E CASAIS

**SISTEMA DE CONTROLE PARA OTIMIZAÇÃO DE LUMINOSIDADE EM PLANTA
REDUZIDA PARA GALINHAS POEDEIRAS**

Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia da Universidade Federal
de Lavras, como parte das exigências para
graduação nível Bacharelado em Engenharia de
Controle e Automação.

Prof. DSc. Leonardo Silveira Paiva
Orientador

LAVRAS – MG

2023

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca
Universitária da UFLA, com dados informados pelo próprio autor.**

Casais, Breno Cauê Saturnino

Sistema de Controle para Otimização de Luminosidade
em Planta Reduzida para Galinhas Poedeiras / Breno Cauê
Saturnino Carlos e Casais. – Lavras : UFLA, 2023.

63 p. : il.

TCC(graduação)–Universidade Federal de Lavras, 2023.

Orientador: Prof. DSc. Leonardo Silveira Paiva.

Bibliografia.

1. TCC. 2. Controle PI. 3. Funções de Otimização. I.
Silveira Leonardo. II. Título

CDD-808.066

BRENO CAUÊ SATURNINO CARLOS E CASAIS

**SISTEMA DE CONTROLE PARA OTIMIZAÇÃO DE LUMINOSIDADE EM PLANTA
REDUZIDA PARA GALINHAS POEDEIRAS**

Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia da Universidade Federal
de Lavras, como parte das exigências para
graduação nível Bacharelado em Engenharia de
Controle e Automação.

APROVADA em 15 de Dezembro de 2023.

Dr. Leonardo Silveira Paiva	UFLA
Dr. Daniel Augusto Pereira	UFLA
Ms. Antonio Mendes Magalhães Júnior	UFLA

Prof. DSc. Leonardo Silveira Paiva
Orientador

**LAVRAS – MG
2023**

RESUMO

No âmbito da automação avícola, a implementação de sistemas de controle para a regulação da intensidade luminosa desempenha um papel crucial para promover seu bem-estar e otimizar a produção de ovos. Este trabalho apresenta um estudo aprofundado sobre o desenvolvimento e aplicação de um sistema de controle, baseado em um controlador Proporcional Integral (PI). O processo de desenvolvimento envolveu a coleta de dados experimentais e a aplicação de estratégias de controle adaptativas para otimizar o desempenho do sistema. Os parâmetros iniciais do PI, derivados do método de Ziegler-Nichols, foram utilizados como sementes para as funções de otimização, proporcionando ajustes mais precisos. Os métodos ITA, ITAE e MSE foram empregados para refinar os parâmetros, considerando diferentes critérios de desempenho. Este trabalho não só alcançou com sucesso os objetivos de implementação e otimização do sistema de controle de iluminação, como, também introduziu novas perspectivas das estratégias adaptativas de controle. A aplicação bem-sucedida das funções de otimização proporcionou uma abordagem robusta e eficiente para ajustar os parâmetros do controlador, resultando em melhorias notáveis no desempenho do sistema. Essa contribuição não apenas impacta positivamente o bem-estar das galinhas poedeiras, mas, também sinaliza avanços na eficiência e na responsabilidade ambiental da produção avícola.

Palavras-chave: Controle de luminosidade, Funções de Otimização, Ambiente Avícola.

ABSTRACT

In the context of poultry automation, the implementation of control systems for regulating light intensity plays a crucial role in promoting their well-being and optimizing egg production. This paper presents an in-depth study on the development and application of a control system based on a Proportional Integral (PI) controller. The development process involved the collection of experimental data and the application of adaptive control strategies to optimize system performance. The initial PI parameters, derived from the Ziegler-Nichols method, were used as seeds for optimization functions, providing more precise adjustments. The ITA, ITAE, and MSE methods were employed to refine the parameters, considering different performance criteria. This work not only successfully achieved the goals of implementing and optimizing the lighting control system but also introduced new perspectives on adaptive control strategies. The successful application of optimization functions provided a robust and efficient approach to adjusting controller parameters, resulting in noteworthy improvements in system performance. This contribution not only positively impacts the well-being of laying hens but also signals advances in the efficiency and environmental responsibility of poultry production.

Keywords: Lighting Control, Optimization Functions, Poultry Environment

LISTA DE FIGURAS

Figura 3.1 – Curva de resposta em forma de S	20
Figura 3.2 – Resposta ao degrau unitário de uma planta	21
Figura 3.3 – Resposta ao degrau de sistemas sobreamortecido.	25
Figura 3.4 – Sintaxe para função fmincon	30
Figura 4.1 – Celeiro parte externa	31
Figura 4.2 – Celeiro parte interna	32
Figura 4.3 – MOSFET IRF520	33
Figura 4.4 – Sensor LDR	33
Figura 4.5 – Arduino Uno	34
Figura 4.6 – Fonte chaveada 12v	35
Figura 4.7 – Lâmpada 12V 60/55W	36
Figura 4.8 – Esquemático das ligações	37
Figura 4.9 – Celeiro com equipamentos montados vista frontal	38
Figura 4.10 – Celeiro com equipamentos montados vista superior	38
Figura 4.11 – Malha aberta do sistema de luminosidade.	39
Figura 4.12 – Tabela com os valores encontrados para porcentagem de luminosidade em relação ao Duty Cycle de 33%, 66% e 99%	39
Figura 4.13 – Gráfico da intensidade luminosa em relação ao Duty Cycle de 33%	40
Figura 4.14 – Gráfico da intensidade luminosa em relação ao Duty Cycle de 66%	40
Figura 4.15 – Gráfico da intensidade luminosa em relação ao Duty Cycle de 99%	41
Figura 4.16 – Resposta do sistema ao degrau de 33% de Duty Cycle juntamente com o modelo obtido.	42
Figura 4.17 – Resposta do sistema ao degrau de 66% de Duty Cycle juntamente com o modelo obtido.	43
Figura 4.18 – Resposta do sistema ao degrau de 99% de Duty Cycle juntamente com o modelo obtido.	43
Figura 4.19 – Diagrama de Blocos do sistema de controle	45

Figura 4.20 – Resposta do sistema com parâmetros de Ziegler-Nichols	46
Figura 4.21 – Diagrama de blocos para funções de otimização	46
Figura 4.22 – Código principal para executar a função fminsearch	47
Figura 4.23 – Resposta ao degrau unitário do sistemas	48
Figura 4.24 – Diagrama de blocos com perturbação	49
Figura 4.25 – Resposta ao degrau do sistema aplicado uma perturbação	50
Figura 4.26 – Resposta do sistema para os parâmetros de Ziegler-Nichols	53
Figura 4.27 – Resposta do sistema para os parâmetros de MSE	54

LISTA DE TABELAS

Tabela 3.1 – Regra de sintonia de Ziegler-Nichols baseada na resposta ao degrau.	21
Tabela 4.1 – Resultados para as funções de otimização	48
Tabela 4.2 – Resultados para as funções de otimização com perturbação	50

SUMÁRIO

1	INTRODUÇÃO	9
2	Objetivo Geral	11
2.1	Objetivos específicos	11
3	Revisão Bibliográfica	12
3.1	Contexto histórico das galinhas poedeiras	12
3.2	Importância das galinhas poedeiras na economia brasileira	13
3.3	Manejo da criação avícola	13
3.4	Manejo de luminosidade	14
3.4.1	Fotoperíodo (duração da iluminação)	15
3.4.2	Intensidade da luz (lux)	15
3.4.3	Comprimento de onda (Cor)	16
3.5	Taxa de iluminação no Brasil	17
3.6	Sistema de controle	17
3.7	Tipos de controladores	18
3.8	Métodos de sintonia de malha	20
3.8.1	Método de Ziegler-Nichols	20
3.9	Modelagem matemática	22
3.9.1	Identificação de sistemas	22
3.9.2	Função de transferência	22
3.10	Métodos determinísticos	23
3.10.1	Método Sunderasen	24
3.11	Funções de otimização	26
3.12	Sintonia com a função <i>fmincon</i> de otimização do MATLAB	28
4	Metodologia	31
4.1	Estrutura física	31
4.1.1	Módulo IRF520	32
4.1.2	Sensor LDR	33

4.1.3	Microcontrolador Arduino	34
4.1.4	Fonte de alimentação 12V	35
4.1.5	Lâmpada	35
4.1.6	Protoboard	36
4.2	Esquemático	36
4.3	Modelagem do sistema	37
4.4	Controle	44
4.5	Funções de otimização	45
4.5.1	Aplicação de uma perturbação no sistema	49
4.6	Discretização da função	51
4.7	Análise e resultados	53
5	CONCLUSÃO	56
	REFERÊNCIAS	59
	APENDICE A – Código de aquisição de Dados	61
	APENDICE B – Código do Controle de Luminosidade	62
	APENDICE C – Código das funções de otimizações	63

1 INTRODUÇÃO

No cenário da produção avícola, a criação de galinhas poedeiras desempenha um papel essencial na garantia do abastecimento de ovos de alta qualidade para a população. A eficiência e o bem-estar dessas aves são aspectos de extrema importância para a produtividade e a qualidade dos produtos avícolas.

Nesse contexto, surge a necessidade de aprimorar as condições de criação dessas aves, otimizando o ambiente em que estão inseridas. Este trabalho se concentra na otimização do controle de luminosidade em instalações de produção avícola, com ênfase na criação de galinhas poedeiras. A busca por melhores condições de criação para essas aves é fundamental tanto para a qualidade dos produtos avícolas quanto para o bem-estar dos animais.

No decorrer deste estudo, uma abordagem abrangente é adotada, compreendendo várias etapas interligadas. Inicialmente, o Método de Sunderase é empregado para modelar o sistema físico de luminosidade. Tal modelagem é fundamental para uma compreensão detalhada do sistema e serve como base para o desenvolvimento das estratégias de controle.

A etapa subsequente consiste na implementação de um controlador Proporcional e Integral (PI) para regular a intensidade luminosa. O método de Ziegler-Nichols foi utilizado para realizar uma sintonia inicial dos parâmetros do controlador. Logo em seguida, foram aplicadas funções de otimização como IAE, ITAE e MSE, garantindo um controle preciso e responsivo da iluminação do ambiente de criação das aves. A sintonia adequada do PI é essencial para atingir os objetivos de maximizar a produção de ovos e o bem-estar das aves.

O microcontrolador Arduino foi utilizado como plataforma central para o controle preciso da intensidade luminosa em tempo real. Além disso, a ferramenta MATLAB foi adotada para a implementação, análise e simulação dos modelos e algoritmos, proporcionando um ambiente robusto para a avaliação do desempenho do sistema proposto.

Para avaliar a eficácia das estratégias propostas, este trabalho conduziu simulações detalhadas em uma planta reduzida que replica as condições reais de criação das galinhas poedeiras. Estas simulações têm como objetivo principal analisar o impacto das estratégias de otimização no desempenho das aves e no controle de luminosidade.

Ao enfatizar o controle de iluminação e a otimização dos parâmetros do controlador, este estudo representa uma contribuição relevante para aprimorar a produção de galinhas poedeiras, alinhando-a com as melhores práticas de bem-estar animal e eficiência produtiva.

2 OBJETIVO GERAL

O principal objetivo deste trabalho foi desenvolver um sistema de controle de luminosidade em uma planta reduzida, visando simular de como seria a implementação em um sistema real em instalações de produção avícola, com ênfase na criação de galinhas poedeiras.

2.1 Objetivos específicos

Como objetivos específicos se tem:

- criação e estruturação de uma planta reduzida de um galpão para galinhas;
- montagem do circuito elétrico que foi utilizado na planta reduzida;
- encontrar funções de transferências pelo método de Sunderase para cada faixas de valores capturados;
- validar os modelos encontrados com ênfase em buscar o que melhor representa os dados coletados;
- propor e implantar um controlador Proporcional Integral (PI) para regular a intensidade luminosa;
- incorpora funções de otimização para melhorar o desempenho do sistema de controle de luminosidade;
- avaliar a eficácia das estratégias propostas.

3 REVISÃO BIBLIOGRÁFICA

3.1 Contexto histórico das galinhas poedeiras

A criação de galinhas poedeiras é uma atividade que remonta à antiguidade, com evidências arqueológicas que datam de cerca de 2.300 anos atrás (ANAND; RAJAGOPALAN; MOHAN, 2012). No entanto, segundo os mesmos autores, a produção comercial de ovos só começou a se desenvolver no final do século XIX e início do século XX, com a introdução de raças de galinhas especializadas na postura de ovos.

Inicialmente, a criação das galinhas era realizada em pequena escala, principalmente por agricultores familiares, de acordo com Nascimento (2017). Com o passar do tempo, a produção comercial de ovos cresceu em importância, tornando-se uma atividade econômica relevante em muitos países.

As primeiras galinhas comerciais foram criadas nos Estados Unidos e na Inglaterra Nascimento (2017). Essas aves foram criadas para produzir ovos de forma eficiente e consistente, o que permitiu a criação de uma indústria de produção de ovos em larga escala.

Ao longo dos séculos, diferentes raças e variedades de aves foram desenvolvidas em várias partes do mundo para atender às demandas específicas por ovos. A seleção genética desempenhou um papel importante nesse processo, permitindo a criação de linhagens especializadas com maior eficiência de postura, qualidade de ovos e adaptação a diferentes condições ambientais.

Atualmente, conforme afirmam Soares e Goulart (2018), a criação de galinhas poedeiras é realizada em todo o mundo, com diferentes raças e sistemas de produção, desde criações em pequena escala até grandes instalações industriais.

