

**UNIVERSIDADE FEDERAL DE ALAGOAS
UNIDADE ACADÊMICA CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA QUÍMICA**

LARISSA YASMIM ALVES BATISTA

**DESENVOLVIMENTO DE UM CONTROLADOR PID NA PLATAFORMA DE
PROTOTIPAGEM ARDUINO PARA CONTROLE DE VAZÃO DE UM TANQUE**

Maceió
2024

LARISSA YASMIM ALVES BATISTA

**DESENVOLVIMENTO DE UM CONTROLADOR PID NA PLATAFORMA DE
PROTOTIPAGEM ARDUINO PARA CONTROLE DE VAZÃO DE UM TANQUE**

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Química da Universidade Federal de Alagoas como requisito parcial para obtenção do título de Bacharel em Engenharia Química.

Orientador/a: Prof. Dr. Júlio Inacio Holanda Tavares Neto

Coorientador/a: Prof. Dr. João Inácio Soletti

Maceió
2024

Catálogo na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 – 1767

B333d Batista, Larissa Yasmim Alves.
 Desenvolvimento de um controlador PID na plataforma de prototipagem
 Arduino para controle de vazão de um tanque / Larissa Yasmim Alves
 Batista. – 2024.
 63 f. il. : figs. ; tabs. color.

Orientador: Júlio Inacio Holanda Tavares Neto.
Co-orientador: João Inácio Soletti.
Monografia (Trabalho de Conclusão de Curso em Engenharia Química).
Universidade Federal de Alagoas. Centro de Tecnologia. Maceió, 2024.

Bibliografia: f. 60-61.
Apêndices: f. 62-63.

1. Automação. 2. Instrumentação. 3. Arduino (Controlador programável). I.
Título.

CDU:66.0:004

AGRADECIMENTOS

A minha mãe que sempre foi minha base moral, a pessoa que mais me apoiou e acreditou em mim, tudo que eu já fiz e ainda farei eu devo a ela. Aos meus avós que torcem por mim e comemoram minhas vitórias como se fossem as deles, o amor deles me formou como pessoa e como profissional. Ao meu pai, aos meus irmãos e a toda a minha enorme família, agradeço pela força, companheirismo e amor que eu sempre recebi de sobra de todos vocês.

Ao meu orientador Prof. Dr. Júlio Inacio Holanda Tavares Neto, por acreditar em mim durante todo o desenvolvimento do trabalho, por embarcar nas minhas ideias e por todo o suporte e conhecimento que eu recebi. Ao prof. Dr. Leonardo Faustino Lacerda de Souza e aos alunos de engenharia elétrica, Carlos Eduardo e Alisson Farias, pelo suporte com toda a parte eletrônica deste trabalho e por responder as minhas diversas dúvidas sobre todo o processo.

Ao meu coorientador Prof. Dr. João Inácio Soletti e ao Prof. Dr. Rodolfo Junqueira Brandão, pelos ensinamentos durante a graduação e pelo apoio durante a realização deste trabalho.

A toda equipe do LASSOP e aos departamentos de Engenharia Elétrica e Engenharia de Energia do CECA pela disponibilidade dos equipamentos utilizados na realização deste trabalho.

Ao meu orientador de iniciação de científica prof. Dr. Andre Luiz Lins de Aquino e a toda equipe do LACCAN, pela confiança, por enormes aprendizados, por me tornar uma profissional mais completa e por todo o café.

Aos amigos que tenho desde sempre (Glória e Igor) por manterem minha sanidade e me apoiarem sempre.

Aos amigos que a UFAL me deu (Wanessa, Ana Beatriz, Sarah, Andressa, Patrick, Carlos, Matheus, Jadde, Lucas) e a todos os quais eu conheci durante a graduação, vocês tornaram essa jornada muito mais feliz.

“Somos irmãos naquilo que compartilhamos. Você não tem nada, você não possui nada. Você é livre. Tudo o que você tem é o que você é e o que você oferece.”

— Ursula K. Le Guin, The Dispossessed.

RESUMO

Sistemas de controle e automação são fundamentais na operação e competitividade das diversas indústrias químicas. Tais sistemas fornecem meios para monitoramento, regulação e otimização dos processos industriais através da coleta e manipulação de variáveis de processo; tornando o processo produtivo mais previsível, consistente, seguro e menos propenso a falhas. Entre esses sistemas de controle, o controlador PID (Proporcional, Integral e Derivativo) é o mais comumente utilizado para o controle de sistemas dinâmicos. Este sistema se baseia na combinação entre três termos ajustáveis que atuam sobre o valor de erro (diferença entre um valor medido e um valor desejado) através dos coeficientes ajustáveis para gerar uma ação de controle sobre uma variável manipulada, a qual é enviada ao sistema para ajustar uma ou mais variáveis controladas. O ajuste adequado desses coeficientes é fundamental para obter um desempenho ótimo do controlador PID, com resposta rápida, estabilidade e supressão de oscilações indesejadas. Todavia, sistemas de controle complexos são custosos e exigem alta adaptação do processo para os comportar. Com base nisso, o presente trabalho visou a implementação de um sistema de controle PID de vazão cujo algoritmo foi implementado na plataforma de prototipagem Arduino. A estratégia de controle utilizada visou o controle da vazão de alimentação de um sistema eletrolítico através da manipulação da tensão contínua de uma bomba. Esta implementação foi dada em três partes: inicialmente foi construído um sistema supervisor com uma interface gráfica amigável feita utilizando Python para possibilitar o monitoramento das variáveis do sistema assim como a interação do operador com o mesmo; logo após a dinâmica entre tensão e vazão foi analisada através da manipulação direta da tensão e registro da resposta da vazão ao longo do tempo. Assim, os dados registrados foram submetidos a dois modelos de regressão (linear e polinomial) de forma a obter mais informações sobre o sistema. Em seguida, os dados de vazão e tensão ao longo do tempo foram submetidos à função PROCEST do Matlab, a qual estimou uma função de transferência para o processo usando os dados inseridos. Nesta ferramenta, diversos modelos para o sistema foram calculados e comparados com base nos seus valores de erro e ajuste aos dados. Por fim, a função de transferência mais adequada (uma função de primeira ordem com um zero à esquerda do eixo real) foi inserida tanto em uma malha aberta quanto em uma malha fechada (ambas simuladas utilizando a ferramenta de simulação chamada SIMULINK) e submetida a quatro métodos de sintonia clássicos (Ziegler-Nichols de , Cohen-Coon e Tentativa e Erro), cujos desempenhos foram comparados através da observação de suas respectivas respostas a uma perturbação degrau inserida na entrada do sistema no início da simulação. Observou-se resultados analogamente positivos para a sintonia por Ziegler-Nichols em malha aberta e por Tentativa e Erro, ambos com erro do estado estacionário quantitativamente irrelevante e rápido tempo de ascensão e acomodação. Os demais métodos foram identificados como inadequados para o sistema identificado. De forma geral, a implementação foi bem sucedida e o sistema se adaptou tanto ao controle PID quanto ao controle PI.

Palavras-Chave: Controle, automação, instrumentação, Arduino

ABSTRACT

Control and automation systems are crucial in the operation and competitiveness of various chemical industries. These systems provide means for monitoring, regulation, and optimization of industrial processes through the collection and manipulation of process variables, making the production process more predictable, consistent, safe, and less prone to failures. Among these control systems, the PID controller (Proportional, Integral, and Derivative) is the most commonly used for dynamic systems control. This system relies on the combination of three adjustable terms that act on the error value (difference between a measured value and a desired value) through adjustable coefficients to generate a control action on a manipulated variable, which is sent to the system to adjust one or more controlled variables. Proper adjustment of these coefficients is crucial to achieve optimal PID controller performance with quick response, stability, and suppression of unwanted oscillations. However, complex control systems are costly and require high process adaptation to accommodate them. Based on this, the present work aimed at implementing a flow PID control system, whose algorithm was implemented on the Arduino prototyping platform. The control strategy used aimed at controlling the feed flow of an electrolytic system by manipulating the continuous voltage of a pump. This implementation was carried out in three parts: initially, a supervisory system with a user-friendly graphical interface was built using Python to enable monitoring of system variables and operator interaction; then the dynamics between voltage and flow were analyzed through direct manipulation of voltage and recording of flow response over time. The recorded data were then subjected to two regression models (linear and polynomial) to obtain more information about the system.

Next, the flow and voltage data over time were submitted to the PROCEST function in Matlab, which estimated a transfer function for the process using the input data. In this tool, various models for the system were calculated and compared based on their error values and fit to the data. Finally, the most suitable transfer function (a first-order function with a zero to the left of the real axis) was inserted into both an open-loop and a closed-loop system (both simulated using the SIMULINK simulation tool) and subjected to four classical tuning methods (Ziegler-Nichols, Cohen-Coon, and Trial and Error), whose performances were compared by observing their respective responses to a step disturbance inserted at the system input at the beginning of the simulation. Results showed similarly positive outcomes for Ziegler-Nichols tuning in open-loop and Trial and Error tuning, both with quantitatively irrelevant steady-state error and rapid rise and settling times. The remaining methods were identified as inadequate for the identified system. Overall, the implementation was successful, and the system adapted well to both PID and PI control.

Keywords: Control, automation, instrumentation, Arduino

1. LISTA DE SÍMBOLOS

t	Tempo
K_p	Ganho Proporcional
$e(t)$	Erro em função do tempo
T_D	Tempo Derivativo
T_I	Tempo Integral
u_o	Constante de resposta do sistema no início do sistema
SP	“Set Point” (valor de referência)
A, C	Amplitude das curvas senoidais
T	Constante de tempo
L	Tempo morto
T_R	Tempo de Subida
T_S	Tempo de Assentamento
T_Z	Constante de tempo zero
T_P	Constante de tempo do polo
$NaCl$	Cloreto de Sódio
H_2O	Água
$NaOCl$	Hipoclorito de Sódio
H_2	Hidrogênio
δ	Delta de Dirac
σ	Função de Heaviside
\mathcal{L}	Transformada de Laplace
j	Parte imaginária de um número complexo
$u(t)$	Entrada de um sistema no domínio do tempo

$r(t)$	Saída de um sistema no domínio do tempo
$U(s)$	Entrada de um sistema no domínio da frequência
$R(s)$	Saída de um sistema no domínio da frequência
$G(s)$	Função de transferência de um sistema
u_0	Amplitude do distúrbio do sistema
$r(\infty)$	Resposta final do sistema
ζ	Razão de amortecimento
ω_n	Frequência natural
ω_d	Frequência natural amortecida
γ	Coefficiente de amortecimento
V_L	Tensão presente no indutor
V_S	Tensão de saída

LISTA DE ILUSTRAÇÕES

FIGURAS

Figura 1 - Diagrama de um sistema de controle em malha aberta	15
Figura 2 -Diagrama de um sistema de controle de malha fechada	15
Figura 3 - Exemplo de estabilidade na analogia da rampa de skate	18
Figura 4 - Diagrama representativo da estabilidade das raízes típicas de uma equação característica	18
Figura 5 - Procedimento Básico para Modelagem Matemática	20
Figura 6 - Representação gráfica da relação entre as funções degrau e impulso	24
Figura 7 - Resposta de um sistema de 1º ordem a uma perturbação degrau	27
Figura 8 - Resposta de um sistema de 2º ordem a uma perturbação degrau	29
Figura 9 - Resposta dinâmica de uma malha de controle PID clássica	30
Figura 10 - Resposta ao degrau unitário de um processo usada para determinar os parâmetros do processo	31
Figura 11 - Componentes de um Arduino UNO clássico	33
Figura 12 - Exemplo de sinais PWM gerados no Arduino	35
Figura 13 - Representação gráfica de um Conversor Buck	36
Figura 14 - Conversor Buck com MOSFET ligado e diodo desligado	37
Figura 15 - Conversor Buck com MOSFET desligado e diodo ligado	37
Figura 16 - Visão geral da arquitetura do sistema supervisorio	38
Figura 17 - Tela inicial do sistema supervisorio	39
Figura 18 - Tela de monitoramento monitorando concentração de Hidrogênio	39
Figura 19 - Tela de alteração do sistema supervisorio	40
Figura 20 - Representação gráfica do algoritmo do sistema auxiliar	42
Figura 21 - Comunicação de leitura de sensores entre o Arduino e o computador	43
Figura 22 - Comunicação de alteração de vazão entre o computador, o Arduino e o sistema	43
Figura 23 - Disposição do sistema físico durante a medição	44
Figura 24 - Identificação da função de transferência usando PROCEST	46
Figura 25 - Simulação do sistema de bombeamento em malha aberta	46
Figura 26 - Simulação do sistema de bombeamento em malha fechada	47
Figura 27 - Simulação do sistema completo em malha aberta	47
Figura 28 - Simulação do sistema completo em malha fechada	48
Figura 29 - Método de Ziegler-Nichols em Malha Fechada	49

Figura 30 - Regressão linear da relação entre vazão e tensão	50
Figura 31 - Regressão polinomial de segunda ordem da relação entre vazão e tensão	51
Figura 32 - Localização dos polos e zeros da função de transferência escolhida	53
Figura 33 - Resposta ao degrau do sistema simplificado sintonizado Ziegler-Nichols (malha aberta)	54
Figura 34 - Resposta ao degrau do sistema simplificado sintonizado por Cohen-Coon	55
Figura 35 - Resposta ao degrau do sistema completo sintonizado Ziegler-Nichols (malha aberta)	55
Figura 36 - Resposta ao degrau do sistema completo sintonizado por Cohen-Coon	56
Figura 37 - Resposta ao degrau do sistema simplificado sintonizado por Tentativa e Erro.....	57
Figura 38 - Resposta ao degrau do sistema completo sintonizado por Tentativa e Erro.....	58

TABELAS

Tabela 1 - Coeficientes do controlador PID para os métodos de Ziegler-Nichols e Cohen-Coon em malha aberta.....	48
Tabela 2 - Medições de tensão e vazão no sistema	50
Tabela 3 - Condições das funções de transferência estimadas	52
Tabela 4 – Parâmetros de processo após a resposta a uma perturbação degrau.....	53
Tabela 5 - Parâmetros de um controlador PID para o sistema simplificado	54
Tabela 6 - Parâmetros de um controlador PID para o sistema completo	54
Tabela 7 - Coeficientes das ações de um PID dados por Tentativa e Erro para o sistema simplificado	57
Tabela 8 - Coeficientes das ações de um PID dados por Tentativa e Erro para o sistema completo	58

2. SUMÁRIO

1.	INTRODUÇÃO	11
1.1	Controle e Automação.....	11
1.2	Processo Eletrolítico.....	12
2.	OBJETIVOS.....	13
2.1	Geral.....	13
2.2	Específicos	13
3.	REVISÃO BIBLIOGRÁFICA.....	14
3.1	Sistemas de Controle.....	14
3.2	Sistemas de Controle em Malha Aberta.....	14
3.3	Sistemas de Controle em Malha Fechada	15
3.4	Análise de Sistemas de Controle.....	16
3.4.1	Resposta Transitória do Sistema	16
3.4.2	Estabilidade do Sistema	17
3.5	Modelagem Matemática de um Sistema	18
3.5.1	Modelagem Fenomenológica e Experimental do Sistema	19
3.5.2	Modelagem Empírica para Sistemas Dinâmicos	21
3.5.3	Linearidade dos Sistemas.....	22
3.5.4	Sinais de Teste.....	23
3.6	Determinação de Parâmetros Característicos.....	24
3.7	Algoritmo de Controle	29
3.7.1	Controladores do tipo PID	30
3.8	Hardware para construção de malhas de controle.....	32
4.	METODOLOGIA	38
4.1	Sistema Supervisório.....	38
4.1.1	Interface Gráfica	38
4.1.2	Sistema Auxiliar.....	41
4.1.3	Código Integrado no Arduino	42
4.2	Modelagem Matemática.....	44
4.3	Sintonia do Controlador	46
5.	RESULTADOS E DISCUSSÃO	50
6.	CONCLUSÃO	59
	REFERÊNCIAS	60
	APÊNDICES	62

1. INTRODUÇÃO

2.1 Controle e Automação

Sistemas de controle e automação são fundamentais na competitividade das indústrias químicas modernas. Segundo Campos e Teixeira (2006), a implementação destes sistemas acarreta no aumento proveitoso dos níveis de qualidade do produto final, da segurança da unidade produtiva e da confiabilidade dos sistemas físicos e dos equipamentos. Ademais, ao garantir que a produção ocorra sob condições controladas de maior eficiência, é reduzida a necessidade de reprocessamento, o que influencia diretamente na redução de consumo energético e de custos operacionais da unidade.