A criação de aves para a produção comercial de ovos teve um aumento significativo no século XX, impulsionada pelo crescimento populacional e pela demanda crescente por ovos em todo o mundo. Novas técnicas de manejo e melhorias na nutrição foram desenvolvidas para maximizar a produção de ovos e garantir a saúde e o bem-estar das aves.

Ao longo das décadas, as práticas de manejo e as instalações de criação foram aprimoradas, visando melhorar a eficiência produtiva e o bem-estar das próprias. Isso incluiu a introdução de

sistemas de criação em gaiolas, que permitiam uma maior densidade de aves e um melhor controle sanitário (SOUZA, 2015).

3.2 Importância das galinhas poedeiras na economia brasileira

As galinhas poedeiras possuem grande importância econômica no Brasil, sendo responsáveis por uma grande parte da produção de ovos no país.

O Brasil é o 13º maior produtor de ovos do mundo, com uma produção de cerca de 45 bilhões de unidades por ano, segundo a Anual (2022).

Além disso, a criação é uma importante fonte de renda para muitas famílias brasileiras, principalmente na zona rural. A atividade de criação de aves é uma das principais atividades agroindustriais do país, gerando emprego e renda para milhares de pessoas Anual (2022).

3.3 Manejo da criação avícola

De acordo com Oliveira (2004), o manejo adequado da criação de galinhas poedeiras é fundamental para garantir a qualidade dos ovos e a rentabilidade do negócio.

O uso de técnicas de manejo, como o controle de luminosidade, pode aumentar a produção de ovos em até 20%. Além disso, o manejo adequado contribui para a saúde e o bem-estar das aves, reduzindo o risco de doenças.

É fundamental que os produtores adotem práticas de manejo adequadas, como a alimentação balanceada, o controle de luminosidade e a manutenção de um ambiente limpo e confortável, como destacado por (OLIVEIRA, 2002). Além disso, o manejo adequado pode contribuir para a redução do impacto ambiental da atividade de criação de aves, por meio do uso de práticas sustentáveis de produção (SOUZA, 2015).

Ademais, a adoção de práticas de manejo adequadas também pode, também, melhorar a imagem da indústria avícola junto aos consumidores, que estão cada vez mais preocupados com a origem e qualidade dos alimentos que consomem (MENDES et al., 2013).

Em resumo, o manejo adequado da criação é crucial para garantir a qualidade dos ovos, a rentabilidade do negócio, a saúde e bem-estar das aves, a redução do impacto ambiental e a satisfação dos consumidores (OLIVEIRA, 2004).

Os principais tipos de manejo são:

- alimentação balanceada;
- controle de luminosidade;
- manutenção de um ambiente limpo e confortável;
- temperatura do ambiente de criação.

Dentre as diversas técnicas de manejo essenciais na criação de galinhas poedeiras, uma que se destaca pela sua influência direta na produção de ovos e no bem-estar das aves é o controle de luminosidade. Essa técnica de manejo desempenha um papel crucial no desenvolvimento das aves em diferentes fases, visando melhorar sua adaptação, ingestão de alimentos e água, e, consequentemente, a eficiência produtiva.

3.4 Manejo de luminosidade

O manejo luminoso é uma técnica utilizada para manipular o período de iluminação das aves, com o objetivo de maximizar a produção de ovos. Ele desempenha um papel crucial no desenvolvimento das aves em diferentes fases, visando melhorar sua adaptação, ingestão de alimentos e água, bem como estimular o crescimento. Durante a fase de crescimento, o objetivo é proporcionar um ambiente com iluminação adequada para permitir que as aves tenham acesso satisfatório à ração e água, favorecendo seu desenvolvimento PADOVAN (2009).

<https://www.overleaf.com/project/657279a12d08ce2c197e51c0> Além disso, na fase de produção de ovos o manejo luminoso tem um efeito decisivo sobre a maturidade sexual, produção, persistência, peso dos ovos e ocorrência de prolapsos de oviduto PADOVAN (2009).

Ao analisar o manejo luminoso, é importante considerar três variáveis principais: período de iluminação (fotoperíodo), intensidade da luz (lux) e o comprimento de onda (cor).

3.4.1 Fotoperíodo (duração da iluminação)

O fotoperíodo é o período de tempo em que as aves são expostas à luz. Existem dois tipos de fotoperíodo: hemeral e ahemeral. O fotoperíodo hemeral é aquele em que as aves são expostas a um período de luz seguido por um período de escuridão, enquanto o fotoperíodo ahemeral é aquele em que as aves são expostas a luz constante sem períodos de escuridão.

O fotoperíodo é uma variável importante para o desenvolvimento e produção de aves. É necessário que as galinhas recebam entre 14 e 16 horas de luz por dia para estimular a produção de ovos e garantir o seu bem-estar (PADOVAN, 2009). Ele tem, também, um impacto significativo na maturidade sexual das aves. O estudo realizado por Mendes et al. (2013) demonstrou que um fotoperíodo de 14 horas de luz e 10 horas de escuridão foi o mais eficiente para estimular a maturidade sexual.

Além disso, o mesmo estudo demonstrou que sendo submetidas a um fotoperíodo de 16 horas de luz e 8 horas de escuridão apresentaram uma produção de ovos mais elevada do que aquelas submetidas a um fotoperíodo de 12 horas de luz e 12 horas de escuridão.

3.4.2 Intensidade da luz (lux)

Para determinar a quantidade de lux necessária em instalações avícolas, é crucial levar em consideração as diferentes fases do ciclo de produção das galinhas.

Durante a fase de crescimento, pesquisas de referência, como o estudo de Mendes et al. (2013), indicam que uma intensidade luminosa em galpões fechados gira em torno de 20 a 30 lux e é adequada para promover a adaptação das aves e estimular o consumo de ração e água, contribuindo para um crescimento saudável.

A quantidade de lux necessária varia de acordo com a fase de crescimento e produção das aves, mas manter níveis apropriados é fundamental para garantir a saúde das aves e a qualidade dos ovos. Uma boa distribuição das lâmpadas e uma intensidade luminosa adequada são importantes para evitar áreas de sombreamento nos galpões avícolas.

Na fase de produção de ovos, a quantidade de lux necessária varia de acordo com as condições específicas da instalação e a genética das aves, mas geralmente, um nível de 5 a 10 lux é considerado adequado para estimular a postura regular de ovos (SOUZA, 2015).

No entanto, é importante mencionar que esses valores podem variar dependendo das diretrizes regionais e das condições ambientais específicas de cada criação.

Para evitar áreas de sombreamento nos galpões avícolas, é importante garantir uma boa distribuição das lâmpadas e uma intensidade luminosa adequada em torno de 10 lux em galpões abertos (JORDAN; TAVARES, 2005). É importante destacar que as aves percebem a luz de forma diferente dos seres humanos, sendo mais sensíveis a diferentes tipos de lâmpadas e intensidades luminosas (LEWIS; MORRIS, 2000).

3.4.3 Comprimento de onda (Cor)

As aves possuem uma percepção visual diferenciada em relação aos seres humanos, sendo mais sensíveis a certos comprimentos de onda de luz e apresentando uma capacidade superior de detectar variações na intensidade luminosa. Essa sensibilidade visual peculiar é atribuída às características fisiológicas e anatômicas específicas de seus olhos (Koenig, 2017; Hart, 2019).

Estudos científicos têm demonstrado que diferentes espécies de aves possuem sensibilidades específicas a determinados comprimentos de onda de luz. Por exemplo, as aves são particularmente sensíveis à luz azul e verde, enquanto apresentam uma sensibilidade reduzida à luz vermelha (HART; PARTRIDGE; CUTHILL, 1999).

Essa sensibilidade diferenciada a diferentes cores de luz está intimamente ligada à forma como as aves percebem seu ambiente, identificam alimentos e reconhecem sinais visuais importantes para a comunicação intraespecífica.

Além de sua sensibilidade a diferentes comprimentos de onda de luz, as galinhas são capazes de detectar variações sutis na intensidade luminosa. Mesmo quando os seres humanos não conseguem distinguir tais diferenças, as aves têm a habilidade de perceber variações na intensidade luminosa, mesmo em níveis baixos de luz (LEWIS; MORRIS, 2000).

Essa capacidade de detecção de variações na intensidade luminosa é especialmente relevante em ambientes naturais, nos quais a luz passa por mudanças ao longo do dia. As aves utilizam

essas variações na intensidade luminosa para regular seu comportamento, como determinar os horários de alimentação, descanso e atividade. Essa capacidade adaptativa é fundamental para a sobrevivência e sucesso das aves em seu ambiente natural (MENDES et al., 2013).

Considerando a percepção visual diferenciada das aves, é essencial selecionar lâmpadas adequadas e ajustar a intensidade luminosa de acordo com as necessidades visuais específicas delas. Esse cuidado permite criar um ambiente de iluminação adequado que promova o bem-estar e o desempenho das galinhas, contribuindo para a saúde e produtividade dos animais em sistemas de produção avícola.

3.5 Taxa de iluminação no Brasil

A intensidade luminosa do sol no Brasil varia ao longo do ano devido à sua localização geográfica próxima à linha do Equador e às mudanças sazonais causadas pela inclinação da Terra. O Brasil experimenta um clima tropical, o que significa que geralmente recebe uma quantidade significativa de luz solar durante todo o ano. No entanto, essa intensidade luminosa pode variar ao longo das estações.

Durante o verão, especialmente nas regiões mais próximas ao Equador, o Brasil recebe uma alta intensidade de luz solar, com dias mais longos. Já no inverno, essa intensidade pode diminuir ligeiramente, com dias mais curtos e menos luz solar direta, principalmente em regiões mais ao sul do país.

A intensidade luminosa do sol desempenha um papel importante na produção de ovos, principalmente em galinhas criadas em sistemas de produção ao ar livre ou semicontínuo, nos quais as aves são expostas à luz natural.

3.6 Sistema de controle

A utilização de controladores é uma prática comum em engenharia e indústria, devido ao fato de que a estabilidade e a precisão dos sistemas são fatores críticos. Os controladores têm a capacidade de manter variáveis de processo dentro de limites desejados, garantindo que os sistemas operem de maneira consistente e confiável. Essa consistência é crucial em áreas como manufa-

tura, pois a qualidade do produto final depende diretamente do controle preciso de variáveis, como temperatura, pressão e velocidade.

Os controladores realizam várias funções-chave em sistemas de controle. Primeiramente, eles comparam o valor de uma variável de processo com um valor de referência, conhecido como *setpoint*. A diferença entre esses valores, denominada erro é utilizada para calcular um sinal de controle que será aplicado ao sistema. Isso resulta em uma série de benefícios, incluindo a minimização de erros estáticos, a adaptação a perturbações no sistema e a melhoria na qualidade e eficiência da produção.

Além de garantir a precisão e a estabilidade dos processos, os controladores também desempenham um papel vital na automação. Eles permitem a operação de sistemas complexos com intervenção humana mínima, economizando tempo e recursos. A capacidade de adaptar-se a variações nas condições de operação torna os controladores uma peça fundamental na automação de sistemas dinâmicos.

3.7 Tipos de controladores

Diversos tipos de controladores desempenham funções específicas na regulação de sistemas, sendo essenciais na engenharia de controle e automação. Os principais controladores incluem o controlador Proporcional (P), o controlador Integral (I), o controlador Derivativo (D), o controlador Proporcional Integral (PI), e o controlador PID (Proporcional, Integral e Derivativo). Cada um desses desempenha um papel fundamental na estabilização e controle de sistemas dinâmicos.

O controlador Proporcional (P) é amplamente empregado na redução de erros estáticos (DUTTON; THOMPSON; BARRACLOUGH, 1997). Sua atuação consiste em ajustar a saída proporcionalmente à diferença entre a variável controlada e o *setpoint*. Essa abordagem é particularmente eficaz quando se busca um ajuste que corresponda diretamente à discrepância entre o valor desejado e o estado atual do sistema, sendo comumente utilizado em sistemas de controle.

O controlador Integral (I) desempenha um papel distinto, concentrando-se na eliminação de erros em estado estacionário. Sua funcionalidade envolve a acumulação do erro ao longo do tempo e a aplicação de correções graduais (ASTROM, 1995). Este é um componente valioso

quando o sistema enfrenta perturbações constantes, demandando uma compensação progressiva e consistente.

O controlador Derivativo (D) entra em cena com o objetivo de aprimorar a estabilidade e a resposta a perturbações nos sistemas. Sua ação reage à taxa de variação do erro, o que possibilita ajustes rápidos em situações em que o erro está sujeito a mudanças abruptas.

Além desses controladores individuais, é importante destacar o controlador PI. Esta abordagem combina a ação proporcional, que ajusta a saída de acordo com o erro atual, com a ação integral, que corrige erros passados (ASTROM, 1995). O controlador PI é altamente eficaz em uma variedade de contextos, especialmente quando se deseja eliminar erros estáticos e manter a estabilidade do sistema.

Por fim, o controlador PID integra todas essas ações em um conjunto coeso (OGATA, 2010). Essa abordagem é amplamente empregada em sistemas complexos e variáveis, nos quais a manutenção da estabilidade, precisão e capacidade de resposta é crucial. A equação geral para o controlador PID está representada na Equação 3.1.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (3.1)$$

Nesta fórmula:

- $u(t)$ representa o valor de controle desejado;
- K_p é o ganho proporcional;
- K_i é o ganho integral;
- K_d é o ganho derivativo;
- $e(t)$ é o erro entre a saída desejada e a saída real do sistema no instante t ;
- $\int_0^t e(\tau) d\tau$ representa a integral do erro ao longo do tempo;
- $\frac{de(t)}{dt}$ é a derivada do erro em relação ao tempo.