Dentre as configurações de sistemas de controle possíveis, os sistemas de malha fechada são os mais eficientes para o controle de processos em que a precisão de controle e uso de automação apresentam alto custo-benefício para o processo.

De forma geral, estes sistemas de malha fechada têm seu processo de controle baseado no cálculo do erro, o qual pode ser definido como a diferença entre o valor desejado (*set point*) e o valor da medição durante a captação do sinal da variável controlada. Com base na magnitude e no sinal desse erro, o controlador gera um sinal de ajuste que é enviado para um elemento final de controle, o qual age sobre uma variável manipulada e leva o processo à condição desejada.

Há uma ampla diversidade de controladores com padrões de ações diferentes disponíveis para sistemas de malha fechada, entre eles se destacam os controladores: Liga-Desliga, proporcionais, proporcionais-integrais (PI), proporcionais-derivativos (PD) e proporcionais-integrais-derivativos (PID).

Algumas desvantagens comuns de sistemas de malha fechada são seu alto custo de instalação e a tendência de que a ação de controle automático dada pelo sinal do controlador cause oscilações no processo e dificulte sua estabilização.

Abordando o problema do alto custo, o presente trabalho propõe a criação de um sistema de controle de vazão de baixo custo utilizando a plataforma de prototipagem Arduino. Devido ao baixo custo e baixa especificidade da placa de desenvolvimento e do microcontrolador que compõem a plataforma, ela se torna consideravelmente mais acessível e flexível do que os dispositivos industriais de controle e monitoramento tradicionais, especialmente para projetos de menor escala.

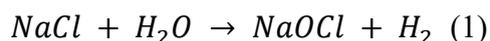
Já em relação à instabilidade, o presente trabalho visa a exploração dos métodos de sintonia disponíveis na literatura de forma a alcançar os parâmetros de controle de melhor

adaptação a um processo eletrólito de produção de hipoclorito de sódio (NaClO) e hidrogênio gasoso.

2.2 Processo Eletrolítico

A geração de desinfetantes “in-loco” é um processo eletrolítico que possui diversas aplicações industriais e domésticas como, por exemplo, a geração de desinfetantes para a eliminação de patógenos em sistemas de distribuição de água. Essa tecnologia se baseia na introdução de salmoura em uma célula eletrolítica, a qual, é prontamente submetida à passagem de corrente elétrica. Com isso, é produzida uma solução oxidante junto com gás hidrogênio o qual, nos sistemas tradicionais, é exaurido do sistema através de uma saída de ventilação (BOAL, 2009).

Neste sistema, quando é aplicada uma voltagem à célula eletrolítica, a qual está em contato com a solução de salmoura, a seguinte reação de oxidação (Equação 1) ocorre na superfície de ambos os eletrodos presentes na célula:



Assim, o presente trabalho visa a implementação de um controlador do tipo PID para o controle da vazão de recirculação de salmoura na célula eletrolítica de um sistema em semi batelada de geração “in-loco” de hipoclorito de sódio e hidrogênio gasoso de forma a possibilitar o bom desempenho do processo sob condições de fluxo diferentes.

3. OBJETIVOS

3.1 Geral

Desenvolver um sistema de controle PID utilizando a plataforma Arduino de forma a possibilitar o controle da vazão de circulação em uma célula eletrolítica e, assim, viabilizar uma produção eficiente de hidrogênio gasoso e hipoclorito de sódio.

3.2 Específicos

- ✓ Construir uma interface gráfica capaz de capturar as medições das variáveis monitoradas por sensores específicos implementados no sistema em questão, sendo elas: Temperatura da solução na célula eletrolítica, corrente e tensão elétrica referentes ao bombeamento da solução, concentração de hidrogênio exaurido e vazão de circulação da solução;
- ✓ Adaptar a mesma interface para o envio de um sinal digital para o Arduino de forma que este manipule a tensão da bomba e, conseqüentemente, altere a vazão de circulação da solução;
- ✓ Coletar dados da resposta (vazão de circulação) do sistema em malha aberta durante um conjunto de perturbações experimentais (manipulação da tensão da bomba), de forma a obter um modelo matemática empírico acurado para o mesmo;
- ✓ Identificar a função de transferência que melhor descreva o sistema;
- ✓ Comparar o desempenho do sistema sob diferentes conjuntos de parâmetros de controlador PID obtidos para o sistema em questão por meio de algoritmos clássicos de sintonia;
- ✓ Integrar o algoritmo de controle do tipo PID ao sistema criado;

4. REVISÃO BIBLIOGRÁFICA

4.1 Sistemas de Controle

Um sistema de controle consiste da combinação de uma ou mais malhas de forma a se obter uma variável de saída de um processo (variável controlada) sob um valor desejado, dado uma outra variável de entrada especificada (variável manipulada). Isto é, sistemas de controle têm por objetivo principal a manutenção de uma variável de processo a fim de que a mesma seja mantida em um valor desejado a despeito das perturbações que podem afetar os processos de interesse (NISE, 2011). Estes sistemas automáticos são tradicionalmente categorizados pela sua configuração característica, podendo esta ser aberta ou fechada.

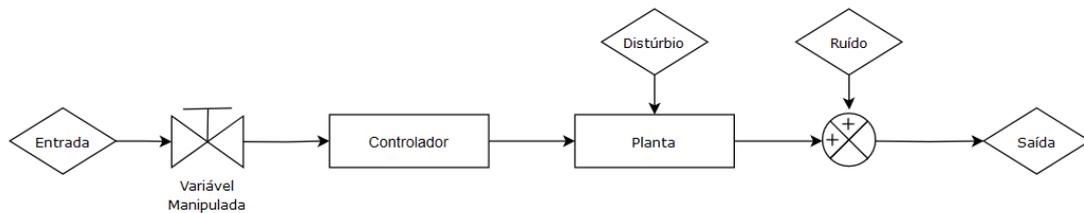
4.2 Sistemas de Controle em Malha Aberta

Segundo Ogata (1977), sistemas de malha aberta são sistemas nos quais a variável de saída não afeta a ação de controle. Dessa forma, o bom desempenho desses sistemas depende da calibração assertiva da variável de entrada. Nesta configuração de sistema, o controlador é ajustado com base no conhecimento de padrões de comportamento do sistema em questão.

Exemplos comuns desses comportamentos são: os padrões de fluxo e perda de carga descritos nas curvas de calibração para o controle de fluídos ou, no exemplo clássico de uma máquina de lavar roupas, no qual apenas a decisão do usuário acerca da necessidade de tempo de lavagem das suas roupas define a operação da máquina sem ajustes automáticos relacionados ao resultado final da lavagem e possível perturbações durante a operação. Assim, após a calibração inicial, a planta pode ser operada sem a necessidade de medições ou ações de controle adicionais.

Devido à sua simplicidade, sistemas de controle de malha aberta costumam apresentar baixo custo de instalação e de operação, entretanto, eles são limitados a sistemas nos quais a relação entre entrada e saída é conhecida e nos quais não há distúrbios internos ou externos consideráveis. Como pode ser observado na Figura 1, esta configuração do sistema é incapaz de compensar distúrbios de diversas naturezas (CAMPOS e TEIXEIRA, 2006).

Figura 1 - Diagrama de um sistema de controle em malha aberta



Fonte: Autora, 2024.

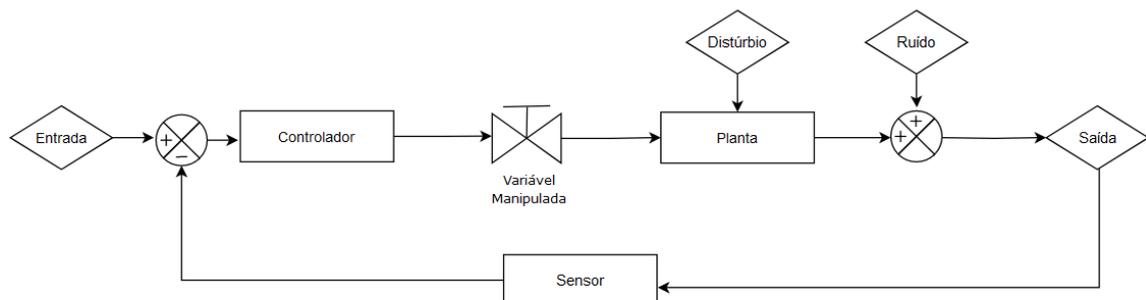
4.3 Sistemas de Controle em Malha Fechada

Em contrapartida, sistemas de controle de malha fechada possuem disposição mais complexa e custosa, entretanto tem como principal vantagem sua considerável eficiência na correção de desvios causados por variações internas e distúrbios externos. Dentre as configurações possíveis para o controle por malha fechada, a mais tradicional é a operação por retroalimentação.

Segundo Luyben (1996), sistemas de controle do tipo retroalimentação (*feedback*) são operados através da seguinte metodologia: há a medição de uma variável a ser controlada, em seguida o valor medido é comparado com um valor desejado (*setpoint*) e esta diferença é alimentada para um controlador que irá manipular uma outra variável de forma a levar a variável controlada a assumir o valor desejado.

Neste processo, é necessário que um sensor detecte a variável a ser controlada e que um transmissor converta este sinal do sensor para um sinal equivalente que possa ser interpretado pelo controlador. Após a leitura do sinal pelo controlador, o mesmo realiza tanto o cálculo do erro e produz um sinal de saída que pode ser, por sua vez, interpretado por um elemento final de controle, por exemplo uma válvula cuja abertura é dependente do valor de sinal que esta recebe. Este processo pode ser observado no diagrama da Figura 2.

Figura 2 -Diagrama de um sistema de controle de malha fechada



Fonte: Autora, 2024.

4.4 Análise de Sistemas de Controle

Devido à diferença de complexidade, de custo e, possivelmente, de instabilidade da planta, a escolha do tipo de malha de controle a ser implementada em um processo deve ser estruturada na análise dos parâmetros padrões de desempenho deste sistema. Segundo Nise (2011), o processo de análise de um sistema é essencial para a determinação quantitativa da performance do mesmo. Dentre os objetivos pertinentes à construção desses sistemas os quais são observados durante o processo de análise, destacam-se: a geração de uma resposta transitória adequada, a redução do erro de regime permanente e a obtenção de estabilidade no sistema. Todos esses objetivos podem ser quantificados com o auxílio de técnicas matemáticas conhecidas.

Luyben (1996) afirma que o estudo da dinâmica de sistemas de controle, sob o domínio do tempo, envolve a solução de equações diferenciais devido a sua natureza dinâmica. As soluções dessas equações diferenciais, portanto, podem ser determinadas através de diversos métodos analíticos e/ou numéricos (por exemplo, a aplicação da transformada de Laplace e o método de Runge-Kutta, respectivamente).

4.4.1 Resposta Transitória do Sistema

Sob essa ótica, a obtenção da resposta transitória de um sistema de controle de retroalimentação, por exemplo, depende inicialmente do conhecimento da função característica do distúrbio sofrido pelo sistema. Esses distúrbios podem ser modelados através das seguintes configurações: degrau (função unitária), pulso (podendo apresentar diversas formas como retangular ou triangular), impulso (função delta de Dirac), rampa (função linear) ou curva sinusoidal.

Ademais, há a importante distinção entre objetivos de controle e, conseqüentemente, comportamento de resposta do sistema, dados pelos problemas do tipo Regulatório e do tipo Servo. Em problemas do tipo regulatório, perturbações de carga que afetam o processo (mudanças na composição de alimentação e na temperatura de resfriamento, por exemplo) resultam em respostas regulatórias, onde o mecanismo de controle é baseado na ação sob a variável manipulada de forma a manter a variável controlada em um valor constante independente do distúrbio. Já em problemas do tipo servo, a mudança do valor desejado para a variável controlada resulta em uma estratégia de controle diferente. Nesse caso, a resposta do sistema é chamada de resposta servo e visa levar a variável controlada ao novo valor desejado de forma que o valor dessa variável mude ao longo do tempo até alcançar, em seu estado estacionário, o novo valor desejado (LUYBEN, 1996).

4.4.2 Estabilidade do Sistema

Como já dito, um outro parâmetro essencial para a análise de eficiência de uma malha de controle é a estabilidade do sistema. Nise (2011), afirma que a definição de estabilidade de um sistema é diversa e depende do tipo de sistema observado.

No entanto, considerando sistemas lineares e sabendo que a resposta de um sistema é dada pela soma das respostas naturais e forçadas pelo controlador, tem-se que, neste contexto, uma definição de estabilidade viável é a dada pela categoria de sistemas de “entrada-limitada, saída-limitada” ou “*bounded-input, bounded-output*” (BIBO, na sigla em inglês). Podendo estes, serem delimitados pelas categorias a seguir:

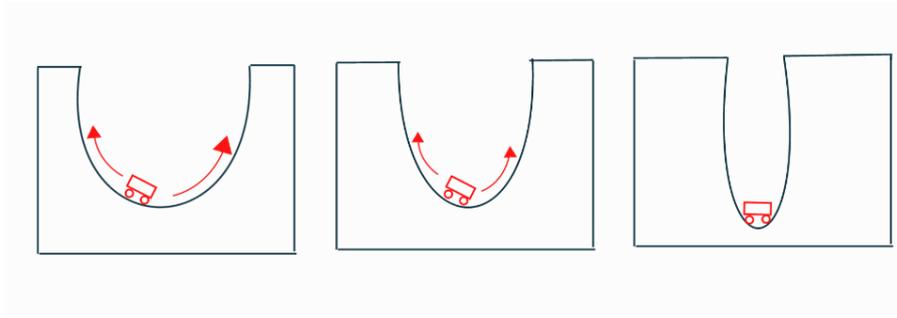
- Estáveis: Quando a resposta natural do sistema tende a zero quando o tempo tende ao infinito.
- Instáveis: Quando a resposta natural do sistema cresce de forma não limitada quando o tempo tende ao infinito;
- Marginalmente estáveis: Quando a resposta natural do sistema se mantém constante ou oscila quando o tempo tende ao infinito.