3.8 Métodos de sintonia de malha

Esses métodos englobam um conjunto de técnicas e abordagens, utilizadas para ajustar os parâmetros dos controladores de uma malha. O objetivo principal da sintonia de malha é otimizar o desempenho do sistema, garantindo estabilidade, resposta dinâmica adequada e minimização de erros.

3.8.1 Método de Ziegler-Nichols

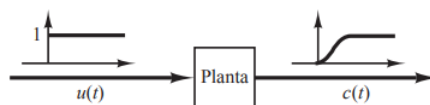
O método de sintonia de Ziegler-Nichols é um dos métodos mais populares para ajuste de controladores PID. Ele utiliza um método empírico baseado em observações práticas do comportamento do sistema.

De acordo com Ogata (2010), o método consiste em três etapas: identificação dos parâmetros críticos do sistema, determinação dos valores iniciais dos parâmetros do controlador PID e ajuste desses valores por meio de experimentação.

É possível obter experimentalmente a resposta da planta quando submetida à uma entrada em degrau unitário. No entanto, vale destacar que essa abordagem é adequada somente se a planta em questão não possuir integradores ou polos complexos conjugados dominantes. Nestes casos, a curva de resposta ao degrau unitário pode apresentar a característica de um 'S', como evidenciado na Figura 3.1.

Esse método é aplicável somente quando a curva de resposta à entrada de degrau unitário exibe essa peculiar forma de 'S' (OGATA, 2010). A configuração dessa curva pode ser obtida tanto por meio de experimentos práticos, quanto através de simulações dinâmicas da planta.

Figura 3.1 – Curva de resposta em forma de S



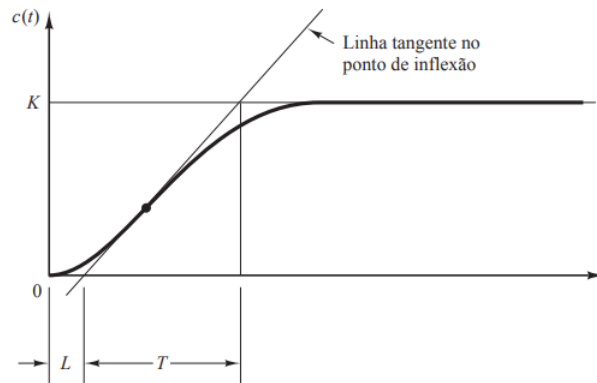
Fonte: Aguirre 2015.

A caracterização dessa curva em forma de ‘S’ envolve a identificação de duas constantes cruciais, conhecidas como atraso (L) e constante de tempo (T). Para determiná-las, torna-se necessário traçar uma linha tangente ao ponto de inflexão da curva ‘S’ e, subsequentemente, identificar o ponto de interseção dessa linha com o eixo do tempo e com a linha horizontal $c(t) = K$, como ilustrado na Figura 3.2.

A partir dessas informações, a função de transferência 3.3 pode ser aproximada por um sistema de primeira ordem com um atraso de transporte, de acordo com a Equação 3.2.

$$\frac{Y(s)}{X(s)} = \frac{Ke^{-\tau_d s}}{Ts + 1} \quad (3.2)$$

Figura 3.2 – Resposta ao degrau unitário de uma planta



Fonte: Aguirre 2015.

De acordo com (OGATA, 2010) os valores de K_p , T_i e T_d devem ser escolhidos de acordo com a tabela 3.1.

Tabela 3.1 – Regra de sintonia de Ziegler-Nichols baseada na resposta ao degrau.

Tipos de Controladores	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5L$

Fonte: Ogata,2015.

3.9 Modelagem matemática

A modelagem matemática é a área do conhecimento que visa obter representações de modelos matemáticos provenientes de sistemas físicos reais. Segundo Garcia (2005), para tais ações faz-se necessário obter profundos conhecimentos do sistema que pretende-se ser modelado. Com isso, a complexidade de tal ação está diretamente relacionada à complexidade física, matemática e à exatidão desejada na investigação do seu desempenho.

3.9.1 Identificação de sistemas

A identificação de sistemas é uma abordagem alternativa à modelagem matemática (AGUIRRE, 2015). Ao contrário da modelagem matemática, a identificação de sistemas não requer um profundo conhecimento prévio do sistema em estudo.

Em vez disso, ela se baseia na obtenção de dados experimentais do sistema e na criação de um modelo a partir deles. De acordo com (OGATA, 2010), a identificação de sistemas permite obter modelos que se aproximam do comportamento real, tornando-a uma ferramenta valiosa.

3.9.2 Função de transferência

A função de transferência é uma ferramenta fundamental na identificação de sistemas e na análise de sistemas dinâmicos. Ela fornece uma representação matemática simplificada das relações entre a entrada e a saída de um sistema, facilitando a compreensão e a modelagem de sistemas lineares invariantes no tempo.

De acordo com Aguirre (2015), a representação matemática da função de transferência de um sistema linear invariante no tempo é, geralmente, definida como mostrada na Equação 3.3:

$$H(s) = \frac{Y(s)}{X(s)} \quad (3.3)$$

Onde:

- $H(s)$ é a função de transferência do sistema;
- $Y(s)$ é a transformada de Laplace da saída do sistema;

- $X(s)$ é a transformada de Laplace da entrada do sistema;
- s é a variável complexa de Laplace.

A função de transferência descreve como o sistema responde às diferentes frequências da entrada, ou seja, como a amplitude e a fase da entrada são afetadas ao passar pelo sistema, (AGUIRRE, 2015). Ela é expressa em termos da variável complexa s , que representa a frequência complexa. A partir da função de transferência, é possível determinar várias propriedades do sistema, como a resposta em frequência, a estabilidade e a capacidade de atenuar ou amplificar determinadas frequências.

Na identificação de sistemas, a função de transferência é frequentemente obtida a partir de dados experimentais. Isso envolve a aplicação de uma entrada conhecida ao sistema e a medição da resposta correspondente.

Com base nessas medições, é possível estimar os coeficientes da função que melhor descrevem o comportamento do sistema. Essa é uma tarefa crucial na modelagem e no controle de sistemas reais, pois permite a criação de modelos matemáticos que representam com precisão o comportamento dinâmico dos sistemas em estudo.

3.10 Métodos determinísticos

Os Métodos determinísticos são abordagens de análise e modelagem de sistemas que se baseiam na suposição de que o comportamento de um sistema pode ser completamente representado por equações matemáticas precisas e previsíveis (RAKE, 1979).

Esses métodos se concentram em descrever sistemas de forma determinística, sem considerar variações aleatórias ou ruídos em suas respostas. Eles assumem que os sistemas são governados por leis físicas exatas e que seus resultados podem ser previstos com alta precisão.

Os métodos determinísticos não são adequados para lidar com sistemas sujeitos a flutuações imprevisíveis, como ruídos em medições ou sistemas altamente complexos nos quais fatores não determinísticos desempenham um papel significativo. Entre os principais métodos determinísticos de análise e modelagem de sistemas, destacam-se:

- **Método das Diferenças Finitas:** este método se baseia na discretização das equações que governam o sistema para representá-lo em termos de diferenças finitas, tornando-o mais acessível para análise numérica;
- **Método de Elementos Finitos:** amplamente utilizado na engenharia, este método divide um sistema em elementos menores, permitindo a análise detalhada de cada parte do sistema;
- **Método de Runge-Kutta:** especialmente eficaz na resolução numérica de equações diferenciais, o Método de Runge-Kutta é frequentemente empregado na simulação de sistemas dinâmicos;
- **Método Sunderasen:** o Método Sunderasen baseia-se na aplicação de princípios analíticos e matemáticos para entender o funcionamento de sistemas complexos, permitindo a análise detalhada de suas características e comportamento.

Dentre os principais citados, destaca-se o Método Sunderasen, que desempenhou um papel fundamental na condução desta pesquisa. Sua aplicação proporcionou *insights* valiosos e uma compreensão aprofundada do comportamento dos sistemas complexos investigados.

3.10.1 Método Sunderasen

O método de Sunderasen pressupõe que o sistema pode ser aproximado satisfatoriamente por uma função de transferência de segunda ordem, apresentando características sobreamortecidas (que não apresentam oscilações quando em sua entrada é aplicado um degrau unitário), representada pela Equação 3.4.

$$H(s) = \frac{e^{-\tau_d s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (3.4)$$

Onde:

- τ_d é atraso puro de tempo;
- τ_1 e τ_2 são constantes de tempo fixas.

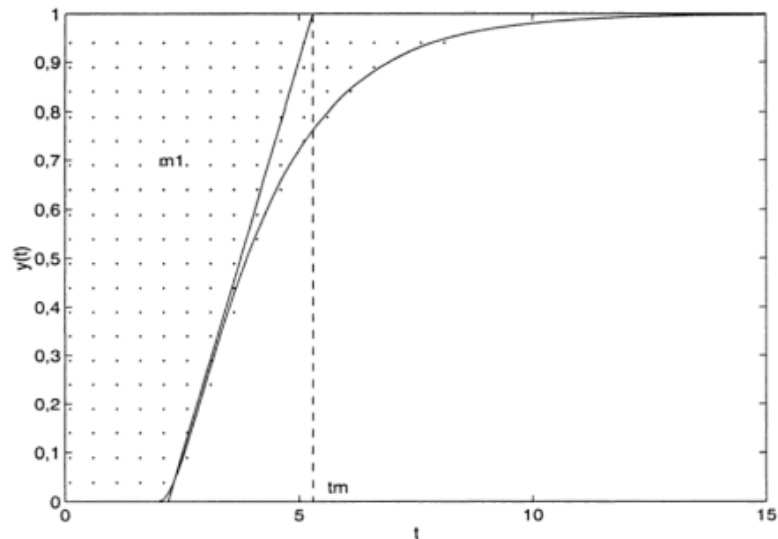
Segundo Aguirre (2015) o objetivo do método é determinar os parâmetros τ_d, τ_1 e τ_2 da respectiva função de transferência, a partir de uma entrada ao degrau.

A Figura 3.3 representa a resposta de uma sistema sobreamortecido, no qual, aplicando uma entrada degrau unitário no instante $t = 0$, pressupõe que a saída obtenha valor unitário em regime permanente.

Portanto, para que o sistema realizado experimentalmente se enquadre, é necessário determinar o ganho DC e, logo em seguida, normalizar os dados reais antes de aplicar o método de Sundaresan.

Para realizar o procedimento é necessário encontrar o valor da área hachurada, presente na Figura 3.3, e, a partir disso, utilizar a Equação 3.5, na qual está sendo realizada a subtração do valor do degrau e da função $y(t)$ em todo t .

Figura 3.3 – Resposta ao degrau de sistemas sobreamortecido.



Fonte: Aguirre,2015.

$$m_1 = \int_0^{\infty} (1 - y(t)) dt \quad (3.5)$$

Logo em seguida, faz-se necessário remover o valor da condição inicial e realizar a normalização, de maneira que seu valor em estado estacionário seja igual a 1 ou $y(\infty) = 1$.

Realizada a normalização e encontrado o valor de m_1 presente na Equação 3.5, o próximo passo é encontrar o valor de Mi (inclinação da reta tangente no ponto de inflexão de $y(t)$, sendo obtido pela Equação 3.6.

$$Mi = \frac{\eta^{1/(1-\eta)}}{\alpha(\eta - 1)} \quad (3.6)$$

onde $\eta = \tau_1/\tau_2$ e $\alpha = \tau_1\tau_2/(\tau_1 - \tau_2)$.

A reta Mi intercepta o valor final de $y(t)$ em um ponto correspondente ao instante de tempo t_m dado pela Equação 3.7.

$$t_m = \tau_1 + \alpha \left[\ln(\eta) + \frac{\eta^2 - 1}{\eta} \right] \quad (3.7)$$

Combinando as equações 3.5, 3.6, 3.7, obtém-se:

$$\lambda = (t_m - m_1)Mi \quad (3.8)$$

Por fim, é possível determinar os valores de τ_1, τ_2 e τ_3 , como se segue:

$$\tau_1 = \frac{\eta^{\frac{\eta}{1-\eta}}}{Mi} \quad (3.9)$$

$$\tau_2 = \frac{\eta^{\frac{1}{1-\eta}}}{Mi} \quad (3.10)$$

$$t_d = m_1 - \tau_1 - \tau_2 \quad (3.11)$$

3.11 Funções de otimização

As funções de otimização desempenham um papel essencial na engenharia, permitindo avaliar e aprimorar o desempenho de sistemas de controle. Essas funções são utilizadas para quantificar o quão bem um sistema atende a seus objetivos de controle. Elas podem variar em termos de complexidade e aplicação, mas, em essência, todas buscam medir o desvio entre a saída desejada e a saída real de um sistema ao longo do tempo.

As funções de otimização podem ser usadas para calcular medidas de erro, como o erro absoluto, o erro quadrático médio ou outras métricas específicas. A escolha da função de otimização adequada depende dos objetivos do controle e das características do sistema.

Essas funções são valiosas ferramentas para avaliar o desempenho de sistemas, comparar diferentes estratégias de controle e ajustar os parâmetros do controlador para atender aos requisitos.