Um exemplo interessante desse tipo de sistema é a dinâmica entre um skate e duas rampas dispostas de forma convexa. Ao aplicar força considerável (ou seja, uma perturbação impulso) sob um skate parado no piso que conecta as duas rampas, o mesmo tenderá a oscilar entre a rampa da direita e da esquerda até se estabilizar novamente no piso.

Nesse exemplo, há duas condições possíveis do sistema que são relevantes para a discussão de estabilidade: a primeira é quando o sistema atinge um nível de inclinação teórico (sem resistência) no qual o sistema deixa de estabilizar o skate no fundo e o comportamento do mesmo continua oscilatório infinitamente e o segundo é quando a inclinação supera o nível anterior, na marginal de estabilidade, e se torna completamente instável.

Nessa analogia, quanto mais íngreme forem as rampas, menor será o deslocamento e a oscilação do skate. Entretanto, quanto mais estável for o sistema, mais lenta será sua resposta e menor sua tendência a sair de seu equilíbrio. Na analogia já descrita, caso a rampa seja extremamente íngreme, o skate não irá mais se deslocar independente da força aplicada, seja ela uma perturbação desejada ou indesejada. Uma representação visual desse sistema pode ser observada na Figura 3.

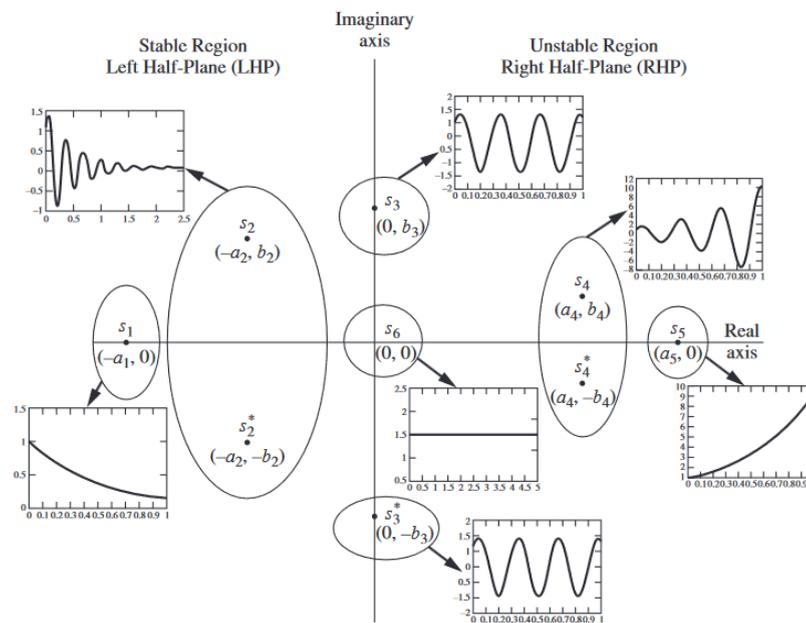
Figura 3 - Exemplo de estabilidade na analogia da rampa de skate



Fonte: Autora (2024).

Segundo Coughanowr e LeBlanc (2009), de forma geral, a estabilidade de um sistema de controle é diretamente relacionado à ação do controlador sob o sistema e, portanto, pode ser matematicamente caracterizado, para sistemas lineares, com base no seguinte critério: Um sistema de controle linear só é instável quando qualquer uma das raízes de sua equação característica (função de transferência geral) estiver sob ou à direita do eixo imaginário. Uma representação gráfica deste conceito pode ser vista na Figura 4:

Figura 4 - Diagrama representativo da estabilidade das raízes típicas de uma equação característica



Fonte: Coughanowr, Donald R. (2009)

4.5 Modelagem Matemática de um Sistema

Com base nos conceitos já descritos, é fácil ver a importância da definição adequada da estratégia de controle proposta, de forma que esta esteja de acordo com a dinâmica específica do processo a ser controlado. Para que esta definição ocorra da melhor forma, é necessário inicialmente compreender, através da modelagem matemática, o sistema de interesse.

No ponto de vista da modelagem matemática, um sistema pode ser definido como um conjunto confinado de entidades mutuamente impactadas. Neste contexto, essas entidades são processos, sendo o processo uma atividade que converte e/ou transporta matéria, energia e/ou informação. Estas entidades podem ser ainda subdivididas nos subprocessos que as compõem. Estes subprocessos são processos individuais nas quais há uma única responsabilidade principal (ISERMANN; MÜNCHHOF, 2011).

Relembrando o sistema eletrolítico descrito na introdução, tem-se que um exemplo interessante da relação entre sistemas, processos e subprocessos é o de uma planta industrial onde ocorre um processo de geração de desinfetantes. A planta como um todo é equivalente ao sistema, já os processos são dados, de forma geral, pelos comportamentos da célula eletrolítica, do processo de bombeamento e das operações unitárias relacionadas ao tratamento inicial da solução e ao sistema de ventilação utilizado para a exaustão do hidrogênio gasoso. Por fim, cada um desses processos é composto de subprocessos como a reação de eletrólise e a difusão que ocorrem na célula e a conversão de energia que ocorre entre o motor e a bomba.

Dessa forma, fica evidente que o comportamento de um sistema é definido pelo conjunto de comportamentos de todos os seus processos e subprocessos. Com base nisso, tem-se que a modelagem matemática de um sistema pode ser realizada através de duas possíveis metodologias: a modelagem fenomenológica ou teórica, na qual é possível identificar a relação entre a entrada e a saída de um sistema, ou a modelagem empírica, na qual apenas a relação entre entrada e saída do sistema pode ser determinada.

4.5.1 Modelagem Fenomenológica e Experimental do Sistema

A determinação de modelos matemáticos para sistemas dinâmicos depende inicialmente da escolha diligente entre a realização da modelagem fenomenológica ou experimental (ISERMANN; MÜNCHHOF, 2011).

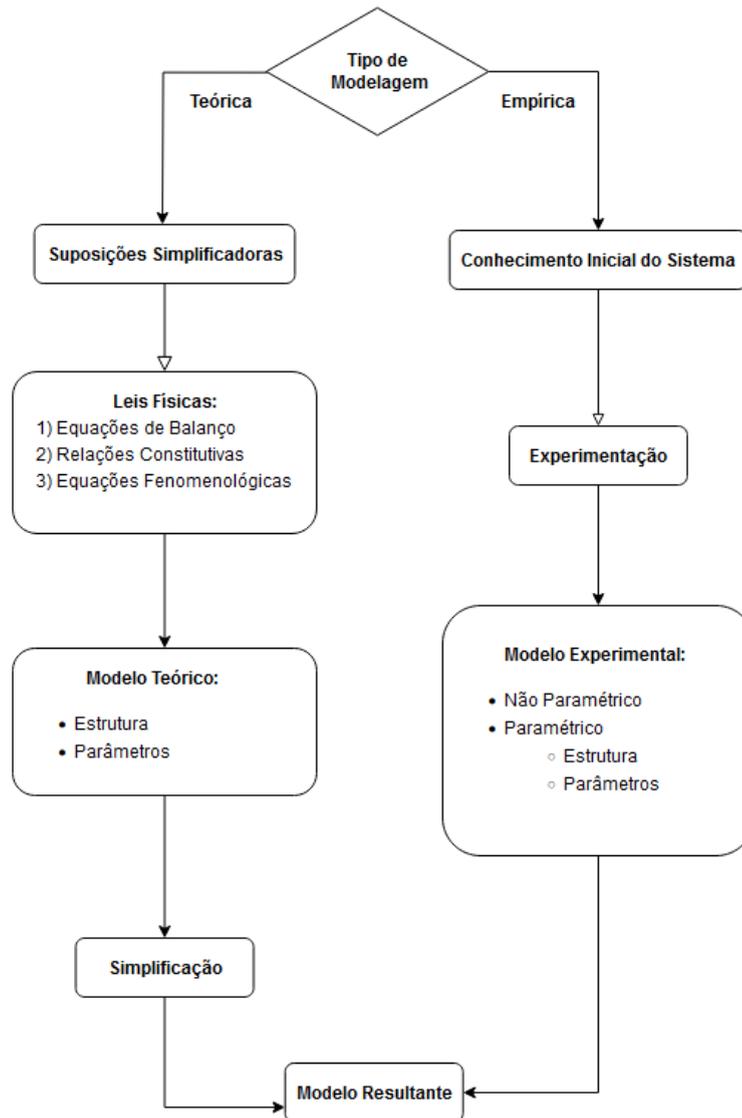
Seborg e Mellichamp (1989), afirmam que, de forma a modelar a dinâmica de um processo, é necessário o conhecimento dos fenômenos físicos e químicos que descrevem o estado daquele sistema. Esses fenômenos são diversos, mas podem ser classificados de acordo com leis fundamentais como as equações de continuidade, a conservação de energia, os fenômenos de transporte, equações de equilíbrios, entre outros.

Já para Isermann e Münchhof (2011), a modelagem por vias experimentais, também conhecida como identificação de sistema, é um processo no qual o modelo é resultante de medições feitas no próprio sistema e da utilização destas em métodos de identificação que visam obter uma relação entre a entrada e saída do sistema. Estes modelos experimentais são

primordialmente classificados como modelos paramétricos ou não-paramétricos dependendo da existência de estrutura e da quantidade de parâmetros (finita ou infinita) do modelo.

Independentemente do tipo de modelagem escolhida, há procedimentos conhecidos na literatura para a geração e validação dos mesmos. A sequência desses procedimentos pode ser observada na Figura 5:

Figura 5 - Procedimento Básico para Modelagem Matemática



Fonte: Adaptada de Isermann e Münchhof (2011).

No caso da escolha pela modelagem teórica, é essencial que a investigação das leis físicas que descrevem o sistema tenha origem em suposições simplificadoras de forma que sejam selecionados os efeitos físicos e/ou químicos mais relevantes para o sistema. Assim que as suposições forem tomadas e as leis relevantes forem selecionadas, é possível obter um modelo teórico preliminar.

Entretanto, na maioria dos casos, este modelo inicial será bastante complexo, logo é indicado que o mesmo seja submetido à simplificação de forma que as equações obtidas possam ser analiticamente ou numericamente resolvidas no menor tempo possível.

Em contrapartida, caso seja escolhida a modelagem empírica, o procedimento é dado pela coleta de dados experimentais de sinais de entrada e de saída do sistema e pelo uso dos mesmos - com base em conhecimentos fundamentais sobre o funcionamento do sistema e seu comportamento ao longo do tempo - em métodos de identificação relevantes (ex: análise de Fourier, redes neurais, medição de resposta de frequência, entre outros).

Segundo Isermann e Münchhof (2011), apesar da análise teórica ser capaz de oferecer mais informações sobre o sistema, o estudo e aplicação de técnicas de identificação de sistema por modelagem empírica têm se popularizado devido aos seguintes fatos:

- Modelagem teórica tende a ser bastante complexa até mesmo para sistemas simples;
- De forma geral, os coeficientes resultantes de considerações teóricas não são muito precisos;
- Nem todas os subprocessos que ocorrem em um sistema são conhecidas;
- Uma parcela dos subprocessos conhecidos não podem descritos matematicamente mantendo acurácia suficiente e possibilidade de resolução das equações geradas;
- Modelagem experimental pode ser obtida em um tempo mais curto e com menos esforço se comparada com a modelagem fenomenológica.

4.5.2 Modelagem Empírica para Sistemas Dinâmicos

Com base no exposto acima, é evidente que há consideráveis vantagens na aplicação de modelagem empírica, especialmente para aplicações simples como o desenvolvimento e sintonia de controladores clássicos. Assim, sob a ótica dessa metodologia de modelagem, é necessário distinguir os modelos empíricos paramétricos e não paramétricos de forma a optar pelo mais adequado ao sistema em questão.

Para Kashiwagi (2009), os métodos paramétricos geram modelos com uma estrutura matemática bem definida a qual está associada com um conjunto de coeficientes ou parâmetros. O método mais simples para modelagem paramétrica é a determinação de valores característicos, a qual se baseia na utilização de medições de resposta a uma perturbação do tipo degrau ou impulso para determinação de valores característicos como tempo morto ou ordem da dinâmica do processo.

Já os métodos não paramétricos modelam o sistema diretamente a partir de suas respostas. Para sistemas lineares, exemplos comuns de modelos não paramétricos são: as respostas a um impulso, a um degrau ou em frequência. Nestes métodos, as medições de resposta permitem a determinação direta de pontos discretos da resposta em frequência para diversos processos lineares.

4.5.3 Linearidade dos Sistemas

Como já foi mencionado, de forma a optar pelo método de modelagem mais apropriado para um sistema, é necessário ter alguns conhecimentos iniciais sobre o mesmo. Uma das informações mais relevantes relacionadas a essa escolha, é a da linearidade do sistema.

Segundo Lathi e Green (2018), a linearidade de um sistema implica no fato de que a sua representação matemática satisfaz simultaneamente as propriedades de aditividade e homogeneidade.

A propriedade de aditividade pode ser definida através da seguinte expressão: Para um sistema linear, se uma entrada x_1 de ação individual tem um efeito y_1 e, se uma outra entrada x_2 , também de ação individual, tem um efeito y_2 , tem-se que, se ambas as entradas agirem em conjunto no sistema, o efeito resultante será $y_1 + y_2$. Dessa forma, em um sistema aditivo, a seguinte expressão (Equação 2) é válida:

$$x_1 \rightarrow y_1 \quad e \quad x_2 \rightarrow y_2 \quad \Rightarrow \quad x_1 + x_2 \rightarrow y_1 + y_2 \quad (2)$$

Já a propriedade da homogeneidade, é dada de forma que, se uma entrada x do sistema é multiplicada por um valor k real ou imaginário, seu efeito y deve também se tornar um múltiplo de k . Assim, a expressão (3) é válida:

$$x \rightarrow y \quad \Rightarrow \quad kx \rightarrow ky \quad (3)$$

De forma geral, a superposição de ambas as propriedades descritas acima em um mesmo sistema o torna linear, logo é possível dizer que a resposta desse sistema pode ser expressa como a soma das condições iniciais do sistema (tempo = 0) junto com a entrada $x(t)$ para qualquer valor de tempo, sendo tempo maior ou igual a zero.

A propriedade de sistemas lineares descrita acima é chamada de propriedade de decomposição e sua definição é válida para sistemas de única entrada e única saída (SISO, na

sigla em inglês), mas também pode ser facilmente estendida para sistemas lineares de múltiplas entradas e múltiplas saídas (MIMO, na sigla em inglês).

4.5.4 Sinais de Teste

Há uma grande diversidade de métodos capazes de produzir modelos experimentais válidos para sistemas com características diferentes. Uma parcela considerável desses métodos se dá pela geração inicial de sinais de teste.

Segundo Isermann e Münchhof (2011), para que a identificação de um sistema seja favorável ao uso de sinais de testes, os sinais relevantes devem satisfazer os seguintes critérios:

- A geração e reprodução do sinal de teste deve ser simples, com ou sem um gerador de sinais;
- A descrição matemática do sinal e de suas propriedades deve ser simples;
- A geração e reprodução deve ser possível com os atuadores disponíveis no sistema;
- Os sinais devem ser aplicáveis ao processo;
- Os sinais de teste devem gerar distúrbios consideráveis na dinâmica do sistema.

Há diversas possibilidades de sinais de teste que podem satisfazer esses critérios na maioria dos sistemas. Alguns exemplos dessas possibilidades são: sinais não periódicos como pulsos degrau ou pulso quadrado, sinais periódicos como curvas seno ou ondas quadradas e sinais estocásticos como sinais binários de ruído.