Três funções amplamente utilizadas na análise de desempenho de sistemas de controle: o IAE (Integral of Absolute Error), o ITAE (Integral of Time-weighted Absolute Error) e o MSE (Mean Squared Error). Cada uma dessas funções possui características específicas que a tornam adequada para diferentes cenários de controle.

O IAE, ou Integral do Erro Absoluto, é uma métrica valiosa quando se busca a minimização de erros absolutos em sistemas de controle. Ele avalia o desempenho calculando a integral do valor absoluto do erro entre a saída desejada e a saída real ao longo do tempo.

Isso significa que o IAE coloca grande ênfase na redução de erros independentemente do tempo em que ocorrem, sendo particularmente útil em sistemas em que é crítico manter os erros absolutos tão baixos quanto possível. A equação do IAE está descrito na Equação 3.12.

$$\int_0^{\infty} |e(t)| dt \quad (3.12)$$

Por outro lado, o ITAE, ou Integral do Erro Absoluto Ponderado no Tempo, leva em consideração a importância de minimizar erros em momentos diferentes durante a resposta do sistema. Ele aplica um peso ao erro que varia em função do tempo, permitindo que erros ocorridos em momentos mais tardios tenham um impacto menor na função de otimização.

Isso torna o ITAE valioso quando se deseja reduzir erros em momentos específicos da resposta do sistema, dando prioridade a janelas temporais críticas. A equação do ITAE está representado na Equação 3.13.

$$\int_0^{\infty} t |e(t)| dt \quad (3.13)$$

O MSE, ou Erro Quadrático Médio, é uma função amplamente empregada em sistemas que buscam minimizar o desvio quadrático médio. Ele calcula a média dos quadrados dos erros entre a saída desejada e a saída real.

O MSE é particularmente útil quando se deseja atenuar o impacto de erros mais extremos, uma vez que a elevação ao quadrado amplifica erros maiores.

Portanto, o MSE é comumente utilizado em sistemas que buscam minimizar o desvio quadrático médio global ao longo do tempo. A equação do MSE está representado na Equação 3.14.

$$\int_0^{\infty} |e(t)|^2 dt \quad (3.14)$$

3.12 Sintonia com a função *fmincon* de otimização do MATLAB

Para utilização das funções de otimização que foram utilizados neste trabalho, foi utilizado o software MATLAB (Versão 2020b). O MATLAB oferece facilidades para configurar os parâmetros das funções e permite estender escrevendo código em um arquivo M-file.

Além disso, é possível combinar os códigos com outras *toolboxes* de otimização. Essas características, juntamente com as funções de otimização do MATLAB para comparação, influenciaram a escolha desta ferramenta.

A função *fmincon* faz parte da toolbox de otimização do MATLAB e é projetada para resolver problemas com múltiplas variáveis sujeitas a restrições, sejam elas lineares ou não lineares. Essa função emprega o método conhecido como Programação Quadrática Sequencial (PSQ).

O PSQ tem demonstrado ser altamente eficaz na resolução de problemas de programação não linear, resultando em soluções bem-sucedidas. Durante cada iteração, o problema é subdividido em subproblemas para simplificar a busca pelo ponto ótimo, e esses subproblemas são resolvidos por meio de programação quadrática. A função *fmincon* esta descrita na Fórmula 3.15.

Para obter informações mais detalhadas sobre esse método de busca, é possível consultar a documentação de ajuda do (SIVANANDAM et al., 2020).

$$\text{Min}f(x) = \begin{cases} c(x) \leq 0 \\ \text{ceq}(x) = 0 \\ A \cdot x \leq b \\ Aeq = beq \\ lb \leq x \leq ub \end{cases} \quad (3.15)$$

Onde:

- x : representa o vetor que define os valores iniciais das variáveis independentes;
- $f(x)$: representa a função a ser minimizada.
- b e beq : são vetores que representam, respectivamente, os termos independentes das restrições lineares de desigualdade e igualdade;
- lb e ub : são vetores que representam os limites inferiores e superiores das variáveis independentes, respectivamente;
- A e Aeq : são matrizes que definem as funções das restrições lineares de desigualdade e igualdade, respectivamente;
- $c(x)$ e $ceq(x)$: são funções que retornam vetores e definem, respectivamente, as restrições não lineares de desigualdade e igualdade.

A função *fmincon* emprega uma abordagem baseada no princípio da "subida da encosta" para otimização, avaliando os pontos adjacentes no espaço de busca e se movendo na direção que minimiza a função objetivo. O critério de parada para essa função é acionado quando um determinado mínimo padrão é alcançado.

Esses métodos de otimização buscam encontrar uma solução com base em uma semente, que é uma estimativa inicial da solução. Quanto mais próxima a semente estiver da solução real, maior será a probabilidade de sucesso da otimização.

A complexidade na dedução das sementes varia de acordo com o tipo de problema, sendo mais simples em alguns casos e mais desafiadora em outros. Em situações mais complexas, é

difícil estimar sementes adequadas e, geralmente, é necessário recorrer a algum conhecimento prévio sobre o problema.

No entanto, é crucial evitar o uso de sementes com valores extremos. Nesse trabalho, os valores das sementes foram dos parâmetros K_p , K_i e K_d , alcançados por meio do método de Ziegler-Nichols, garantindo que as sementes iniciais razoáveis para iniciar as simulações.

A estrutura de sintaxe empregada para a função *fmincon* encontra-se ilustrada na Figura 3.4.

Figura 3.4 – Sintaxe para função *fmincon*

```
[x, fval, exitflag, output] = fmincon(nomearquivo, x0, op)
```

Fonte: do Autor

Onde:

- x - representa a solução encontrada pelo processo de otimização;
- $fval$ - indica o valor da função objetivo (FO) calculado na solução x ;
- $exitflag$ - é um indicador da condição de término da otimização. Se $exitflag > 0$, isso significa que a função convergiu para a solução x . Se $exitflag = 0$, indica que o número máximo de iterações foi atingido. Se $exitflag < 0$, significa que a função não convergiu para a solução;
- $output$ - é uma estrutura que contém informações detalhadas sobre o processo de otimização;
- 'nomearquivo' - é o nome do arquivo que armazena a função objetivo;
- $x0$ - representa a semente inicial utilizada no processo de otimização;
- op - consiste no vetor de opções do algoritmo de otimização.

4 METODOLOGIA

4.1 Estrutura física

No estágio inicial do desenvolvimento do projeto de controle de luminosidade, uma base sólida foi estabelecida para orientar este projeto. Com esse propósito, optou-se pela aquisição de um celeiro em MDF. Essa decisão estratégica não apenas proporciona uma representação física concreta do ambiente, mas também, se mostra fundamental para garantir a eficácia na implementação das estratégias de controle luminoso.

Além disso, com o intuito de otimizar a seleção, foi buscado um celeiro em MDF com dimensões eficientes e requisitos de montagem acessíveis, visando garantir a viabilidade prática. A escolha recaiu sobre um celeiro cujas características permitissem uma representação fiel do espaço, priorizando uma estrutura simples para facilitar o processo de montagem.

A estrutura possui uma espessura de material de 2,8 mm, altura de 26,5 cm, largura de 32 cm, profundidade de 40 cm, altura das portas de 10 cm e segue uma escala aproximada de 1:20. A representação do celeiro pode ser observada na Figura 4.1.

Figura 4.1 – Celeiro parte externa



Fonte: do Autor.

A parte interna do celeiro está salientada na Figura 4.2.

Figura 4.2 – Celeiro parte interna



Fonte: do Autor.

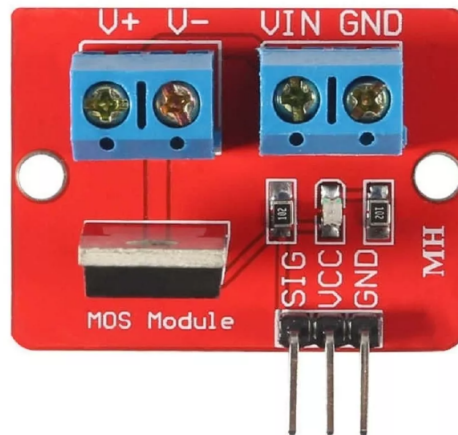
4.1.1 Módulo IRF520

O IRF520 é um transistor MOSFET de canal N. MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors). Eles são dispositivos semicondutores que podem ser utilizados como interruptores eletrônicos para controlar mono motores, lâmpadas ou outros dispositivos que consomem corrente significativa.

O IRF520, Figura (4.3), opera como um interruptor controlado por tensão. Quando uma tensão adequada é aplicada ao terminal de controle *gate*, ele permite ou bloqueia o fluxo de corrente entre os terminais de fonte e dreno. Isso o torna útil para controlar dispositivos de alta potência com sinais de baixa potência.

As Conexões típicas do IRF520 são: *gate* (G), que é o terminal de controle, a tensão aplicada aqui determina se o MOSFET está ligado ou desligado; *drain* (D), que é o terminal de saída de alta potência que faz com que a corrente flua do dreno para a fonte quando o MOSFET está ligado; *source* (S), que é o terminal de referência ou ponto de retorno para a corrente.

Figura 4.3 – MOSFET IRF520

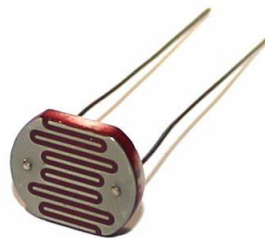


Fonte: do Autor

4.1.2 Sensor LDR

O sensor LDR, ou Resistor Dependente de Luz, representado na Figura 4.4, é um componente eletrônico sensível à luz que desempenha um papel crucial em sistemas de controle de luminosidade. A funcionalidade do módulo LDR é baseada na característica intrínseca do semicondutor utilizado em sua construção. Quando exposto à luz, a resistência do LDR diminui, permitindo uma maior passagem de corrente elétrica. Em contrapartida, em ambientes mais escuros a resistência aumenta, restringindo o fluxo de corrente. Essa variação na resistência proporciona uma medida eficaz da luminosidade do ambiente. O sensor utilizado possuía dimensões de 10 mm.

Figura 4.4 – Sensor LDR



Fonte: do Autor.

4.1.3 Microcontrolador Arduino

O Arduino, um microcontrolador de código aberto, é projetado para integrar em um único *chip* um processador, memória e periféricos de entrada/saída. Sua funcionalidade abrange desde o acionamento de luzes até o controle de motores e sensores, destacando-se como uma opção ideal para diversos projetos, incluindo automação residencial, sistemas de monitoramento e, no contexto deste estudo, controle de luminosidade.

A linguagem de programação simplificada baseada em C/C++ utilizada pelo Arduino facilita a criação e implementação de códigos. A versatilidade do Arduino é evidente na sua capacidade de se adaptar a diferentes tarefas por meio de *shields* e módulos, ampliando suas capacidades conforme as exigências específicas do projeto.

A praticidade do Arduino, representada na Figura 4.5, é uma característica distintiva, refletida na sua abordagem amigável ao usuário. O ambiente de desenvolvimento integrado (IDE) proporciona uma interface intuitiva para criação, edição e carregamento de códigos no microcontrolador. A ampla comunidade de usuários complementa esses recursos, oferecendo tutoriais, exemplos de código e fóruns de discussão.

Figura 4.5 – Arduino Uno



Fonte: do Autor.

4.1.4 Fonte de alimentação 12V

Foi empregada uma fonte de alimentação de 12V para fornecer a energia necessária ao sistema Figura 4.6. Essa fonte de alimentação opera com uma voltagem de 12V, desempenhou um papel crucial ao garantir a alimentação adequada dos componentes do sistema, assegurando seu funcionamento estável.

Figura 4.6 – Fonte chaveada 12v



Fonte: do Autor.

4.1.5 Lâmpada

Para a iluminação no contexto do experimento, foi empregada uma lâmpada específica, a Lâmpada Farol H4 12V 60/55W da Philips, modelo 12342C1. Esta lâmpada, como mostra a Figura 4.7, opera com uma tensão de 12V e oferece uma potência de 60W para o filamento alto e 55W para o filamento baixo. A escolha dessa lâmpada atende às necessidades de iluminação do sistema, proporcionando uma fonte luminosa estável.

Figura 4.7 – Lâmpada 12V 60/55W



Fonte: do Autor.

4.1.6 Protoboard

A protoboard, por sua vez, desempenhou um papel fundamental como plataforma para a interconexão dos componentes, proporcionando uma organização eficiente e simplificando as ligações elétricas. Ao oferecer um ambiente prático para a montagem dos circuitos, permitiu ajustes e modificações durante o desenvolvimento do projeto, o que contribuiu para a agilidade e a flexibilidade necessárias na fase de testes e implementação do sistema.

4.2 Esquemático

Para realizar a montagem do sistema de luminosidade foi necessário a utilização do microcontrolador Arduino UNO para geração de pulsos PWM e um módulo mosfet IRF520, e assim, controlar intensidades de luz fornecida pela lâmpada. Para a extração dos dados foi utilizado o sensor de luminosidade (LDR) o qual estava localizado próximo da lâmpada.

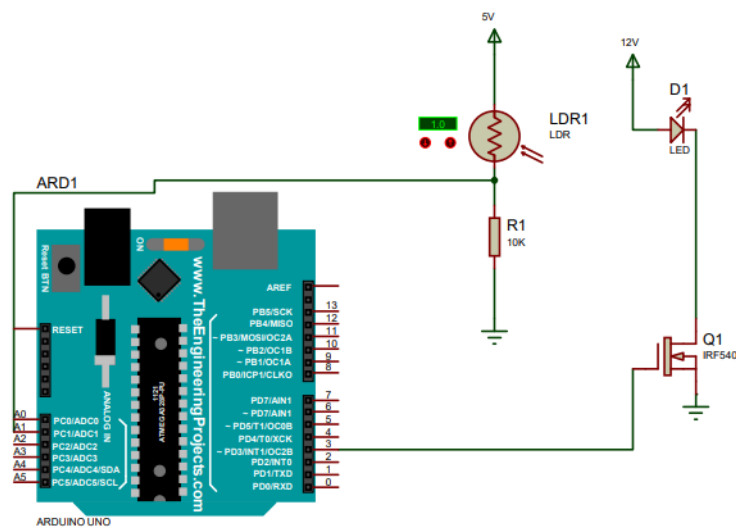
Para a montagem física do projeto, foi estabelecida uma conexão crucial entre a porta 3 do Arduino e o VCC do MOSFET IRF520. Essa conexão foi fundamental para o controle dos pulsos de PWM, possibilitando a modulação eficaz da intensidade luminosa.

Além disso, os pinos VIN e GND do MOSFET foram ligados à saída da fonte de alimentação, garantindo a devida alimentação do componente. O V+ e V- do MOSFET IRF520 foram conectados à lâmpada, estabelecendo assim a ligação essencial para o controle da luminosidade.

Paralelamente, a porta A1 do Arduino foi conectada à interseção entre a resistência de 10 k Ω e o sensor LDR. Essa configuração possibilitou a captação eficiente dos dados de luminosidade pelo sensor. A resistência de 10 k Ω desempenhou o papel na regulação da corrente, garantindo uma leitura precisa e estável do sensor LDR.

O esquemático das ligações está representada na Figura 4.8.

Figura 4.8 – Esquemático das ligações



Fonte: do Autor.

4.3 Modelagem do sistema

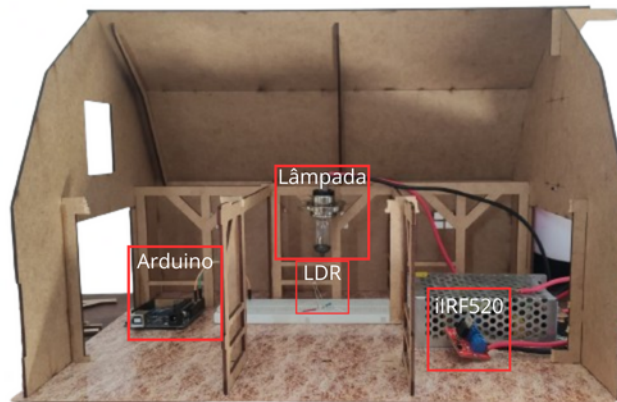
O processo de montagem do celeiro foi realizado, com todas as peças devidamente posicionadas para garantir a integridade e funcionalidade do projeto. Para uma visualização mais detalhada, a vista superior do projeto pode ser observada na Figura 4.9, proporcionando uma perspectiva abrangente da disposição das peças. Além disso, a vista lateral, está representada na Figura 4.10.

Para realizar a modelagem dos dados de forma precisa, procedeu-se com o fechamento completo do ambiente do celeiro, eliminando qualquer interferência de iluminação externa. Essa abordagem visou criar um ambiente controlado, garantindo que as leituras do sensor LDR refletissem exclusivamente as condições de luminosidade interna ao celeiro.

A aquisição dos dados foi conduzida pelo Arduino, que desempenhou um papel central na coleta e registro das leituras do sensor. O Arduino assegurou uma aquisição eficiente e sistemática dos dados de luminosidade dentro do ambiente fechado do celeiro.

A montagem do projeto pode ser observado na Figura 4.9.

Figura 4.9 – Celeiro com equipamentos montados vista frontal



Fonte: do Autor.

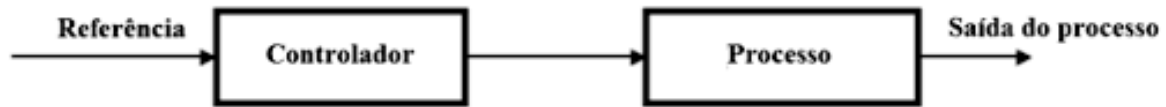
Figura 4.10 – Celeiro com equipamentos montados vista superior



Fonte: do Autor.

Para realizar a modelagem do sistema foi feito o experimento utilizando o modelo em malha aberta como pode ser visto na Figura 4.11 . O controlador foi posto em manual de tal maneira que fosse realizada três entradas degraus no sistema, onde foram utilizados três ciclos de PWM pré-estabelecidos que teria a entrada de 33%, 66% e 99% do Arduino. O código utilizado para realizar a modelagem esta presente no Apêndice A.

Figura 4.11 – Malha aberta do sistema de luminosidade.



Fonte: do Autor.

A Figura 4.12 mostra os valores obtidos para os três processos.

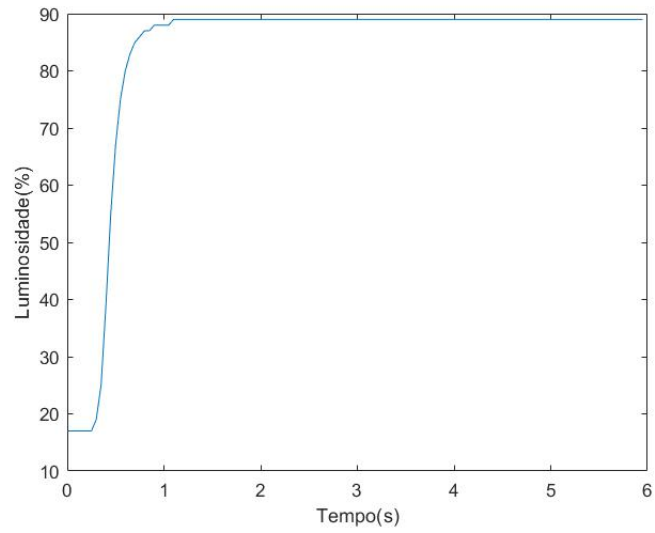
Figura 4.12 – Tabela com os valores encontrados para porcentagem de luminosidade em relação ao Duty Cycle de 33%, 66% e 99%

Tempo(S)	duty cycle(%)	Luminosidade(%)	duty cycle(%)	Luminosidade(%)	duty cycle(%)	Luminosidade(%)
0,05	33	0	66	0	99	0
0,1	33	0	66	0	99	49
0,15	33	0	66	43	99	77
0,2	33	0	66	73	99	79
0,25	33	0	66	77	99	80
0,3	33	2	66	78	99	81
0,35	33	8	66	79	99	81
0,4	33	22	66	79	99	81
0,45	33	38	66	79	99	81
0,5	33	50	66	79	99	81
0,55	33	58	66	79	99	81
0,6	33	63	66	79	99	81
0,65	33	66	66	79	99	81
0,7	33	68	66	79	99	81
0,75	33	69	66	79	99	81
0,8	33	70	66	79	99	81
0,85	33	70	66	79	99	81
0,9	33	71	66	79	99	81
0,95	33	71	66	79	99	81
1	33	71	66	79	99	81

Fonte: do Autor.

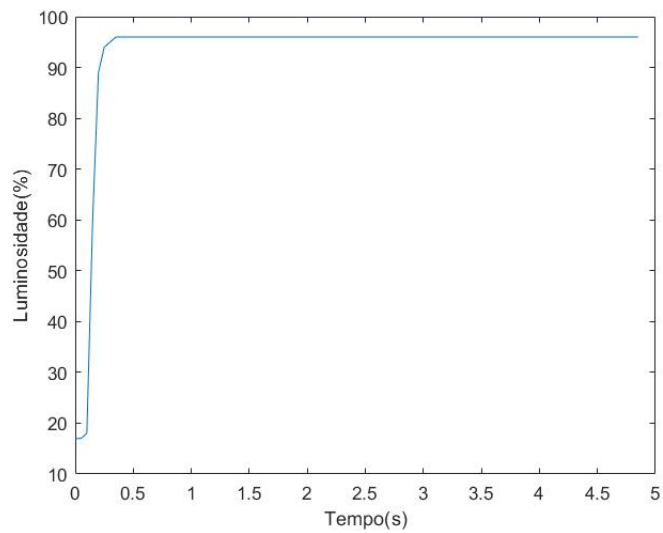
Após a coleta dos dados, as figuras 4.13, 4.14, 4.15 trazem gráficos que ilustram os valores de saída de intensidade luminosa para cada porcentagem de PWM analisada. Esses gráficos fornecem uma representação visual clara e abrangente das relações entre os pulsos de largura modulada e os níveis correspondentes de intensidade luminosa.

Figura 4.13 – Gráfico da intensidade luminosa em relação ao Duty Cycle de 33%



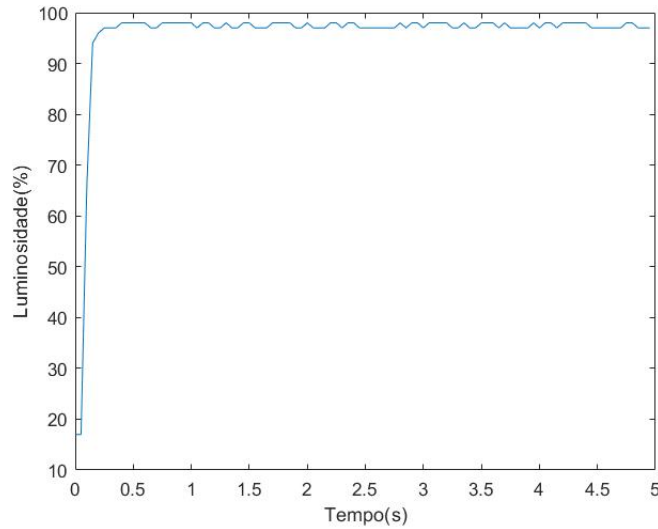
Fonte: do Autor.

Figura 4.14 – Gráfico da intensidade luminosa em relação ao Duty Cycle de 66%



Fonte: do Autor.

Figura 4.15 – Gráfico da intensidade luminosa em relação ao Duty Cycle de 99%



Fonte: do Autor.

Com a os valores coletados por meio do Arduino e salvos em uma planilha eletrônica, foi possível realizar a modelagem dos três sistemas utilizando o método de Sundaresem, o qual pressupõe que o sistema pode ser aproximado satisfatoriamente por uma função de transferência de segunda ordem, representada pela Equação 3.4 .

Após utilizar o método de Sundaresem os valores foram salvos no *software* MATLAB para obtenção gráfica de transferência de cada modelo. Os resultados obtidos para 33%, 66% e 99% do PWM do Arduino estão expostos nas equações 4.1 , 4.2 e 4.3 .

$$H_{33\%}(s) = \frac{-7.558e^{-0.25s}}{s^2 + 18.37s + 99.62} \quad (4.1)$$

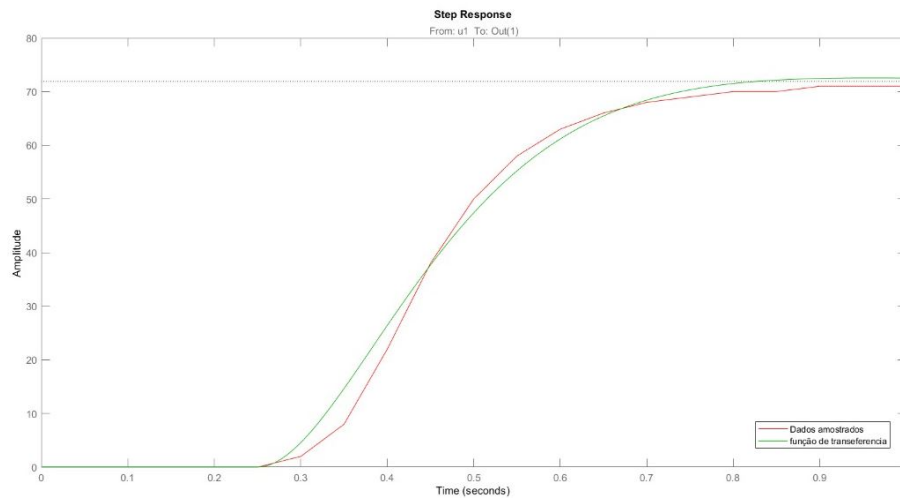
$$H_{66\%}(s) = \frac{1479e^{-0.1s}}{s^2 + 64.97s + 1236} \quad (4.2)$$

$$H_{99\%}(s) = \frac{1666e^{-0.05s}}{s^2 + 74.79s + 17} \quad (4.3)$$

As figuras 4.16 , 4.17 e 4.18 mostram gráficos que apresentam uma comparação visual entre os valores da função de transferência obtidos por meio do método de Sunderasen e os valores amostrados pelo PWM do Arduino.