Também segundo Isermann e Münchhof (2011), um exemplo comum do uso de sinais de teste para modelagem matemática pode ser dado pelo distúrbio de um sistema através da execução de um impulso em sua entrada e do subsequente estudo da resposta obtida. A função de impulso, também conhecida como função delta de Dirac, é definida pelas seguintes relações (Equações 4 e 5):

$$\delta(t) = \{\infty \text{ para } t = 0\} \text{ e } \delta(t) = \{0 \text{ para } t \neq 0\} \quad (4)$$

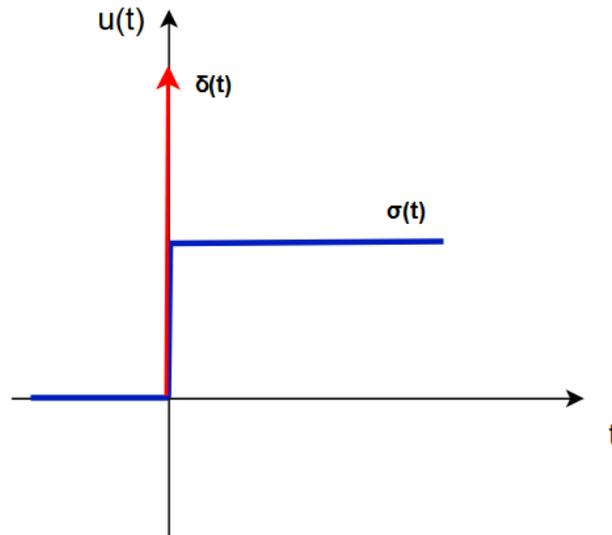
$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (5)$$

Uma outra função também interessante para a identificação de sistemas é a função degrau, também chamada de função de Heaviside. A função degrau pode ser obtida através da integração da função de impulso com respeito ao tempo. Ela é definida matematicamente pela expressão abaixo (Equação 6):

$$\sigma(t) = \{1 \text{ para } t \geq 0\} \text{ e } \delta(t) = \{0 \text{ para } t < 0\} \quad (6)$$

Visualmente, a relação entre a função de Dirac (δ) e a função de Heaviside (σ) em um mesmo sistema pode ser observada na Figura 6:

Figura 6 - Representação gráfica da relação entre as funções degrau e impulso



Fonte: Autora (2024).

4.6 Determinação de Parâmetros Característicos

Apesar da simplicidade, a utilização de sinais de teste, especialmente para a observação da resposta de perturbações degrau ou de impulso, pode ser bastante útil para a identificação de sistemas. Observando o comportamento do sistema antes e depois da perturbação, é possível determinar valores aproximados dos parâmetros característicos da função de transferência do sistema como, por exemplo, os valores de tempo morto, coeficiente de amortecimento, etc.

Entretanto, para se ter clareza sobre o processo de determinação e, assim, conhecer mais sobre a identificação de sistemas, é necessário antes explorar a transformada de Laplace e sua inversa, como também entender a relação de ambas com o conceito de funções de transferência.

Segundo Nise (2011), a transformada de Laplace é uma função bastante usada para simplificar a modelagem de um sistema, pois possibilita que a entrada, a saída e o processo sejam representados como entidades separadas cujas relações são puramente algébricas.

A definição desta função pode ser observada na Equação (7), onde é possível ver que a notação para o limite inferior da sua integral característica é dada de forma que, mesmo um sistema apresentando descontinuidade em $t = 0$, a transformada ainda pode ser encontrada,

desde que a integral seja convergente e as condições iniciais do sistema sejam conhecidas antes da descontinuidade.

$$\mathcal{L}[f(t)] = F(s) = \int_{-0}^{\infty} f(t) * e^{-st} dt \quad (7)$$

Uma relação também bastante importante para o estudo da resposta de sistemas é o da função inversa da transformada de Laplace, a qual relaciona diretamente a função do sistema no domínio do tempo com a função de Heaviside. Esta relação pode ser verificada na definição da função inversa descrita na Equação (8) abaixo:

$$\mathcal{L}^{-1}[F(s)] = f(t) * \sigma(t) = \frac{1}{2\pi j} \int_{\theta-j\infty}^{\infty+j\infty} F(s) * e^{st} ds \quad (8)$$

Tendo definido a transformada de Laplace e a sua inversa, é possível então formular a representação de um sistema através de uma ferramenta crucial para a modelagem matemática: a função de transferência.

Para Nise (2011), ao considerar um sistema linear cuja resposta não varia com o tempo, é possível o descrever através da seguinte equação diferencial genérica de ordem n (Equação 9):

$$a_n \frac{d^n u(t)}{dt^n} + a_{n-1} \frac{d^{n-1} u(t)}{dt^{n-1}} + \dots + a_0 u(t) = b_m \frac{d^m r(t)}{dt^m} + \dots + b_0 r(t) \quad (9)$$

Aplicando a transformada de Laplace em ambos os lados, tem-se a seguinte equação (Equação 10):

$$a_n s^n U(s) + a_{n-1} s^{n-1} U(s) + \dots + a_0 U(s) + u_0 = b_m s^m R(s) + \dots + b_0 R(s) + r_0 \quad (10)$$

Sendo u(t) equivalente à entrada e r(t) à saída do sistema e assumindo que todas as condições iniciais são iguais a zero, a equação acima se torna igual a equação a seguir (Equação 11):

$$(a_n s^n + a_{n-1} s^{n-1} + \dots + a_0) * U(s) = (b_m s^m + \dots + b_0) * R(s) \quad (11)$$

Assim é obtida a função de transferência desse sistema (descrita como $G(s)$), a qual é capaz de descrever o mesmo através da razão entre as suas transformadas de saída e de entrada, como pode ser observado na equação abaixo (Equação 12):

$$\frac{R(s)}{U(s)} = G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_0} \quad (12)$$

Agora que a definição de uma função de transferência foi estabelecida, é possível seguir com o método de determinação dos parâmetros característicos.

Segundo Isermann e Münchhof (2011), esse método é extremamente simples, mas pode apenas ser utilizado para sistemas nos quais a resposta a uma perturbação degrau apresenta apenas perturbações externas insignificantes e nos quais o processo pode ser linearizado e aproximado por um modelo simples.

Este método é capaz de aproximar a descrição da dinâmica do processo por um sistema de genérico de ordem n , mas, para não adicionar complexidade superior à de interesse deste trabalho, serão descritos apenas três casos de aproximação dados pelos modelos de sistemas: de primeira ordem, de segunda ordem e de primeira ordem com tempo morto considerável.

Um sistema de primeira ordem pode ser representado pela seguinte função de transferência (Equação 13) e por sua resposta à perturbação degrau (Equação 14):

$$G(s) = \frac{b_0}{1 + a_1 s} = \frac{K}{1 + sT} \quad (13)$$

$$r(t) = K * u_0 * (1 - e^{-\frac{t}{T}}) \quad (14)$$

Dessa forma, é fácil ver que a resposta ao degrau de um sistema de primeira ordem pode ser descrita através de um valor de ganho K e de uma constante de tempo T . Assim, devido a seu comportamento exponencial conhecido, é possível determinar os valores característicos da função de transferência desse sistema através do estudo da resposta do mesmo a essa perturbação simples.

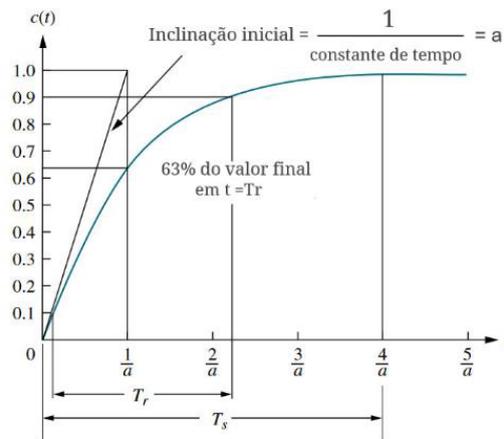
Como base dessa análise, há relações gráficas que facilitam a obtenção desses valores. Essas relações são: a equação (Equação 15) para o ganho dada pela relação entre o valor da resposta final - $r(\infty)$ - e a amplitude do distúrbio (u_0) e a equação diferencial para obtenção da constante de tempo (Equação 16).

$$K = \frac{r(\infty)}{u_0} \quad (15)$$

$$\frac{dr(t)}{dt} = \frac{r(\infty)}{T} * e^{-\frac{t}{T}} \quad (16)$$

Graficamente, estas relações podem ser observadas no gráfico abaixo (Figura 7), onde é possível observar a constante de tempo de ascensão (representada como $t=Tr$) sendo descrita como o tempo até a resposta ao degrau alcançar 63% do seu valor final.

Figura 7 - Resposta de um sistema de 1º ordem a uma perturbação degrau



Fonte: Adaptado de Nise (2011).

Ainda considerando um sistema de primeira ordem, é possível obter os parâmetros de tempo de ascensão (Tr) e tempo de acomodação (Ts) por relações conhecidas, sendo elas, respectivamente, a equação que representa o tempo para a curva ir de 10% a 90% do valor final (Equação 17) e a equação que descreve o tempo para a resposta alcançar e permanecer em 2% do valor final (Equação 18), descritas abaixo:

$$T_r = \frac{2,31}{a} - \frac{0,11}{a} = \frac{2,2}{a} \quad (17)$$

$$T_s = \frac{4}{a} \quad (18)$$

Para sistemas de segunda ordem, entretanto, a equação genérica que os define (Equação 19) apresenta três parâmetros característicos: o ganho K , a razão de amortecimento ζ e a frequência natural ω_n :

$$G(s) = \frac{K}{1 + T_1 s + T_2^2 s^2} = \frac{K}{1 + \frac{2\zeta}{\omega_n} s + \frac{1}{\omega_n^2} s^2} \quad (19)$$

Conforme sua definição, sistemas de segunda ordem são também definidos pela presença de um par de polos dominantes cuja natureza matemática está diretamente relacionada ao valor da sua razão de amortecimento. Essa relação para qualquer um dos polos é dada, de forma genérica, pela equação a seguir (Equação 20):

$$s_{1,2} = \omega_n (-\zeta \pm \sqrt{\zeta^2 - 1}) \quad (20)$$

Com base no valor da razão de amortecimento, há três cenários possíveis para os polos do sistema: o radicando pode ser positivo, negativo ou zero. Há três possíveis estados do sistema que podem alterar a razão de amortecimento do mesmo: sistema sobreamortecido, sistema criticamente amortecido ou sistema subamortecido.

Caso o sistema esteja sobreamortecido, a razão de amortecimento será superior a 1 e haverá dois polos reais, negativos e não iguais no sistema. Dessa forma, a resposta ao degrau deste sistema será igual à Equação (21):

$$\sigma(t) = K * \left\{ 1 + \frac{1}{s_1 - s_2} (s_2 e^{s_1 t} - s_1 e^{s_2 t}) \right\} \quad (21)$$

Entretanto, caso o sistema esteja criticamente amortecido, a razão de amortecimento será igual a 1 e haverá dois polos reais, negativos e iguais no sistema. Para este tipo de sistema, a resposta ao degrau será igual à Equação (22):

$$\sigma(t) = K * \{ 1 - e^{-\omega_n t} [1 + \omega_n(t)] \} \quad (22)$$

Por fim, caso o sistema esteja subamortecido, a razão de amortecimento será menor que 1 e maior que 0 de forma que os polos desse sistema se encontrarão no eixo imaginário. Ao contrário dos casos anteriores, a resposta desse sistema apresentará oscilações amortecidas.

Com base nisso, para que seja possível modelar a resposta ao degrau nesse caso, é necessário antes introduzir outros dois valores característicos adicionais, frequência natural amortecida e coeficiente de amortecimento, dados, respectivamente, pela Equação 23 e Equação 24:

$$\omega_d = \omega_n * \sqrt{1 - \zeta^2} \quad (23)$$

$$\gamma = \omega_n * \zeta \quad (24)$$

Assim, é possível determinar a resposta à perturbação desse sistema (Equação 25):

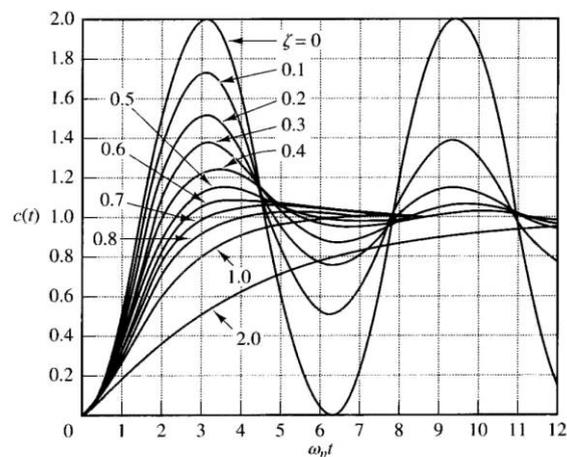
$$\sigma(t) = K * \left(1 - \frac{1}{\sqrt{1 - \zeta^2}} * e^{-\gamma t} * \text{seno}(\omega_d t + \varphi) \right) \quad (25)$$

A resposta descrita acima é uma função senoidal amortecida com mudança de fase, sendo esta fase definida pela equação abaixo (Equação 26):

$$\varphi = \text{arctg}\left(\frac{\omega_d}{\gamma}\right) = \text{arctg}(\zeta\sqrt{1 - \zeta^2}) \quad (26)$$

De forma geral, dada a resposta ao degrau, é possível identificar os parâmetros característicos do sistema observando esta resposta ao longo do tempo, como pode ser visto na Figura 8.

Figura 8 - Resposta de um sistema de 2º ordem a uma perturbação degrau



Fonte: Ogata (1997).

4.7 Algoritmo de Controle

Após a modelagem do sistema a ser controlado, é necessário determinar um algoritmo de controle a ser implementado. Há diversas considerações que podem ser feitas sobre essa escolha, mas em contextos em que não há um alto nível de ruído nos sinais captados e a precisão extrema não é um requisito, controladores do tipo PID são a escolha majoritária da indústria. Esta afirmação está de acordo com o encontrado por Miller e Nordh (2000) em um questionário

realizado em 18 plantas industriais nos Estados Unidos, onde foi observado uma prevalência da utilização de algoritmos do tipo PID em 97% das aplicações observadas.

4.7.1 Controladores do tipo PID

Segundo Campos e Teixeira (2006), a controladores do tipo PID atuam no processo de forma a gerar uma saída proporcional ao erro: instantâneo, à sua integral e à sua derivada. A equação de um algoritmo de um controlador PID clássico em paralelo e sua função de transferência são dados, respectivamente pelas Equações (27) e (28):

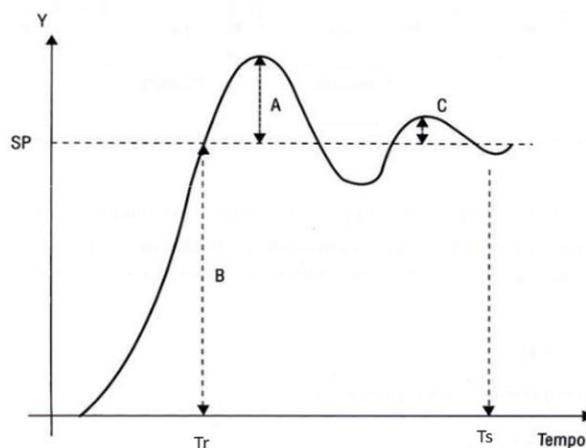
$$u(t) = K_p * e(t) + K_p \left(\frac{1}{T_i} \right) * \int e(t) dt + K_p T_D * \left(\frac{de(t)}{dt} \right) + u_0 \quad (27)$$

$$G(s) = K_p * \left[1 + \frac{1}{s * T_i} + s * T_d \right] * E(s) \quad (28)$$

Além da implementação tradicional descrita acima, é possível também implementar controladores PID do tipo paralelo alternativo (onde o sistema ainda é paralelo, vide a soma em paralelo das ações proporcional, integral e derivativa; mas nessa configuração o ganho proporcional não impacta diretamente a ação integral) e os controladores PID em série (no qual o cálculo de dois ou mais termos do sistema são realizados em série).

Segundo Corripio (1990), essas considerações de implementação do algoritmo influenciam diretamente no processo de sintonia do controlador. Métodos de sintonia são técnicas usadas para ajustar os ganhos do controlador PID para alcançar o desempenho de controle desejado. Considerando o controlador PID tradicional já descrito matematicamente nas equações acima, de forma geral uma resposta dinâmica para uma variável controlada de caráter genérico Y, pode ser expressa segundo a Figura 9:

Figura 9 - Resposta dinâmica de uma malha de controle PID clássica

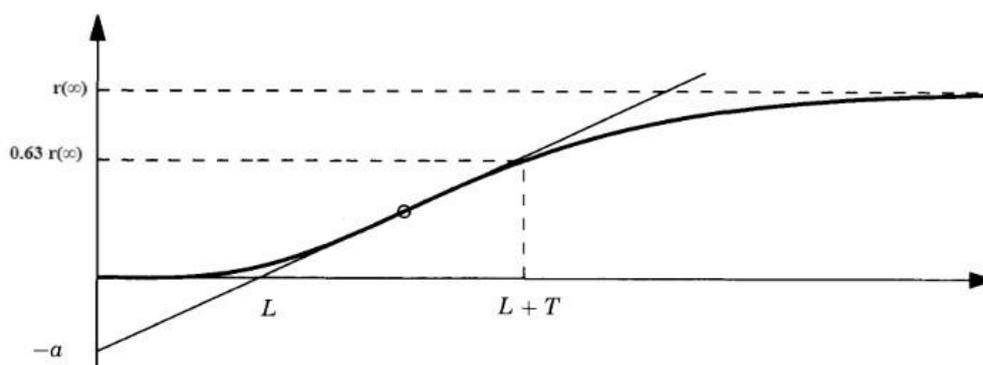


Fonte: Adaptada de Campos e Teixeira, 2006.

Com base na Figura 09 é fácil ver os critérios que costumam ser utilizados para a execução da sintonia do controlador. A sintonia busca, então: controlar a razão de declínio (equivalente a razão C/A), minimizar o “overshoot” (equivalente a razão A/B na figura) e a atuação na variável manipulada, além de reduzir o tempo de subida e de assentamento o tempo de subida. (CAMPOS e TEIXEIRA, 2006).

Segundo Aström e Hägglund (2005), a base de grande parte das metodologias clássicas de procedimento para a sintonia de um controlador PID é dada pelos dois métodos de sintonia descritos por Ziegler e Nichols em 1942. O primeiro método é baseado na resposta de um sistema operando em malha aberta a uma perturbação degrau. A dinâmica e os parâmetros de processo observados nessa resposta são dados pelos observados na Figura 10.

Figura 10 - Resposta ao degrau unitário de um processo usada para determinar os parâmetros do processo



Fonte: Adaptada de Aström e Hägglund, 2005.

O segundo método, por sua vez, consiste na determinação do valor do ganho e do período de oscilação no qual a malha está no limite da estabilidade sob o contexto da presença de um controlador apenas proporcional em configuração de retroalimentação. De forma geral, com base nos parâmetros obtidos no primeiro ou segundo método, os coeficientes característicos dos controladores P, PI e PID podem ser calculados através de relações tabeladas.

Entretanto, nem sempre os métodos de Ziegler-Nichols são eficientes, com base nisso diversos conjuntos de relações foram desenvolvidos durante os anos para compensar essas dificuldades. Entre eles, há o método de sintonia de Cohen-Coon dado para sistemas de primeira ordem com tempo morto (FOTD, na sigla em inglês).

No método de Cohen-Coon, o procedimento de obtenção dos parâmetros do processo é igual ao primeiro método de Ziegler-Nichols, entretanto um novo parâmetro é utilizado nas relações tabeladas, este é τ o qual pode ser obtido pela Equação 29.

$$\tau = \frac{L}{T} \quad (29)$$

Por fim, há o método de Tentativa e Erro, a qual não é baseada nos métodos de Ziegler-Nichols, mas sim no processo, por tentativa e erro, em que os ganhos PID iniciais são ajustados com base nas características de resposta do sistema.

De forma geral, cada método de ajuste tem suas vantagens e desvantagens e é adequado para diferentes características de sistemas e estratégias de controle.

4.8 Hardware para construção de malhas de controle

Por fim, para a implementação de um sistema de controle é necessário entender os elementos básicos de sua instrumentação. Os componentes básicos de uma malha fechada incluem:

- **Sensor:** Equipamento responsável por medir a variável controlada e convertê-la em um sinal elétrico ou digital. Por exemplo, em um sistema de temperatura, um transmissor pode ser usado para medir a temperatura e fornecer um sinal proporcional.
- **Controlador:** O “cérebro” da malha fechada, o qual recebe o sinal do sensor e o compara com o valor de referência.
- **Atuador:** O componente que recebe o sinal de controle do controlador e atua no sistema físico para ajustar a variável controlada. Por exemplo, em um sistema de aquecimento, um relé ou um módulo de controle de potência pode ser usado para ajustar a energia fornecida à resistência.

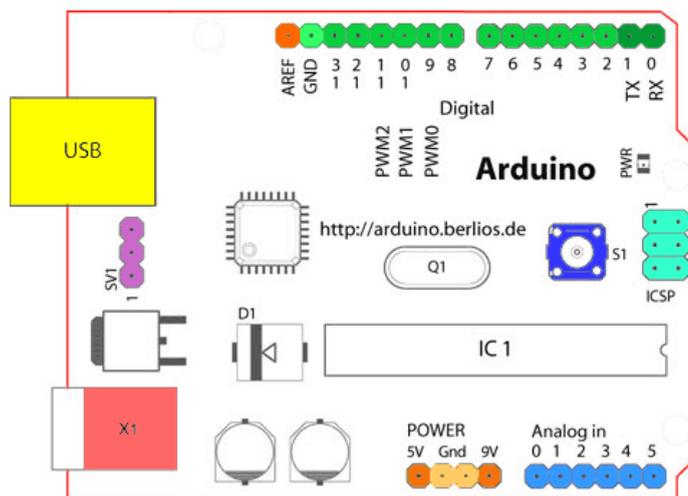
Como já citado, essa instrumentação costuma ser bastante custosa e de implementação complexa. Com base nisso, a proposta deste trabalho está relacionada a implementação da malha fechada de controle com o auxílio da plataforma de prototipagem Arduino, devido a sua acessibilidade, facilidade de programação e a presença de várias entradas analógicas e digitais que permitem a conexão de diversos sensores e atuadores para monitorar e controlar o sistema em tempo real.

O Arduino é uma plataforma de desenvolvimento de protótipos eletrônicos de código aberto, ou seja, seu código fonte pode ser distribuído e utilizado de forma descentralizada e sem custos. Esta plataforma disponibiliza gratuitamente um ambiente de desenvolvimento integrado (IDE, na sigla em inglês) no qual o código nativo da plataforma, uma variante da linguagem C++, é compilado pelo compilador AVR-GCC que é responsável pela conversão do código em linguagem de alto nível e um arquivo binário de forma que o microcontrolador,

tradicionalmente um chip da família megaAVR disposto na placa do Arduino, seja capaz de ler os algoritmos definidos pelo usuário.

Há diversas placas de Arduino disponíveis no mercado, mas a de maior interesse para este trabalho é a placa do tipo Arduino UNO cuja disposição pode ser observada na Figura 11:

Figura 11 - Componentes de um Arduino UNO clássico



Fonte: ARDUINO©.

De forma resumida, tem-se pinos digitais (em verde) e analógicos (em azul) nos quais sinais externos, como por exemplo as medições de um sensor, são alimentados para a placa integrada e, assim, podem ser manipulados pelos algoritmos construídos no ambiente de desenvolvimento.

Ademais, de maior relevância, tem-se o microcontrolador principal descrito como IC1 e o microchip de leitura e escrita do tipo USB-Serial ao lado esquerdo do cristal de oscilação Q1 que, por sua vez, é utilizado como referência de tempo da placa. Por fim, há a entrada USB da placa que é primordialmente utilizada para comunicação entre portas seriais (saídas de comunicação em formato binário em fluxo sequencial as quais estão disponíveis na maioria dos computadores) e o Arduino.

Todos os componentes citados são utilizados neste trabalho para possibilitar o envio e recebimento de sinais entre um computador e o sistema físico. Essa comunicação se baseia no protocolo de comunicação serial assíncrona chamado de receptor-transmissor assíncrono universal (UART, na sigla em inglês), o qual possibilita a configuração do formato dos dados a serem transmitidos e da velocidade de transmissão dos mesmos.

Segundo Osborne (1980), o protocolo UART recebe bytes de dados e transmite sequencialmente bits individuais. No caso do Arduino UNO, há apenas um periférico USART

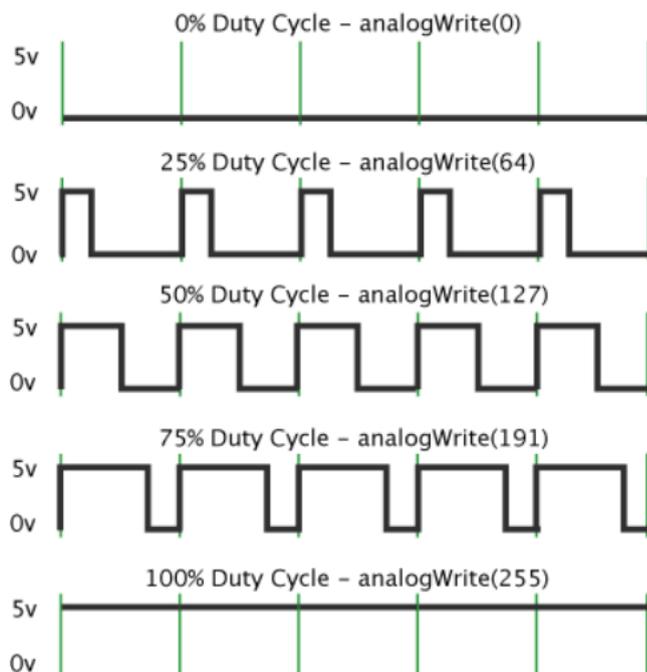
(interface serial que pode ser programada para comunicação síncrona ou assíncrona segundo o protocolo UART) conectado ao PC, logo este é o canal utilizado para a transmissão dos dados que são sequencialmente enviados para comunicação externa.

Além da transmissão de dados para comunicação externa, o Arduino também permite o envio de sinais de saída para os componentes conectados aos seus pinos analógicos e digitais. Esta forma de transmissão ocorre tanto no formato binário - para a escrita em pinos digitais - quanto sob a forma de um valor de 8-bits (na faixa de 0 a 255) - para a escrita em pinos analógicos.

Esta última é de considerável relevância para este trabalho, visto que através da geração de sinais PWM (Modulação de Largura de Pulso, em inglês *Pulse Width Modulation*) é possível controlar a largura do pulso de uma onda quadrada e, assim, proporcionar para o sistema uma potência capaz de assumir valores contínuos.

Por exemplo, através do uso da função `analogWrite` nativa do Arduino é possível enviar um valor entre 0 a 255 para um LED conectado a um pino analógico e, dessa forma, alterar a intensidade do seu brilho de acordo com a voltagem disponível pela alimentação.

Considerando a alimentação sendo equivalente à do conector USB presente na placa (5V), caso fosse inserido um valor de 0 como parâmetro da função `analogWrite`, o LED não emitiria nenhuma luz, pois estaria sendo alimentado por 0V. No caso em que este parâmetro fosse 255, o oposto aconteceria e o LED seria alimentado por 5V (a voltagem máxima de alimentação). Por fim, caso fosse enviado como parâmetro um valor entre 0 e 255, 127 por exemplo, a voltagem a qual o LED seria submetido seria equivalente a um valor intermediário de voltagem. No caso da entrada igual a 127, a voltagem seria igual a 2,5V seguindo uma regra de 3 simples. Visualmente, esta relação pode ser observada pela Figura 12.

Figura 12 - Exemplo de sinais PWM gerados no Arduino

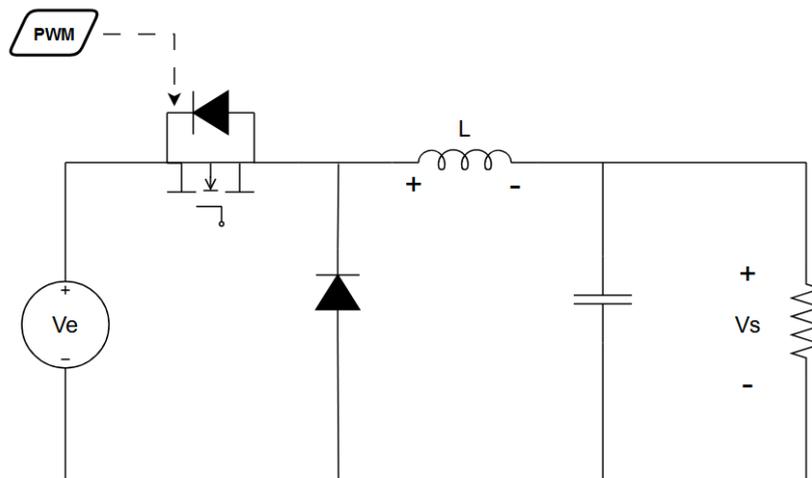
Fonte: HIRZEL, T. (2024)

Considerando o contexto dos componentes básicos da instrumentação de uma malha fechada já citados nesta seção e o conhecimento sobre o método de escrita analógica disponível no Arduino, é fácil ver que há uma vasta gama de possibilidades de arranjos de controle nos quais o componente atuador externo é movido à ação por um sinal PWM enviado pelo Arduino com base em manipulações internas de dados de sensores externos que se comunicam com a placa integrada em questão.

Uma dessas possibilidades, a qual é explorada no presente trabalho, é a do controle digital da tensão de um motor através da implementação de um conversor Buck alimentado por sinais PWM vindos de um Arduino, o controlador neste contexto.

Segundo Viswanatha e Reddy (2017), um conversor Buck é um conversor do tipo CC/CC (circuito eletrônico cuja função é a de converter uma tensão alimentada por uma corrente contínua em outra tensão também contínua) no qual a voltagem de saída é menor em relação à voltagem de entrada. Entretanto, como a potência elétrica de saída é igual à de entrada, tem-se que, neste tipo de circuito, a corrente de saída é maior que a de entrada, de forma a respeitar a lei de conservação de energia. Uma representação deste circuito pode ser vista na Figura 13:

Figura 13 - Representação gráfica de um Conversor Buck



Fonte: Autora (2024).