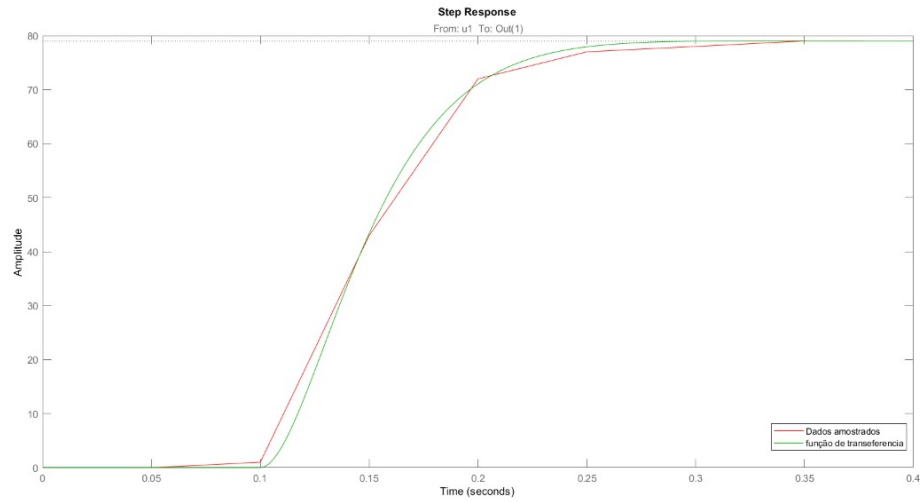
Cada conjunto de gráficos representa uma condição específica de operação, correspondendo aos diferentes níveis de pulsos de largura modulada (PWM). A sobreposição das curvas permite uma avaliação imediata da eficácia do modelo teórico em descrever com precisão o comportamento dinâmico dos sistemas diante das variações no controle de entrada.

Figura 4.16 – Resposta do sistema ao degrau de 33% de Duty Cycle juntamente com o modelo obtido.



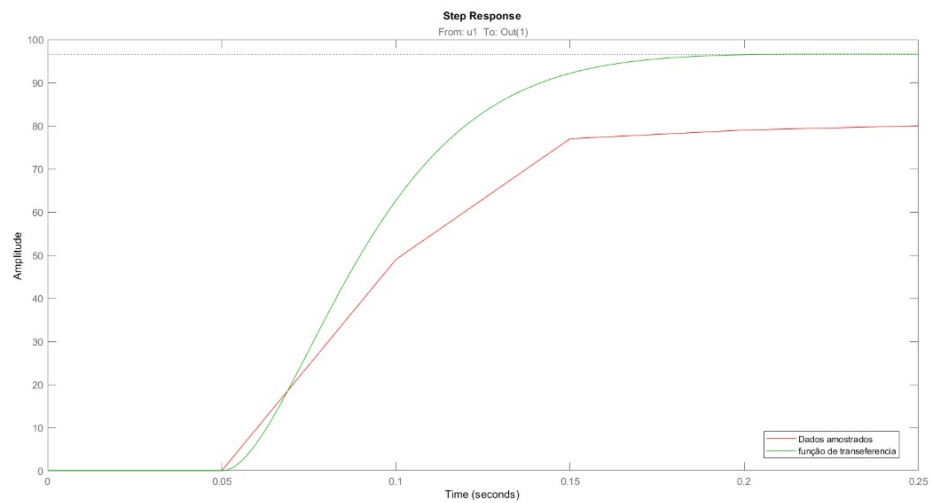
Fonte: do Autor.

Figura 4.17 – Resposta do sistema ao degrau de 66% de Duty Cycle juntamente com o modelo obtido.



Fonte: do Autor.

Figura 4.18 – Resposta do sistema ao degrau de 99% de Duty Cycle juntamente com o modelo obtido.



Fonte: do Autor.

A comparação entre as três funções de transferência obtidas pelo método de Sunderasen e seus respectivos valores amostrados foi realizada utilizando o *software* MATLAB. Os resultados revelaram uma equivalência de 89% para a Equação 4.1, 96% para a Equação 4.2 e 67% para a Equação 4.3.

Esses índices de concordância refletem a capacidade de cada modelo teórico em se alinhar aos dados experimentais provenientes do PWM do Arduino. Notavelmente, a Equação 4.2, associada a 66% do valor do PWM, demonstrou a mais alta correspondência, atingindo 96% de equivalência.

Diante desse resultado, a escolha desta Equação 4.2 para representar o sistema é respaldada pela sua notável capacidade de reproduzir de forma precisa o comportamento dinâmico observado nas condições de controle analisadas.

4.4 Controle

A proposta de implementação de um controle proporcional-integral (PI) destaca-se como uma escolha estratégica para melhorar o desempenho do sistema em análise. A justificativa reside na resposta transitória já adequada do sistema, indicando que o controle proporcional por si só pode não ser suficiente para alcançar os objetivos desejados. Assim, a introdução da ação integral visa corrigir o erro em regime permanente, promovendo uma estabilidade ainda maior ao sistema de controle.

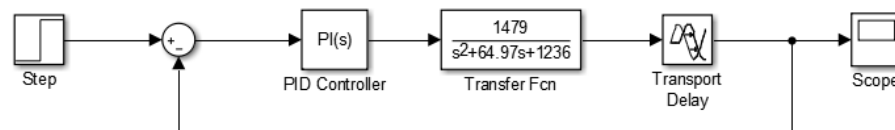
Considerando um sistema com tempo morto $L = 0,1$ e tempo integral $T = 0,15$, extraídos da Figura 4.12, foi realizada a sintonia do controlador PI pelo método de Ziegler-Nichols, o qual foi desenvolvida utilizando a Tabela 3.1. Os valores encontrados foram de $K_p = 0,45$ e $K_i = 1,275$, os quais foram utilizados para implementar o controlador PI do sistema.

No intuito de validar os parâmetros obtidos pelo método de Ziegler-Nichols para a Equação 4.2, uma etapa crucial envolveu a realização de testes utilizando o *software* MATLAB, especificamente a ferramenta Simulink. Por meio dessa ferramenta, foi possível simular o comportamento do sistema ao empregar os valores sintonizados de K_p e K_i .

A montagem do projeto no Simulink, representada na Figura 4.19, proporciona uma visão detalhada da simulação realizada para testar os parâmetros obtidos. A configuração do modelo inclui elementos essenciais, tais como uma função degrau com valor unitário, um bloco para o controlador PI, um bloco representando a Equação 4.2, um bloco que modela um atraso puro de tempo e, por fim, um bloco de visualização (Scope).

A função degrau com valor unitário serve como entrada para o sistema, enquanto o bloco do controlador aplica os parâmetros sintonizados, seguido pela implementação da função de transferência. O bloco de atraso puro de tempo representa as características temporais do sistema, e o Scope permite a observação e a análise visual da resposta do sistema em malha fechada.

Figura 4.19 – Diagrama de Blocos do sistema de controle



Fonte: do Autor.

A representação visual desses testes se encontra na Figura 4.20, que ilustra a resposta do sistema em malha fechada com o controlador PI sintonizado.

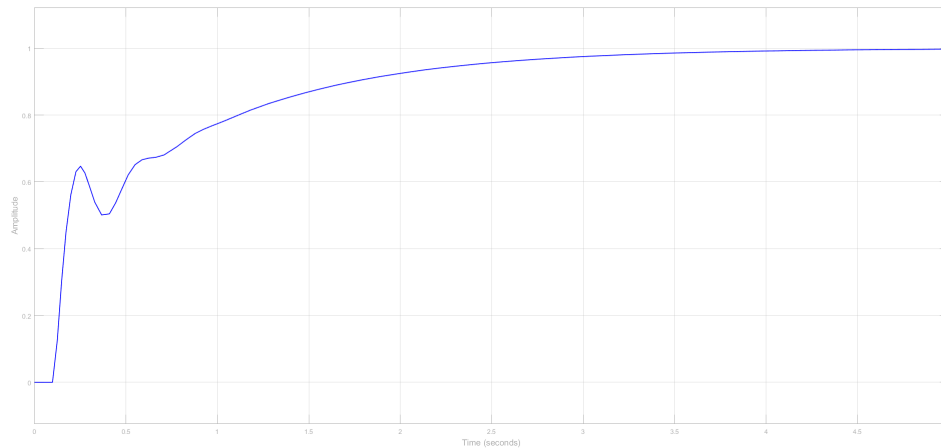
A análise dos resultados gráficos provenientes do Simulink permitiu verificar a eficácia do método de Ziegler-Nichols na sintonia do controlador para a Equação 4.2, fornecendo um valor de *undershoot* de 1,81% e com um tempo de estabelecimento de 5,164s.

4.5 Funções de otimização

No prosseguimento da otimização do sistema de controle, os parâmetros obtidos pelo método de Ziegler-Nichols foram utilizados como sementes para as funções de otimização, especificamente para os índices de desempenho IAE, ITAE e MSE.

O objetivo primordial do sistema de controle é alcançar o melhor desempenho possível. Portanto, é essencial definir parâmetros que minimizem o índice de desempenho escolhido, visando a otimização do sistema.

Figura 4.20 – Resposta do sistema com parâmetros de Ziegler-Nichols

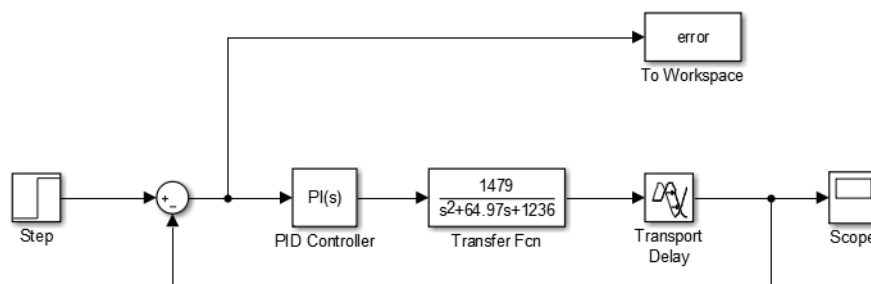


Fonte: do Autor.

Os resultados obtidos de K_p e K_i das funções de otimizações serão comparados e validados em relação à abordagem clássica de sintonia proposta por Ziegler-Nichols (Z-N), a fim de proporcionar uma análise abrangente e confiável.

A Figura 4.21 ilustra a montagem do sistema no Simulink, onde foi utilizado a mesma estrutura da figura 4.19, entretanto foi acrescentado o bloco "To Workspace" com o objetivo de realizar o retorno dos valores calculados de K_p e K_i das funções.

Figura 4.21 – Diagrama de blocos para funções de otimização



Fonte: do Autor.

O programa principal que utiliza a função *fmincon* está salientada na Figura 4.22. Neste trecho de código MATLAB, os valores de K_{p_ziegle} e K_{i_ziegle} representam as sementes iniciais derivadas do método de Ziegler-Nichols. Essas sementes são incorporadas em um vetor chamado "valoresZiegle", contendo as sementes iniciais para os parâmetros do controlador (K_p , K_i e K_d ,

esse último inicializado como 0). As otimizações são realizadas pela função *fmincon* em três iterações distintas para cada índice de desempenho: IAE, ITAE e MSE.

É relevante destacar que as funções de custo para cada otimização, ou seja, custoIAE, custoITAE e custoMSE, estão salientadas no Apêndice C. Essas funções são cruciais para definir os critérios de desempenho a serem otimizados, refletindo diferentes considerações no ajuste dos parâmetros do controlador.

Os resultados otimizados para os parâmetros do controlador PID são armazenados nas variáveis 'PminIAE', 'PminITAE' e 'PminMSE'.

Figura 4.22 – Código principal para executar a função *fminsearch*

```
clear all
close all
warning off
options = optimset('display','iter');
global P I D erro time
Kp_ziegle = 0.451
Ki_ziegle = 1.275
valoresZieagle = [Kp_ziegle Ki_ziegle 0]

PminIAE = fmincon('custoIAE', sementes , options)
PminITAE = fmincon('custoITAE', sementes , options)
PminMSE = fmincon('custoMSE', sementes , options)
```

Fonte: do Autor

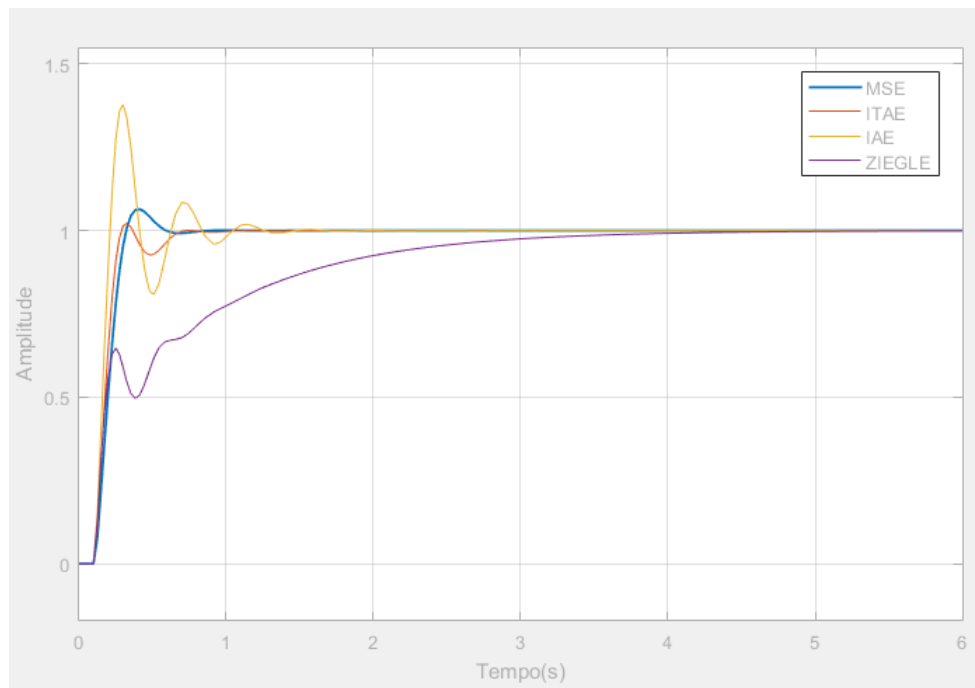
Após a realização das simulações, foram obtidos os resultados dos parâmetros K_p e K_i para as três funções de otimização. Os valores resultantes estão destacados na Tabela 4.1, Além dos parâmetros de ganho, a tabela inclui informações relevantes sobre o desempenho do sistema como tempo de acomodação (t_s), overshoot (M_p), undershoot(u_s).

Tabela 4.1 – Resultados para as funções de otimização

Funções	P	I	T_s (s)	MP(%)	US(%)
MSE	0,240	4,352	0,882	6,81	5,81
IAE	0,367	4,289	0,977	37,43	1,531
ITAE	0,479	6,467	1,693	2,45	11,79
ZIEGLER	0,451	1,275	5,164	0,17	1,812

A Figura 4.23 ilustra a resposta do sistema a uma entrada do tipo degrau unitário, considerando os métodos de sintonia de Ziegler-Nichols e os métodos de otimizações IAE, ITAE e MSE. Essa complementa as informações contidas na Tabela 4.1, contribuindo para uma compreensão mais completa e detalhada do impacto das otimizações nos resultados do sistema de controle.

Figura 4.23 – Resposta ao degrau unitário do sistemas



Fonte: do Autor.

No primeiro método mencionado na Tabela 4.1, no método de Ziegler-Nichols o valor calculado para o tempo de estabelecimento é superior ao comparados com as funções de otimização, nota-se que o sistema se estabiliza mais lentamente ao utilizar o método Ziegler-Nichols, embora,

em relação ao valor de *overshoot*, este método demonstre um desempenho significativamente superior quando comparado aos demais métodos.

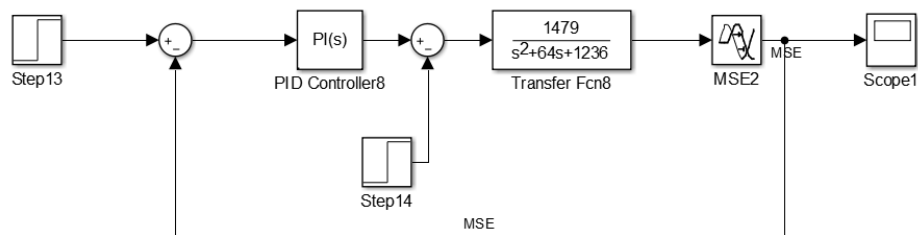
O método utilizando o IAE apresentou o maior *overshoot* dentre todos os métodos, entretanto o tempo de estabelecimento é menor do que o método de Ziegler-Nichols. Os resultados obtidos pelos métodos MSE e ITAE são bastante semelhantes entre si, e, para o resultado em questão, o melhor desempenho para o sistema ocorre quando se utiliza o índice MSE.

Para o sistema descrito pela Equação 4.2, ao empregar o índice de desempenho MSE, apresenta uma considerável melhoria no tempo de estabelecimento e valor máximo de *overshoot*. Dentre os métodos de sintonia utilizados, os resultados indicam que o MSE se destaca como o mais indicado para otimizar o desempenho do sistema devido .

4.5.1 Aplicação de uma perturbação no sistema

Para fortalecer ainda mais a validação das funções de otimização, procedeu-se à introdução de uma perturbação no sistema, conforme representado no diagrama de blocos na Figura 4.24, desenvolvido no ambiente Simulink.

Figura 4.24 – Diagrama de blocos com perturbação



Fonte: do Autor.

Nesse contexto, uma perturbação foi aplicada no sistema no instante de tempo $t_0 = 5$ s e com uma amplitude de 0,3. Esta abordagem foi adotada com o propósito de testar a capacidade de resposta do sistema diante de condições adversas."

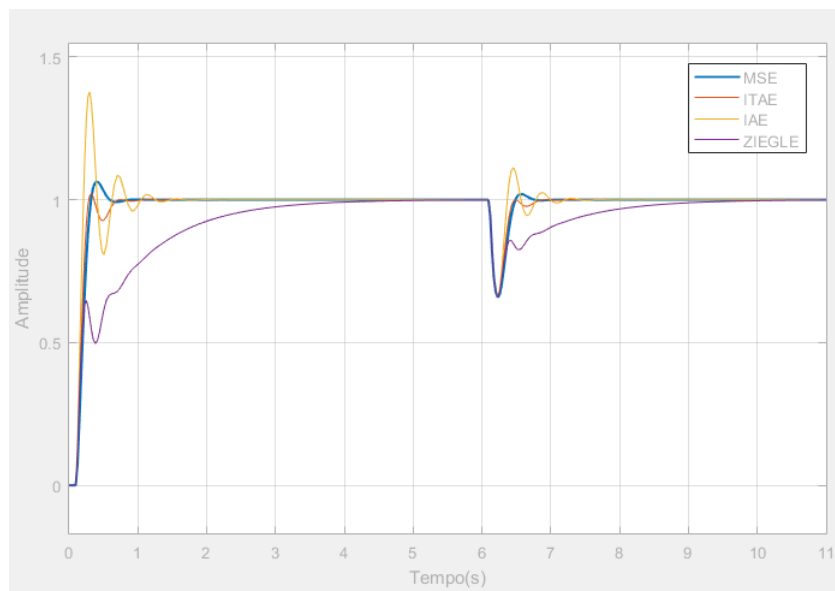
A Tabela 4.2 especifica os ganhos do controlador PID para os três métodos, bem como os valores de T_s , o valor de $MP(\%)$ e $US(\%)$.

Tabela 4.2 – Resultados para as funções de otimização com perturbação

Funções	P	I	T_s (s)	MP(%)	US(%)
MSE	0,240	4,352	1,067	2,33	0,193
IAE	0,367	4,289	1,627	10,41	5,14
ITAE	0,479	6,467	1,017	3,40	2,23
ZIEGLER	0,451	1,275	4,311	0,30	17,34

A Figura 4.25 evidencia a resposta do sistema frente a uma entrada de degrau unitário, considerando o método de Ziegler-Nichols e com as funções de otimização.

Figura 4.25 – Resposta ao degrau do sistema aplicado uma perturbação



Fonte: do Autor.

De acordo com a Tabela 4.2, o método de Ziegler-Nichols revela um valor calculado para o tempo de estabelecimento superior quando comparado às funções de otimização. Evidencia-se que o sistema apresenta uma estabilização mais lenta ao empregar o método de Ziegler-Nichols. Entretanto, em contrapartida, em relação ao valor de *overshoot*, este método demonstra um desempenho significativamente superior em comparação com os demais métodos analisados.

A abordagem utilizando o índice IAE resulta no maior *overshoot* entre todos os métodos considerados. No entanto, é crucial notar que o tempo de estabelecimento é menor em comparação

com o método de Ziegler-Nichols. Os resultados provenientes dos métodos MSE e ITAE mostram uma notável semelhança entre si.

No que diz respeito ao sistema descrito pela Equação 4.2, ao empregar o índice de desempenho MSE, observa-se uma considerável melhoria no tempo de estabelecimento e no valor máximo de *overshoot*. A análise comparativa dos métodos de sintonia sugere que o MSE se destaca como a escolha mais indicada para otimizar eficientemente o desempenho do sistema. Esses resultados fornecem valiosas informações para a seleção criteriosa de métodos de sintonia, contribuindo para uma compreensão mais abrangente do comportamento dinâmico do sistema em diferentes cenários de operação.

4.6 Discretização da função

A fim de viabilizar a implementação no Arduino, foi crucial realizar a discretização da equação no domínio do tempo, representada pela Equação 4.2. Este processo foi efetuado utilizando uma taxa de amostragem desejada, que neste caso, foi configurada como 0,05, o qual representa o intervalo entre as amostras temporais, indicando a periodicidade com que o sistema é avaliado e controlado.

A demonstração para o cálculo da discretização controlador PI pode ser observada :

$$Kp + Ki \frac{1}{s} \quad (4.4)$$

Desta forma, a Equação obtida foi:

$$Kp + Ki * \frac{(0.05 * (z + 1))}{2 * (z - 1)} \quad (4.5)$$

$$\frac{U(z)}{E(z)} = Kp + \frac{Ki * Ts}{2} * \frac{z + 1}{z - 1} \quad (4.6)$$

$$\frac{U(z)}{E(z)} = Kp + \frac{Ki * Ts}{2} * \frac{1 + z^{-1}}{1 - z^{-1}} \quad (4.7)$$

$$\frac{U(z)}{E(z)} = \frac{Kp(1 - z^{-1}) + Ki * \frac{Ki * Ts}{2}}{1 - z^{-1}} \quad (4.8)$$

$$U(z)(1 - z^{-1}) = Kp(1 - z^{-1})E(z) + \frac{Ki * Ts}{2} * (1 + z^{-1})E(z) \quad (4.9)$$

Com isso, foi calculada a transformada Z inversa e foi obtida a Equação de diferenças 4.10.

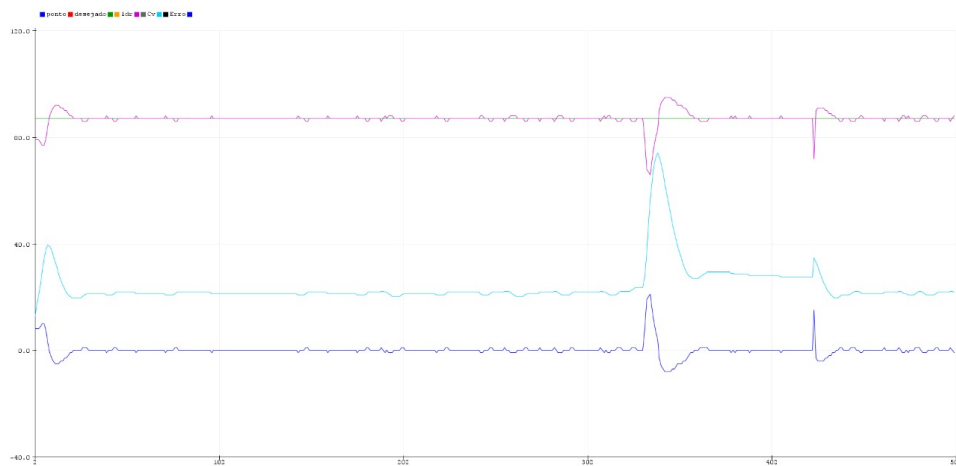
$$u(k) = u(k+1) + Kp(e(k) - e(k-1)) + \frac{Ki * Ts}{2} * (e(k) + e(-1)) \quad (4.10)$$

A implementação do código do controle PI no Arduino está presente no Apêndice B.

4.7 Análise e resultados

A implementação do controle PI no Arduino, utilizando os parâmetros obtidos pelo método de Ziegler-Nichols, teve como objetivo principal analisar e avaliar o desempenho do sistema. Conforme ilustrado no Figura 4.26, as variáveis como erro, PWM do Arduino, *SetPoint* desejado e intensidade de luminosidade são representadas pelas cores azul escuro, azul claro, verde e roxo.

Figura 4.26 – Resposta do sistema para os parâmetros de Ziegler-Nichols



Fonte: Do Autor.

Adicionalmente, foram conduzidos testes de perturbação no sistema, introduzindo um obstáculo próximo ao sensor para parcialmente bloquear a luz da lâmpada até o sensor. Essa ação resultou em uma resposta rápida do sistema, observada pela rápida adaptação do sinal PWM do Arduino.

O valor do PWM aumentou de 32% para 72%, estabilizando posteriormente em 37%. Contudo, é importante destacar que o sistema manifestou pequenos ruídos constantes ao longo do experimento, um aspecto relevante a ser considerado na otimização contínua do controle.

Ao analisar os resultados obtidos da tabela 4.1, destaca-se que a função IAE revelou o maior overshoot em comparação com as funções MSE e ITAE, além de apresentar o maior tempo necessário para atingir a estabilidade.

Contraopondo-se, a função ITA obteve o menor *overshoot* entre as três, embora tenha sido acompanhada pelo maior tempo de estabelecimento.

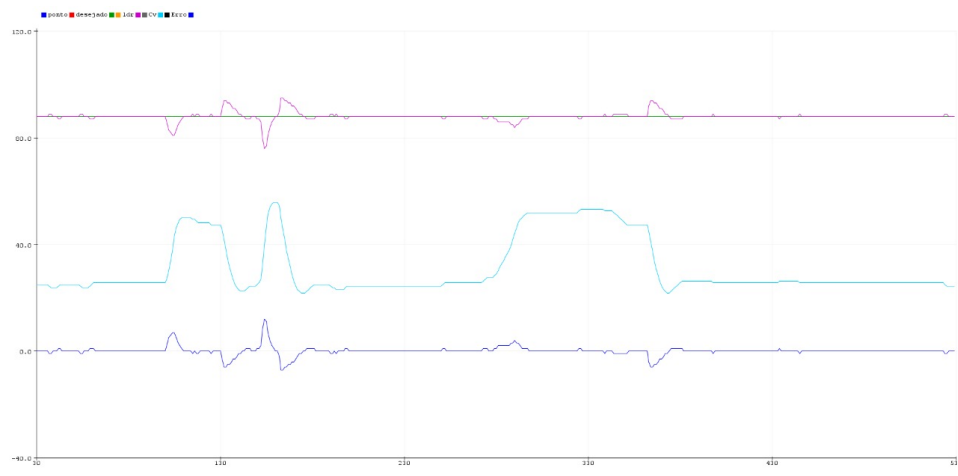
Por sua vez, a função MSE se destacou ao apresentar o menor tempo de estabelecimento, entretanto apresentou valor de *overshoot* aceitável, evidenciando um desempenho mais equilibrado em relação aos índices de desempenho analisados.