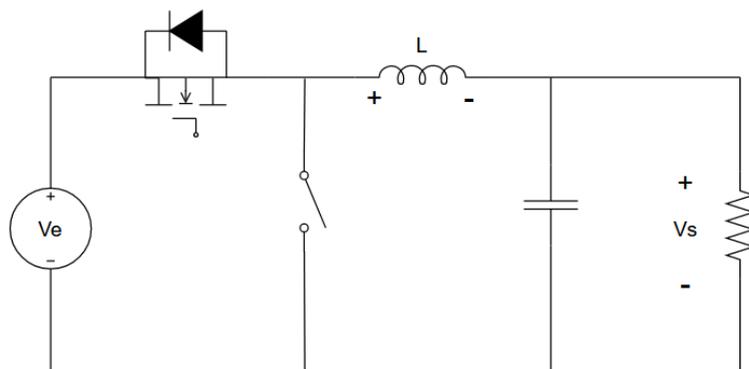
Este tipo de conversor funciona através da ação complementar e oposta de dois componentes eletrônicos: um switch (por exemplo, um transistor do tipo MOSFET) e um diodo. Este mecanismo de ação é dado pela alternância entre dois estados de operação nos quais ocorre a ação oposta dos componentes já citados.

No primeiro estado, o MOSFET está ligado, o diodo está inversamente polarizado, portanto não conduz corrente. Dessa forma, o capacitor é carregado ao mesmo tempo em que ocorre a magnetização do indutor. Já no segundo estado, o MOSFET está desligado e, dessa forma, o diodo consegue conduzir corrente. Neste caso, o indutor é desmagnetizado e, por isso, transfere energia para o condutor (PETRY; BARBI, 2001).

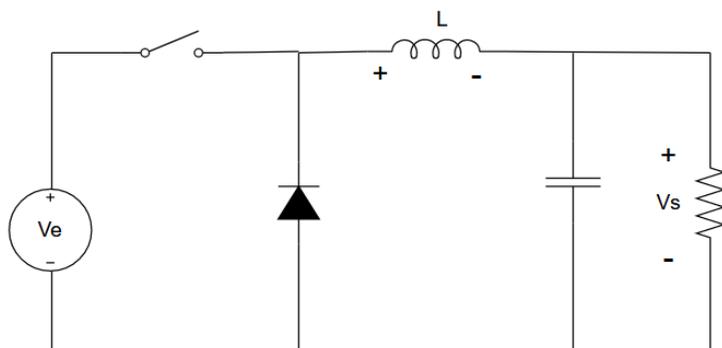
Este circuito possui dois estados distintos de operação: MOSFET ligado e diodo desligado ou diodo ligado e MOSFET desligado. Viswanatha e Reddy (2017) afirmam que as equações de voltagem no indutor para o estado de operação com MOSFET ligado, retratado na Figura 14, e para o estado oposto retratado na Figura 15, podem ser descritas, respectivamente, pelas equações (30) e (31).

$$V_L = (V_e - V_s) * PWM \quad (30)$$

$$V_L = (-V_s) * PWM \quad (31)$$

Figura 14 - Conversor Buck com MOSFET ligado e diodo desligado

Fonte: Autora (2024).

Figura 15 - Conversor Buck com MOSFET desligado e diodo ligado

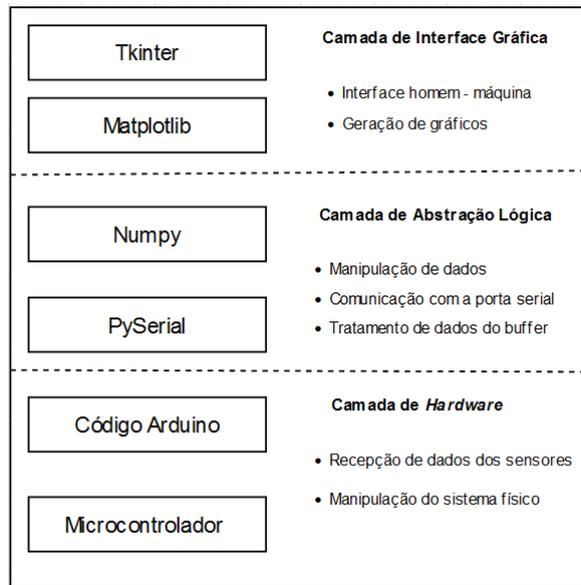
Fonte: Autora (2024)

5. METODOLOGIA

5.1 Sistema Supervisório

Para possibilitar a implementação de um sistema de controle no sistema físico foi necessário a construção de um sistema supervisório robusto composto por três componentes principais: uma interface homem-máquina, um sistema auxiliar de comunicação e o algoritmo de controle definido no Arduino. Esta arquitetura pode ser visualizada na Figura 16 abaixo:

Figura 16 - Visão geral da arquitetura do sistema supervisório



Fonte: Autora, 2024.

5.1.1 Interface Gráfica

A interface homem-máquina foi o componente responsável pelas seguintes tarefas: a apresentação da condição do sistema através da exibição em tempo real das variáveis monitoradas por sensores específicos implementados no sistema em questão (tensão e corrente elétrica, concentração de hidrogênio exaurido, temperatura e vazão da solução circulante) e a leitura de novos valores para a variável controlada (vazão) inseridos pelo operador do sistema.

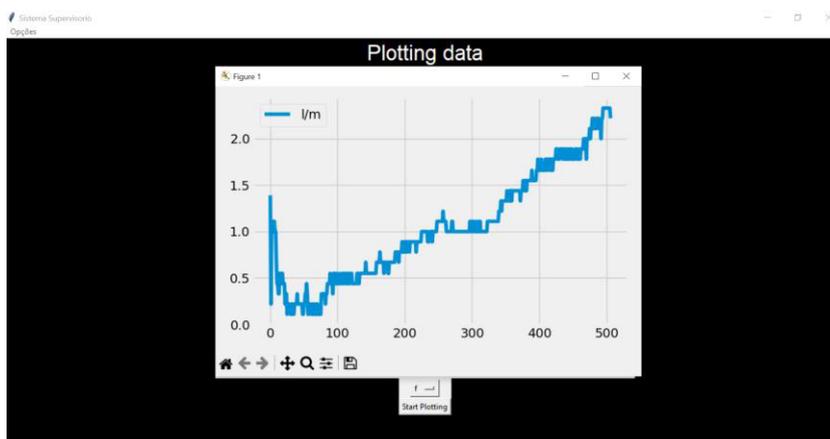
Esta interface gráfica foi construída utilizando Tkinter, um pacote nativo da linguagem Python que possibilita a construção de elementos gráficos denominados *Widgets* que fornecem a base para a construção de interfaces interativas. A interface construída apresenta uma tela inicial, a qual pode ser observada na Figura 17, onde há duas ações possíveis descritas pelos botões de “Monitorar Sistema” e “Alterar Set Point”.

Figura 17 - Tela inicial do sistema supervisorio

Fonte: Autora, 2024.

Com base na ação escolhida pelo operador é aberta uma nova tela com componentes correspondentes à sua escolha.

Para a ação de monitoramento, é aberta uma nova tela (Figura 18) onde é possível escolher a variável a ser monitorada através da interação com um botão customizado. Com base nessa escolha, é realizada a leitura contínua de um arquivo do tipo CSV no qual todas as variáveis são armazenadas. Este processo de leitura ocorre paralelamente ao processo de escrita no arquivo através da utilização de um algoritmo de repetição em laço com o auxílio de dois pacotes disponíveis na linguagem Python: a biblioteca de visualização gráfica Matplotlib e a biblioteca de manipulação e análise de dados chamada Pandas.

Figura 18 - Tela de monitoramento monitorando concentração de Hidrogênio

Fonte: Autora, 2024.

O algoritmo construído para geração de gráficos em tempo real é baseado em três ações principais: Primeiro há a inicialização da subclasse *FuncAnimation* da classe de animação da biblioteca Matplotlib. A inicialização desta subclasse recebe como parâmetros o objeto *fig*

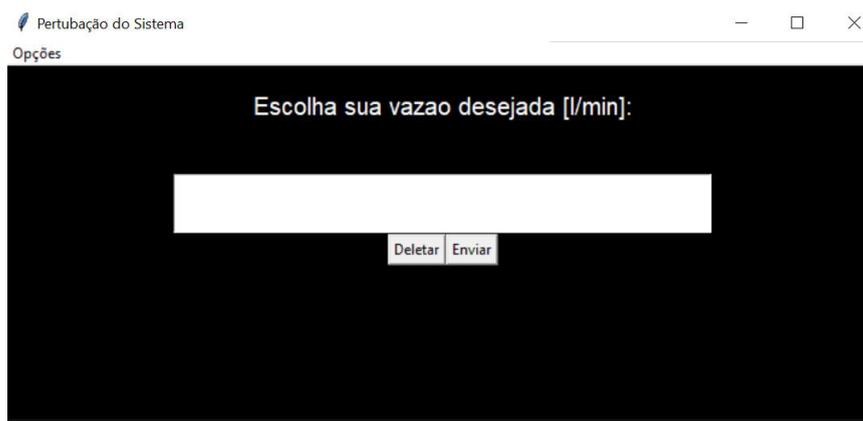
referente à figura ou gráfico a ser animado, a função chamada *func* a qual recebe uma função que será repetidamente animada, uma tupla denominada *fargs* com os argumentos da função a ser animada e dois valores inteiros chamados *interval* e *frames* que, respectivamente, se referem ao intervalo entre quadros e o valor iterativo que alimentará o laço de repetição).

Como argumentos referentes à *figure* e *func*, foram passados, respectivamente, uma figura gerada pelo método *figure* da classe “*pyplot*” da mesma biblioteca e uma função a qual recebe um valor iterativo, lê um arquivo CSV inteiro, faz a sua conversão para uma estrutura de dados bidimensional de tamanho mutável através da instanciação da classe *Dataframe* da biblioteca Pandas e, finalmente, traça um gráfico com os dados presentes na coluna equivalente à variável escolhida pelo operador. Devido a uma limitação do tamanho do texto do eixo, todos os gráficos são exibidos com um eixo horizontal numérico sem significado físico, o qual é gerado automaticamente pela biblioteca.

Por fim, os processos de leitura do arquivo e esboço do gráfico ocorrem de forma iterativa seguindo o número de quadros e o intervalo delimitados nos argumentos da instância. Como um adendo, é importante esclarecer que o processo de escrita das variáveis no arquivo é de responsabilidade do sistema auxiliar, logo não ocorre neste algoritmo da interface.

No caso da ação de alteração, a tela (Figura 19) apresenta um simples campo de entrada no qual é possível escrever um novo valor de interesse para a vazão da solução e, após a digitação, deletá-lo do campo de digitação ou confirmar o valor e o enviar para o sistema.

Figura 19 - Tela de alteração do sistema supervisorio



Fonte: Autora, 2024.

Em ambas as ações, é realizada uma chamada para o algoritmo do sistema auxiliar o qual será responsável pela intermediação entre os comandos visuais e a obtenção ou alteração dos dados no Arduino.

5.1.2 Sistema Auxiliar

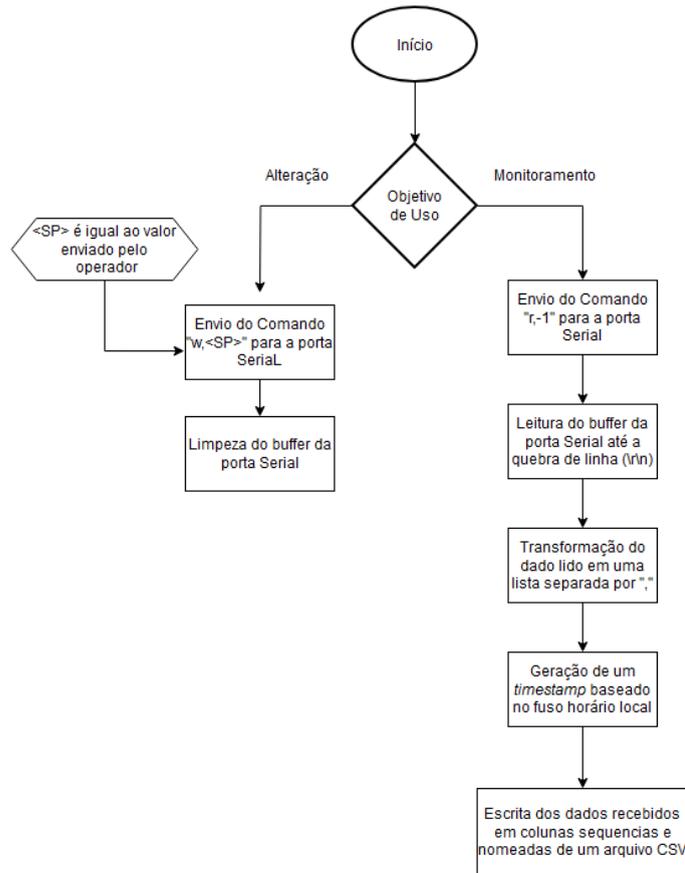
Após a leitura da ação de interesse do operador, o sistema auxiliar é acionado. Este sistema consiste de dois algoritmos que estão diretamente relacionados à interface gráfica.

O primeiro é um algoritmo referente à ação de monitoramento e possui três responsabilidades principais: realizar a comunicação com o Arduino através do envio de um comando para a porta serial de interesse, fazer o tratamento inicial dos dados recebidos (decodificação dos dados recebidos em ASCII e geração de duas variáveis referentes à data e hora de recebimento desses dados) e anexação dos dados tratados em uma linha do arquivo CSV (o qual será lido pelo algoritmo da interface gráfica).

Já o segundo é um algoritmo mais simples que apenas recebe da interface gráfica o novo valor desejado para a variável controlada e envia um comando para o Arduino com a ação desejada (escrita) junto do valor de interesse.

O algoritmo de tratamento de dados do *buffer* serial feito no Arduino foi programado para receber dois tipos de comandos de interação válidos: o primeiro é “r, -1” e se refere a um pedido feito pelo sistema auxiliar para que o Arduino envie as variáveis monitoradas, já o segundo é dado pelos caracteres “w” e “,” seguidos por um número de ponto flutuante qualquer tal que, em conjunto, significam para o algoritmo criado que o sistema auxiliar deseja que uma variável seja alterada. Um resumo deste algoritmo pode ser observado na Figura 20.

Figura 20 - Representação gráfica do algoritmo do sistema auxiliar



Fonte: Autora, 2024.