Diante dessas considerações, a função MSE emergiu como a escolha mais apropriada para a implementação prática. Sua capacidade de proporcionar um compromisso eficiente entre overshoot e tempo de acomodação, a torna uma opção preferencial, alinhada aos objetivos específicos de controle e desempenho do sistema em questão.

Os parâmetros K_p e K_i , derivados da função de otimização do MSE, foram aplicados no controle do sistema e submetidos aos mesmos testes realizados com o modelo de Ziegler-Nichols. O resultado desses experimentos pode ser visualizado no gráfico presente na Figura 4.27.

Notavelmente, a resposta do sistema demonstrou ser mais rápida em comparação com a abordagem de Ziegler-Nichols.

Figura 4.27 – Resposta do sistema para os parâmetros de MSE



Fonte: Do Autor.

Além disso, foi observado que o valor de PWM necessário para estabilizar o sistema foi inferior ao obtido com os parâmetros de Ziegler-Nichols. Os erros em regime permanente foram menos impactados por resíduos, resultando em uma menor oscilação na intensidade luminosa da lâmpada.

Esses achados indicam uma melhoria na eficiência do controle ao utilizar os parâmetros otimizados pela função de MSE, proporcionando uma resposta mais ágil e uma redução significativa dos ruídos presentes no sistema.

Conforme antecipado, todos os métodos examinados neste estudo demonstraram desempenho superior em comparação ao método de sintonia de Ziegler-Nichols. No entanto, é crucial notar que essa comparação pode não ser completamente justa, uma vez que a configuração desses métodos requer um conhecimento prévio do modelo do processo, ao contrário da heurística de Z-N, que não exige tal conhecimento. A escolha pelo método de Z-N foi guiada pela facilidade de obtenção de resultados iniciais, possibilitando utilizá-los como ponto de partida para otimizar o desempenho dos sistemas sob análise.

Assim, o estudo enfatiza a importância contínua dos métodos tradicionais de sintonia, como as Ziegler-Nichols. No entanto, destaca-se que os profissionais responsáveis pelo controle têm a opção de considerar as funções de otimizações como uma alternativa viável para sintonizar processos nos quais os resultados dos métodos tradicionais não atendem adequadamente aos critérios de desempenho desejados.

5 CONCLUSÃO

O presente trabalho proporcionou uma investigação sobre o desenvolvimento e implementação de um sistema de controle para a regulação da intensidade luminosa em ambientes agrícolas.

O projeto, centrado no uso de um controlador PI, foi guiado pelo propósito de aprimorar a eficiência energética e otimizar o ambiente para galinhas poedeiras, considerando a importância da luz em seu desenvolvimento.

Durante a fase de modelagem e otimização, foram exploradas diferentes abordagens, destacando-se a utilização dos parâmetros obtidos pelos métodos de Ziegler-Nichols e otimização pelos métodos ITAE, IAE e MSE. Os resultados obtidos com ambas as estratégias foram cruciais para a compreensão do comportamento do sistema e sua capacidade de resposta aos estímulos.

A análise comparativa revelou que a otimização pelo MSE proporcionou uma resposta mais ágil do sistema, com menor oscilação e um controle mais eficiente em comparação ao método clássico de Ziegler-Nichols.

Ao longo dos testes práticos, o sistema demonstrou sua adaptabilidade a perturbações, respondendo de maneira eficaz a variações na luminosidade ambiental. A introdução de obstáculos para simular situações adversas resultou em ajustes automáticos do sistema, evidenciando a capacidade do controlador PI em manter a estabilidade operacional.

Os objetivos propostos inicialmente foram plenamente alcançados, com o sistema apresentando um desempenho notável na regulação da intensidade luminosa.

A abordagem de controle PI, aliada à otimização pelo MSE, não apenas cumpriu os requisitos estabelecidos, mas, também, ofereceu melhorias significativas em relação à estratégia tradicional de Ziegler-Nichols.

Entretanto, é essencial ressaltar que o caminho até o sucesso não esteve isento de desafios. Uma das principais dificuldades enfrentadas foi na fase inicial de modelagem do sistema, na qual a complexidade intrínseca das interações entre os componentes demandou uma análise minuciosa e refinamentos iterativos.

Outra questão significativa foi a presença de pequenos ruídos constantes nos dados coletados, o que impactou as leituras do sensor LDR e, conseqüentemente, a estabilidade do controle.

A minimização desses ruídos tornou-se um ponto crítico durante a implementação prática, envolvendo ajustes finos nos parâmetros do controlador PI para mitigar essas interferências indesejadas.

Essa etapa destacou a importância da adaptabilidade do sistema diante de condições dinâmicas e, ainda, ressaltou a necessidade contínua de refinamentos para garantir um controle eficaz em cenários reais.

Este trabalho contribui não apenas para a criação das galinhas poedeira, mas, também, para a compreensão mais aprofundada da aplicação de controle de luminosidade em ambientes específicos, abrindo portas para futuras pesquisas e implementações práticas.

Em síntese, as dificuldades encontradas durante o desenvolvimento do projeto não apenas realçaram os desafios inerentes à automação agrícola, como também proporcionaram valiosas lições sobre a importância da flexibilidade e adaptação diante das complexidades inerentes aos ambientes reais. O aprendizado obtido contribui para a conclusão bem-sucedida deste projeto e, além disso, para aprimoramentos futuros e a contínua evolução das estratégias de controle de luminosidade em sistemas avícolas.

Quanto às melhorias sugeridas para o projeto, destacam-se as seguintes perspectivas para aprimoramentos futuros:

- Ampliação do Sistema com mais Sensores LDR e Lâmpadas: considerar a inclusão de uma rede mais extensa de sensores LDR distribuídos pelo ambiente, juntamente com lâmpadas estrategicamente posicionadas. Essa expansão permitiria uma cobertura mais abrangente, possibilitando o monitoramento e controle preciso da luminosidade em diferentes áreas do ambiente;
- Implementação de Controle Multivariável: explorar a aplicação de técnicas de controle multivariável para otimizar simultaneamente vários parâmetros do sistema. Essa abordagem poderia levar a uma adaptação mais eficiente às complexas interações entre os diversos componentes, resultando em um controle mais refinado e adaptável às variáveis dinâmicas do ambiente;
- Integração de alimentação da rede de distribuição em corrente alternada: avaliar a viabilidade de utilizar a rede de distribuição elétrica, fornecendo corrente alternada, como fonte

de alimentação principal para o sistema. Essa integração poderia simplificar a infraestrutura elétrica, oferecendo uma alternativa eficiente para a alimentação do sistema e facilitando a implementação em ambientes que já contam com infraestrutura elétrica consolidada.

Essas melhorias propostas visam aprimorar a eficiência, a cobertura e a versatilidade do sistema, promovendo uma gestão mais sofisticada da iluminação . A implementação dessas sugestões não apenas ampliaria as capacidades do sistema atual, mas também abriria novas oportunidades para uma automação mais avançada e adaptável às demandas específicas de diferentes cultivos.

REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à identificação de sistemas: técnicas lineares e não lineares: teoria e aplicação.** [S.l.: s.n.], 2015. v. 1. 4 p.
- ANAND, S.; RAJAGOPALAN, S.; MOHAN, R. Retracted: Management of liver hydatid cysts—current perspectives. **Medical journal armed forces India**, Elsevier, v. 68, n. 3, p. 304–309, 2012.
- ANUAL, A. R. **Associação Brasileira de Proteína Animal-ABPA.** 2022.
- ASTROM, K. J. **Pid controllers: theory, design, and tuning.** **The International Society of Measurement and Control**, 1995.
- DUTTON, K.; THOMPSON, S.; BARRACLOUGH, B. **The art of control engineering.** [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1997.
- GARCIA, C. **Modelagem e Simulação de Processos Industriais e de Sistemas Eletromecânicos Vol. 1.** [S.l.]: Edusp, 2005.
- HART, N. S.; PARTRIDGE, J.; CUTHILL, I. Visual pigments, cone oil droplets, ocular media and predicted spectral sensitivity in the domestic turkey (meleagris gallopavo). **Vision Research**, Elsevier, v. 39, n. 20, p. 3321–3328, 1999.
- JORDAN, R. A.; TAVARES, M. H. Análise de diferentes sistemas de iluminação para aviários de produção de ovos férteis. **Revista Brasileira de Engenharia Agrícola e Ambiental**, SciELO Brasil, v. 9, p. 420–423, 2005.
- LEWIS, P.; MORRIS, T. Poultry and coloured light. **World's Poultry Science Journal**, Cambridge University Press on behalf of World's Poultry Science Association, v. 56, n. 3, p. 189–207, 2000.
- MENDES, A. S. et al. Performance and preference of broiler chickens exposed to different lighting sources. **Journal of Applied Poultry Research**, Elsevier, v. 22, n. 1, p. 62–70, 2013.
- NASCIMENTO, J. F. d. A criação de galinhas caipiras pelas mulheres do tabuleiro de muqueim–areia-pb. Universidade Federal da Paraíba, 2017.
- OGATA, K. **Modern control engineering fifth edition.** [S.l.: s.n.], 2010.
- OLIVEIRA, B. Importância do manejo na produção de ovos de codornas. **Simpósio internacional de Coturnicultura**, v. 2, p. 91–96, 2004.
- OLIVEIRA, B. d. Manejo racional e produtividade das codornas. **Simpósio Internacional de Coturnicultura**, v. 1, p. 133–145, 2002.
- PADOVAN, A. **Programa de luz em granjas de poedeiras comerciais.** [S.l.], 2009. Disponível em: <http://www.apavi.com.br/index.php?pag=conteudo&id_conteudo=3472&idmenu=165>.

RAKE, H. Step response and frequency response methods. **IFAC Proceedings Volumes**, Elsevier, v. 12, n. 8, p. 519–526, 1979.

SIVANANDAM, S. et al. Genetic algorithm implementation using matlab. **Introduction to genetic algorithms**, Springer, p. 211–262, 2020.

SOUZA, L. F. M. Programas de luz para codornas de postura. 2015.

APÊNDICE A – Código de aquisição de Dados

```
#define AnalogLDR A1
#define BrilhoMax 1024
#define BrilhoMin 0
#define PWM 3
int pot;
int out;
int Brilho = 0;
int Leitura = 0;
int LeituraAnterior = 0;
void setup()
{
  Serial.begin(9600);
  pinMode(PWM, OUTPUT);
  Serial.println("comecando.....");
  delay(5000);
}
void loop()
{
  delay(50);
  Leitura = analogRead(AnalogLDR); //L o valor fornecido pelo LDR
  pot=analogRead(A2);
  out=map(pot,0,1023,0,255);
  analogWrite(PWM,255);
  Serial.print("Leitura sensor LDR = ");
  Brilho = map(Leitura, BrilhoMin, BrilhoMax, 0, 100);
  Serial.println(Brilho);
}
```


APÊNDICE B – Código do Controle de Luminosidade

```

#define AnalogLDR A1
#define BrilhoMax 1024
#define BrilhoMin 0
#define PWM 3
float pv = 0.0; //variavel de processo medida (porcentagem de
luminosidade)
float sp = 87.0; //ponto de ajuste desejado (porcentagem de
luminosidade)
float cv = 0.0;
int pwm = 0;
float error = 0.0; // error atual
float error_1 = 0.0; //erro no instante anterior
float kp = 0.4799;
float ki = 6.4673;
int t = 50;
unsigned long pasado = 0;
unsigned long actual;
void setup() {
  Serial.begin(9600);
}
void loop() {
  actual = millis(); // milisegundos desde que se iniciou o programa;
  unsigned long dt = actual - pasado;
  if(dt >= 1)
  {
    pasado = actual;
    pv = map(analogRead(AnalogLDR), BrilhoMin, BrilhoMax, 0, 100);
    pv = constrain(pv, 0, 100);
    error = sp - pv;
    cv = cv + kp * (error - error_1) + (ki * 0.05 * 0.5) * (error + error_1);
    cv = constrain(cv, 0, 100); // restricao da acao de controle
    pwm = map(cv, 0, 100, 0, 255);
    analogWrite(PWM, pwm);

  }
}

```

APÊNDICE C – Código das funções de otimizações

```

%% IAE
function [error] = custoIAE(x)
global P I D erro time
P=x(1);
I=x(2);
D=x(3);
%Integral do Erro Absoluto (IAE)
time=0:0.1:20;
[T X Y]=sim('dp_pid',time);
error = abs(erro);

%% ITAE
function [error] = custoITAE(x)
global P I D erro time
P=x(1);
I=x(2);
D=x(3);
% Integral do Erro Absoluto ponderada pelo Tempo (ITAE)
time=0:0.1:20;
[T X Y]=sim('dp_pid',time);
error=abs(erro).*T;

%% MSE
% M dia do Erro Quadr tico (MSE)
function [error] = custoMSE(x)
global P I D erro time
P=x(1);
I=x(2);
D=x(3);
time=0:0.1:20;
N=length(time);
[T X Y]=sim('dp_pid',time);
error = erro; error = error.^2;

```