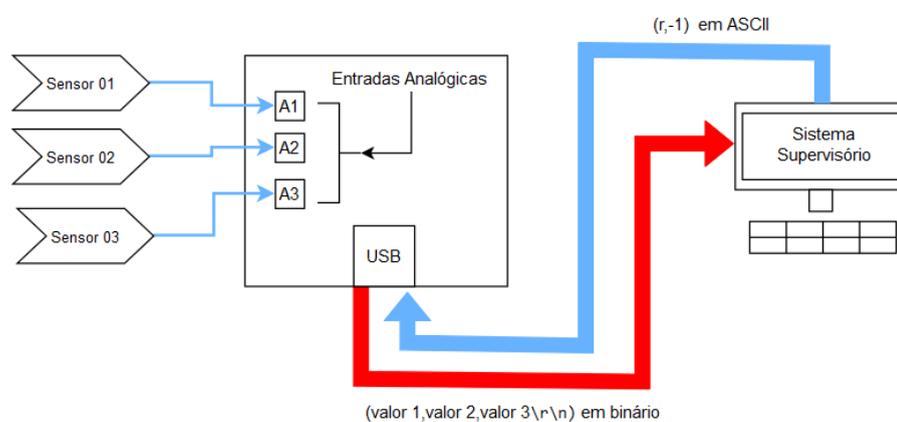
Toda transferência de dados realizada pelo sistema supervisorio para o Arduino se deu pela utilização dos métodos de escrita e leitura do buffer da porta serial disponibilizados pelo pacote pySerial o qual é compatível com a linguagem Python. Já a manipulação do arquivo CSV foi possibilitada pela utilização do módulo de CSV nativo da linguagem Python junto com o protocolo de gerenciamento de contextos da mesma linguagem de forma que o arquivo pudesse ser aberto, manipulado e apropriadamente fechado sem a necessidade de muitos comandos e sem o risco de vazamento de memória (tradução literal do termo em inglês, *memory leak*).

5.1.3 Código Integrado no Arduino

O código Arduino utilizado neste trabalho possui três tarefas principais: monitoramento das variáveis do processo através da calibração dos sensores e posterior leitura dos dados recebidos, comunicação assíncrona com o computador através da interface USART e manipulação da tensão de saída através da geração de um sinal PWM. A comunicação entre

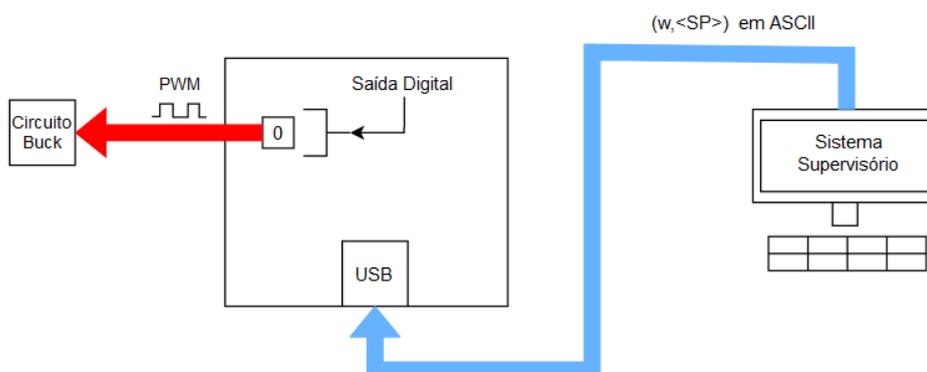
Arduino e o sistema supervisório é reproduzida de forma simplificada nas Figuras 21 e 22 abaixo:

Figura 21 - Comunicação de leitura de sensores entre o Arduino e o computador



Fonte: Autora, 2024.

Figura 22 - Comunicação de alteração de vazão entre o computador, o Arduino e o sistema



Fonte: Autora, 2024.

A tarefa de comunicação foi realizada através da criação de três funções customizadas. A primeira função teve como responsabilidades: a leitura dos dados presentes no *buffer* da porta serial e, enquanto este buffer não estivesse vazio, o armazenamento dos dados lidos em uma lista de tamanho limitado (de forma a evitar vazamento de memória).

A segunda função, entretanto, teve como responsabilidades: lidar com o formato dos dados lidos na porta serial através da separação dos elementos vindos antes e depois da vírgula e conversão dos valores posteriores à vírgula, inicialmente lidos como caracteres (estruturas textuais de 1 byte), em um número de ponto flutuante (estrutura armazenada em 4 bytes).

Ademais, após discernir a ação enviada antes da vírgula (“r” para leitura dos sensores ou “w” para alteração da vazão), uma terceira função é invocada dependendo da ação de

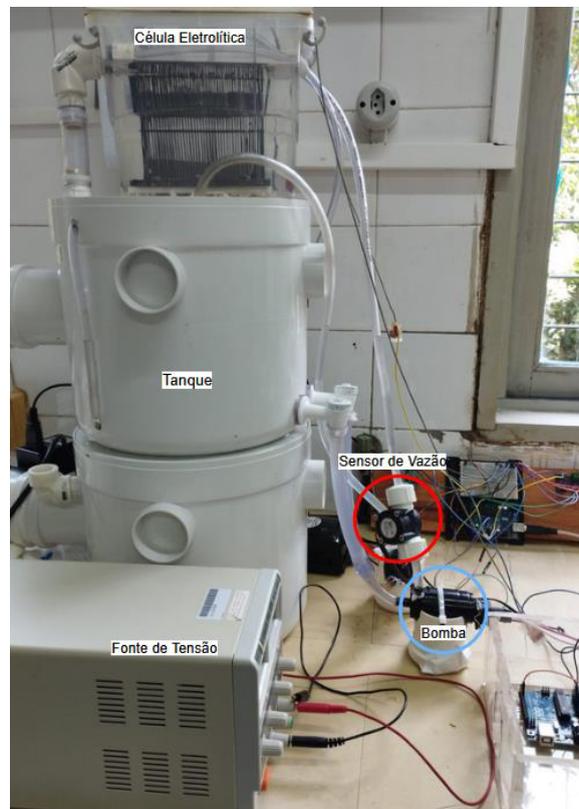
interesse. No caso do interesse pela leitura dos dados dos sensores, a função invocada escreve um valor textual do tipo *String* no *buffer* da porta serial, o qual será lido pelo sistema auxiliar. Já no caso do interesse pela alteração do valor da vazão no sistema, essa terceira função introduz o novo valor desejado em um algoritmo de conversão de tensão que será detalhado posteriormente.

5.2 Modelagem Matemática

Para que fosse possível implementar uma estratégia de controle eficiente, foi necessário antes conhecer a dinâmica do sistema através da identificação empírica do mesmo. Dessa forma, após a obtenção do modelo matemático apropriado, o sistema pôde ser simulado e, conseqüentemente, teve seu controlador adequadamente sintonizado.

A modelagem matemática se iniciou através do monitoramento das variáveis de tensão e vazão ao longo do tempo. O monitoramento foi possível através da alteração manual da tensão da bomba alimentada por uma fonte de corrente contínua e da leitura dos dados de vazão de alimentação da célula gerados por um sensor de vazão acoplado ao subsistema de escoamento. A disposição do sistema durante essa etapa além dos componentes relevantes está disposta na Figura 23:

Figura 23 - Disposição do sistema físico durante a medição



Fonte: Autora, 2024.

A manipulação da tensão foi realizada no intervalo entre 3,5V (a tensão mínima na qual a bomba conseguiu vencer as perdas de carga do sistema e efetivamente gerar alguma vazão) e 9,5V (a tensão na qual o sistema começou a transbordar, logo tornou-se inviável o incremento de vazão).

Após o processo de medição, os dados de tensão (variável independente) e vazão (variável dependente) foram submetidos a uma regressão linear simples e uma regressão polinomial de segunda ordem utilizando o software Excel.

Após o estudo da regressão, os dados obtidos foram lidos no formato de matriz no ambiente de desenvolvimento do MATLAB. Após a leitura da matriz, a mesma foi inserida no conjunto de ferramentas de identificação de sistema do mesmo software de forma a obter uma função de transferência adequada para a modelagem do sistema.

Seleção do Tipo de Modelo:

Escolha o tipo de modelo que melhor representa o comportamento do sistema, como ARX (Auto-Regressive with eXogenous inputs), BJ (Box-Jenkins), entre outros.

Identificação do Modelo:

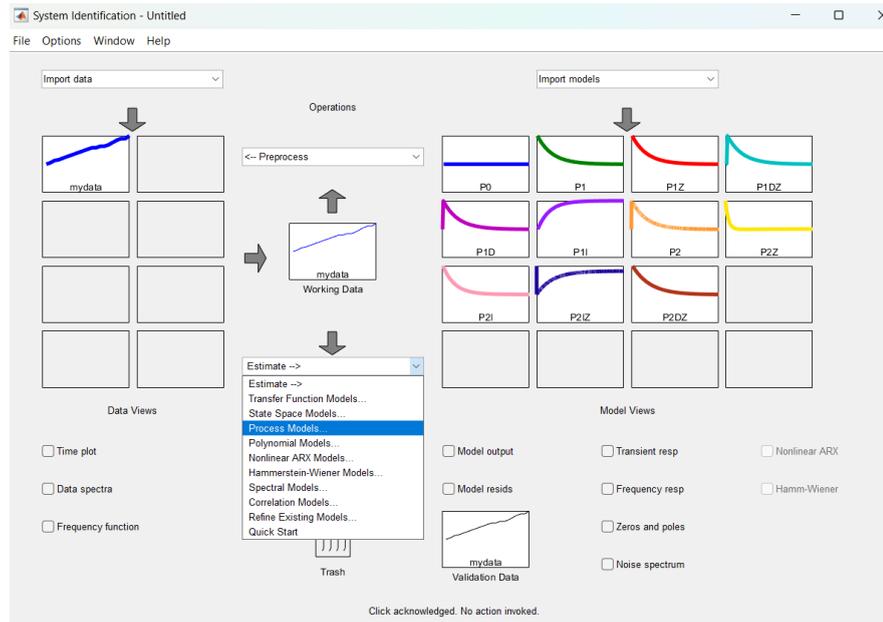
A ferramenta Procest utiliza algoritmos de identificação de sistemas para ajustar automaticamente os parâmetros do modelo aos dados experimentais, procurando o melhor ajuste possível.

A ferramenta de identificação utilizada foi o método *PROCEST*. Com esta, foram geradas algumas funções de transferência compostas pelos termos de ganho, constantes de tempo e tempo morto.

Cada função foi gerada sob considerações de modelagem diferentes baseadas no número de polos, número de zeros, presença ou não de tempo morto e comportamento similar ao de um sistema integrador ou não.

Todas as funções foram, então, comparadas com base na avaliação do seu Erro de Previsão Final (FPE, na sigla em inglês) e ajuste aos dados experimentais. Uma visão gráfica dessa ferramenta e das funções de transferência avaliadas para a modelagem do sistema pode ser observada na Figura 24. Nela a letra P está relacionada a cada processo (função de transferência), o número a seguir está relacionado a ordem da função de transferência e as letras Z, D e I são referentes, respectivamente, ao número de zeros, à consideração de tempo morto (*delay*, em inglês) e ao possível comportamento integrador do sistema.

Figura 24 - Identificação da função de transferência usando PROCEST



Fonte: Autora (2024).

Por fim, a função avaliada com melhor desempenho sob a óptica das medições de erro foi utilizada como base para a simulação simplificada do sistema utilizando o software SIMULINK disponível junto com as ferramentas numéricas do MATLAB.

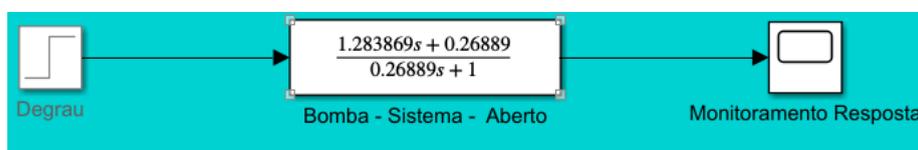
Esta simulação, portanto, foi utilizada para a realização da sintonia do controlador a ser implementado no sistema físico.

5.3 Sintonia do Controlador

De forma a obter os parâmetros mais adequados para o sistema, inicialmente três métodos clássicos de sintonia foram testados: os métodos de controle em malha aberta e malha fechada de Ziegler-Nichols e o método de controle em malha aberta de Cohen-Coon. Duas configurações do sistema diferentes foram submetidas a esses métodos: uma configuração do sistema com o subprocesso atuador como uma caixa preta e um outro com o subprocesso atuador descrito através do conversor Buck.

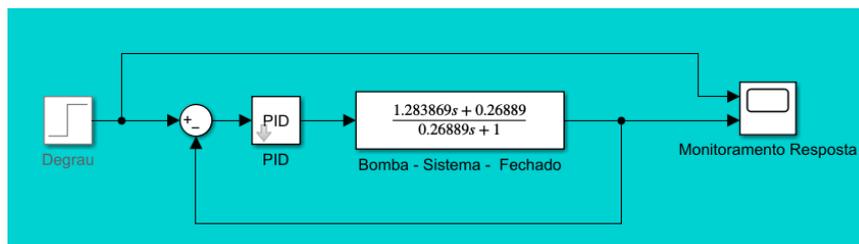
Os métodos de sintonia para o sistema simplificado em malha aberta e malha fechada foram obtidos através das disposições das simulações representadas, respectivamente, nas Figura 25 e 26.

Figura 25 - Simulação do sistema de bombeamento em malha aberta



Fonte: Autora (2024).

Figura 26 - Simulação do sistema de bombeamento em malha fechada



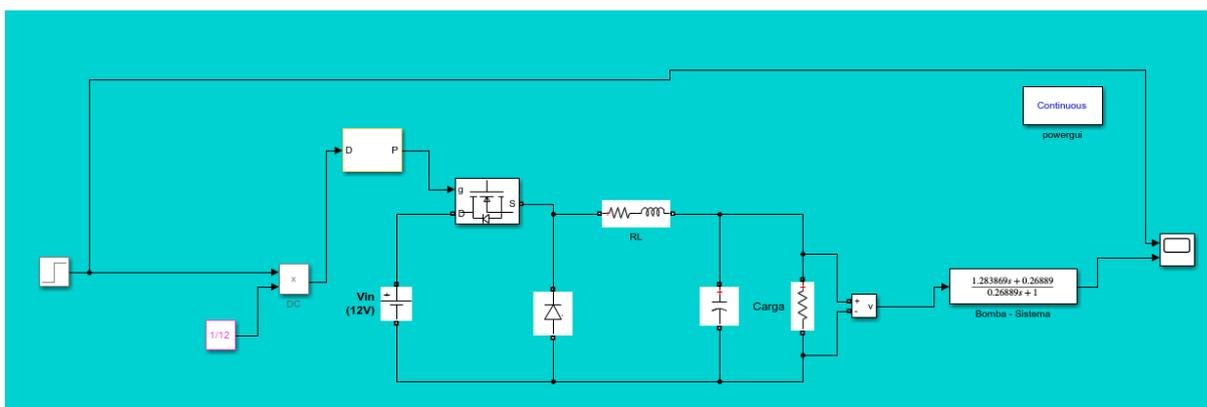
Fonte: Autora (2024).

Para possibilitar a obtenção das malhas simuladas para o sistema completo, foi necessário adicionar uma etapa de conversão do sinal de saída do PID em um sinal de *duty cycle* através da divisão do sinal de saída pela tensão que alimenta o circuito Buck (12V). Logo após essa conversão, o sinal obtido é transformado em um PWM.

É necessário converter o sinal analógico em um sinal digital PWM, pois, no circuito Buck, o sinal PWM controla a operação do transistor, alternando seu estado entre ligado e desligado em intervalos regulares. Assim, a proporção do tempo em que o transistor está ligado (seu *duty cycle*) determina a quantidade de energia média entregue à carga. Um *duty cycle* maior resulta em uma tensão de saída maior, enquanto um *duty cycle* menor resulta em uma tensão de saída menor.

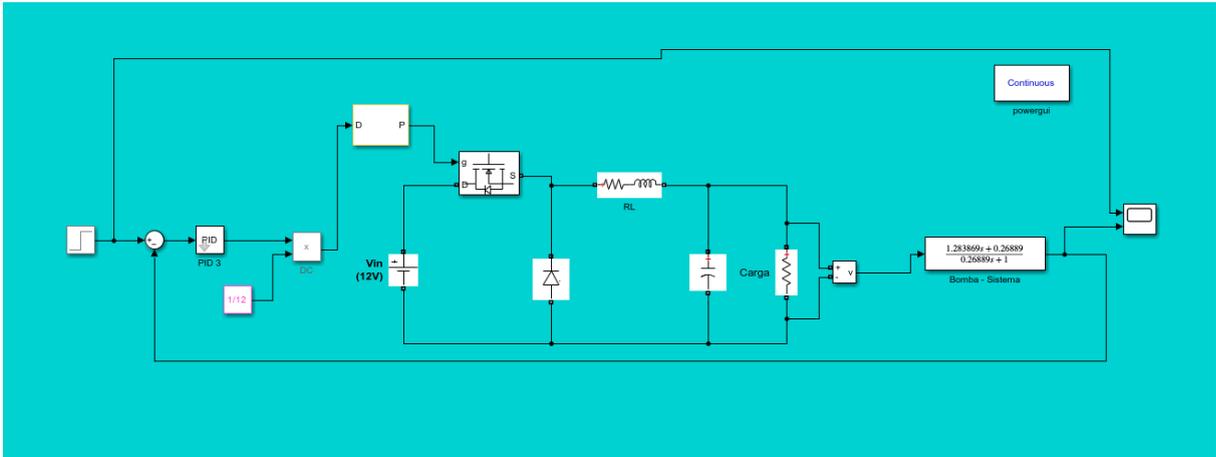
Esta simulação modificada pode ser observada sob operação em malha aberta e malha fechada, respectivamente, nas Figura 27 e 28.

Figura 27 - Simulação do sistema completo em malha aberta



Fonte: Autora (2024).

Figura 28 - Simulação do sistema completo em malha fechada



Fonte: Autora (2024).

Os primeiros métodos avaliados foram os métodos da curva da reação do processo em malha aberta de Ziegler-Nichols e Cohen-Coon. Para sua execução, inicialmente foi gerada uma perturbação degrau na entrada do sistema. Logo em sequência, a resposta do sistema foi observada com base nos parâmetros da planta descritos na Figura 9.

Assim, esses parâmetros foram utilizados nas relações descritas pela Tabela 01 para determinar os coeficientes mais adequados, segundo cada um dos métodos, para o controlador PID do sistema.

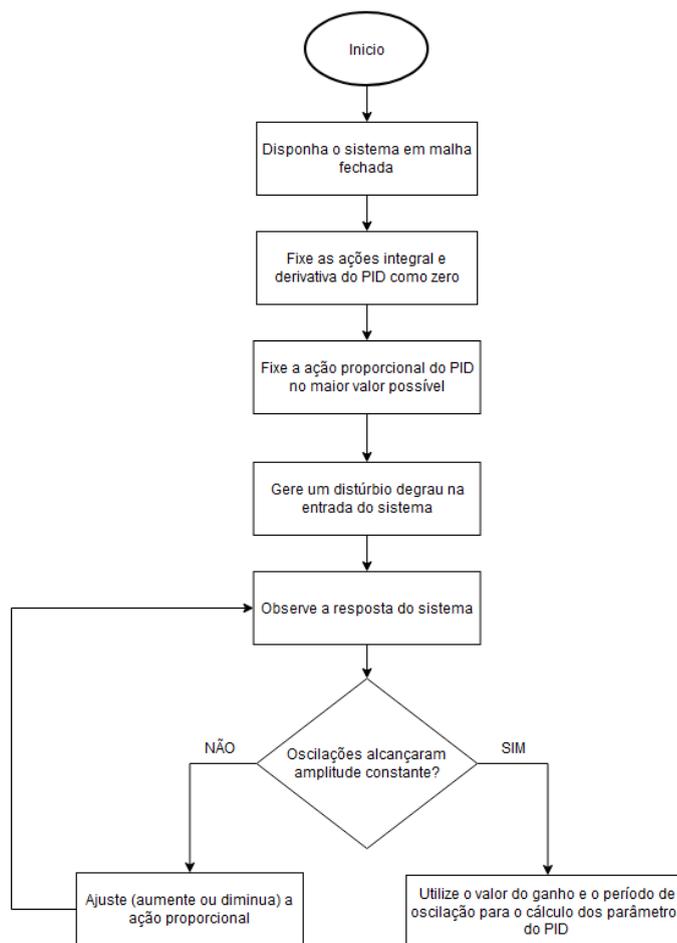
Tabela 1 - Coeficientes do controlador PID para os métodos de Ziegler-Nichols e Cohen-Coon em malha aberta

Método	aKp	Ti/L	Td/L
Ziegler-Nichols	1,2	2	L/2
Cohen-Coon	$1,35 * (1 + [(0,18 * \tau) / (1 - \tau)])$	$(2,5 - 2 * \tau) / (1 - 0,39 * \tau)$	$(0,37 - 0,37 * \tau) / (1 - 0,81 * \tau)$

Fonte: Autora, 2024.

Logo após a avaliação do método de Cohen-Coon, o sistema foi disposto em malha fechada para que o método de malha fechada de Ziegler-Nichols (cujo algoritmo é descrito na Figura 29) pudesse ser executado.

Figura 29 - Método de Ziegler-Nichols em Malha Fechada



Fonte: Autora (2024).

Com o sistema ainda em malha fechada, foi executado o método de tentativa e erro, no qual os valores dos ganhos proporcional, integral e derivativo foram ajustados continuamente até que fosse obtida uma resposta adequada (segundo a estratégia de controle adotada) a um distúrbio degrau.

Assim, os resultados de todos os métodos foram comparados com base em alguns dos parâmetros de desempenho de um controlador PID (erro em estado estacionário, tempo de assentamento, instantes de pico), de forma que os coeficientes do método de melhor desempenho pudessem ser aplicados no algoritmo de controle PID criado (descrito no Apêndice B).

Dessa forma, apesar de não ter sido possível operar o sistema com controlador devido a uma falha operacional no sistema de bombeamento, o sistema de controle completo (representado no Apêndice A) foi obtido através da união entre os algoritmos de manipulação do *buffer* da porta serial (descritos nos Apêndices C e D) e o algoritmo de controle PID - cuja tensão resultante é utilizada como a entrada do algoritmo de manipulação do circuito Buck.

6. RESULTADOS E DISCUSSÃO

Durante a etapa inicial do processo de modelagem, a manipulação de tensão gerou os seguintes valores, descritos na Tabela 2, de tensão e vazão no sistema. Experimentalmente, foi observado um melhor intervalo de desempenho do sistema sob a operação da bomba entre 3,5V e 5,5V.

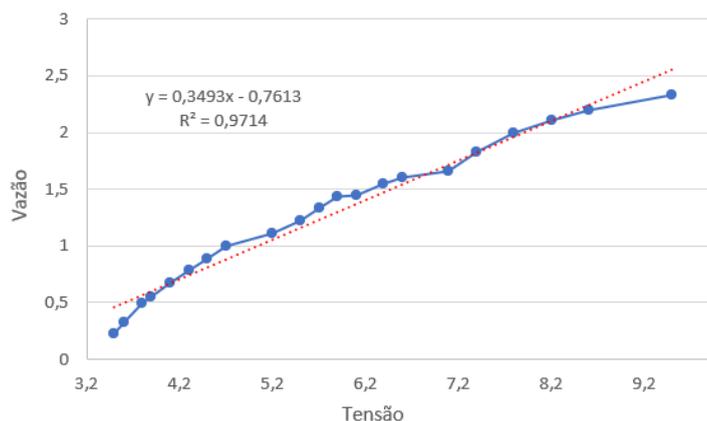
Tabela 2 - Medições de tensão e vazão no sistema

Tensão	Vazão	Tensão	Vazão
3,5	0,22	5,7	1,33
3,6	0,33	5,9	1,44
3,8	0,495	6,1	1,45
3,9	0,55	6,4	1,55
4,1	0,67	6,6	1,6
4,3	0,78	7,1	1,66
4,5	0,89	7,4	1,83
4,7	1	7,8	2
5,2	1,11	8,2	2,11
5,5	1,22	8,6	2,2
		9,5	2,33

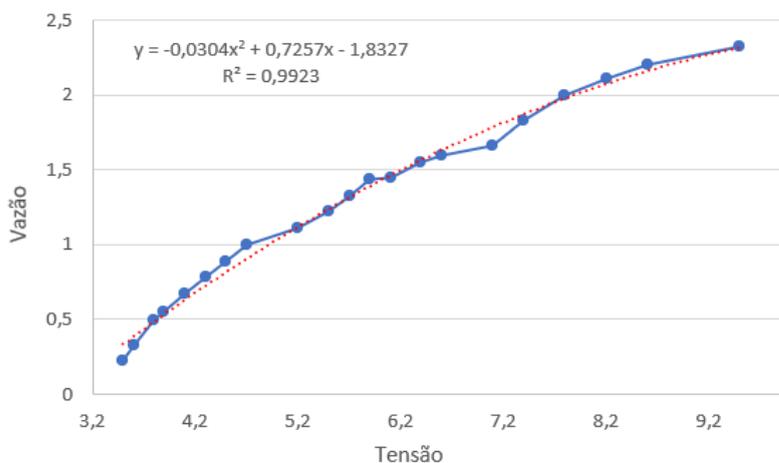
Fonte: Autora, 2024.

Após o processo de medição, os dados foram submetidos a regressões utilizando o software Excel. As equações e valores de R-quadrado da regressão linear simples e da regressão polinomial de segunda ordem podem ser visualizados, respectivamente, nas Figuras 30 e 31.

Figura 30 - Regressão linear da relação entre vazão e tensão



Fonte: Autora, 2024.

Figura 31 - Regressão polinomial de segunda ordem da relação entre vazão e tensão

Fonte: Autora, 2024.

Observando o comportamento do gráfico, foi possível identificar a dinâmica de alta correlação entre as variáveis manipulada e controlada. Comparando os valores de R-quadrado das duas regressões, foi possível observar valores de R-quadrado altos para ambos, entretanto há uma tendência à não linearidade na relação entre a vazão e a tensão seguindo uma equação polinomial de segundo grau.

De forma geral, após os processos de regressão, foram obtidos valores de R-quadrado próximos a 1. Por isso, foi possível assumir uma alta correlação entre as variáveis manipulada e controlada. Ademais, analisando a resposta da vazão às alterações de tensão ao longo do tempo, tornou-se evidente que a resposta do sistema possui um valor de tempo morto muito pequeno.

Após o estudo da regressão, foram geradas funções de transferência utilizando a ferramenta de identificação de sistemas no MATLAB. Um resumo das mesmas assim como seus parâmetros de erro correspondentes pode ser observado na Tabela 3.

Tabela 3 - Condições das funções de transferência estimadas

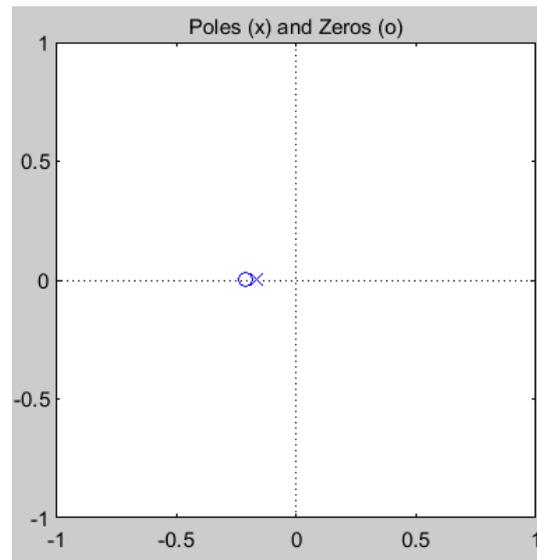
Polos	Zeros	Delay	Integrador	Fit	FPE
0	0	não	não	61,03%	0,0613
1	0	não	não	92,47%	0,0028
1	1	não	não	93,94%	0,0020
1	1	sim	não	91,47%	0,0034
1	0	sim	não	92,47%	0,0031
1	0	não	sim	88,77%	0,0068
1	1	não	sim	87,67%	0,0091
2	0	não	não	86,73%	0,0105
2	1	não	não	93,63%	0,0027
2	1	sim	não	92,41%	0,0042
2	0	sim	não	31,26%	0,3119
2	0	não	sim	75,23%	0,0300
2	1	não	sim	88,60%	0,0070
2	1	sim	sim	8,51%	0,6848

Fonte: Autora, 2024.

Tanto pela avaliação do valor de ajuste (*fit* em inglês) quanto pela comparação dos valores de FPE, é fácil ver que a função de transferência de primeira ordem, com um zero no eixo real, sem tempo morto (*delay*, em inglês) e sem comportamento integrador presente na terceira linha da Tabela 03 acima apresentou a melhor fidelidade na descrição da dinâmica entre as variáveis manipulada e controlada desse sistema.

Essa função de transferência, por sua vez, tem seu comportamento descrito pela Equação 31 abaixo, enquanto a sua estabilidade é ditada pela localização de seu polo e de zero, apresentados na Figura 32.

$$G(s) = K_p * \left[\frac{1 + T_z * s}{1 + T_p * s} \right] = 0,26889 * \left[\frac{1 + 4,7747 * s}{1 + 6,2099 * s} \right] \quad (31)$$

Figura 32 - Localização dos polos e zeros da função de transferência escolhida

Fonte: Autora, 2024.

Observando a Figura 32, é possível notar que, como há um zero e um polo apenas no lado esquerdo do eixo real, o sistema descrito por essa função de transferência tende fortemente à estabilidade mesmo operando sob a configuração de malha aberta.

Tendo em mente as informações sobre o sistema descritas até então, foram iniciadas as simulações da operação do sistema - utilizando o SIMULINK - de forma a sintonizar um controlador PID de alto desempenho. Nesses processos o sistema foi simulado sob duas configurações: a configuração simplificada com o processo de manipulação da tensão descrito como uma caixa preta e o uma configuração completa com a descrição detalhada da manipulação de tensão através da ação do circuito Buck.

Assim, os processos de sintonia em malha aberta utilizando os métodos de Ziegler-Nichols e Cohen-Coon foram iniciados pela identificação da resposta do sistema a uma perturbação degrau, essa observação identificou os parâmetros de processo apresentados na Tabela 04.

Tabela 4 – Parâmetros de processo após a resposta a uma perturbação degrau

Parâmetros	Sistema Simplificado	Sistema Completo
$r(\infty)$	4,773	9,529
a	0,641	0,219
L	0,600	0,080
T	4,470	3,488
τ	0,118	0,022

Fonte: Autora, 2024.

Em sequência, os parâmetros obtidos foram utilizados nas relações de cada metodologia visando a obtenção dos coeficientes de um controlador PID sintonizado. Os coeficientes dados

para o sistema simplificado e o sistema completo podem ser vistos, respectivamente, nas tabelas 05 e 06.

Tabela 5 - Parâmetros de um controlador PID para o sistema simplificado

Método	Kp	Ti	Td
Ziegler-Nichols	1,873	1,200	0,180
Cohen-Coon	1,002	1,424	0,216

Fonte: Autora, 2024.

Tabela 6 - Parâmetros de um controlador PID para o sistema completo

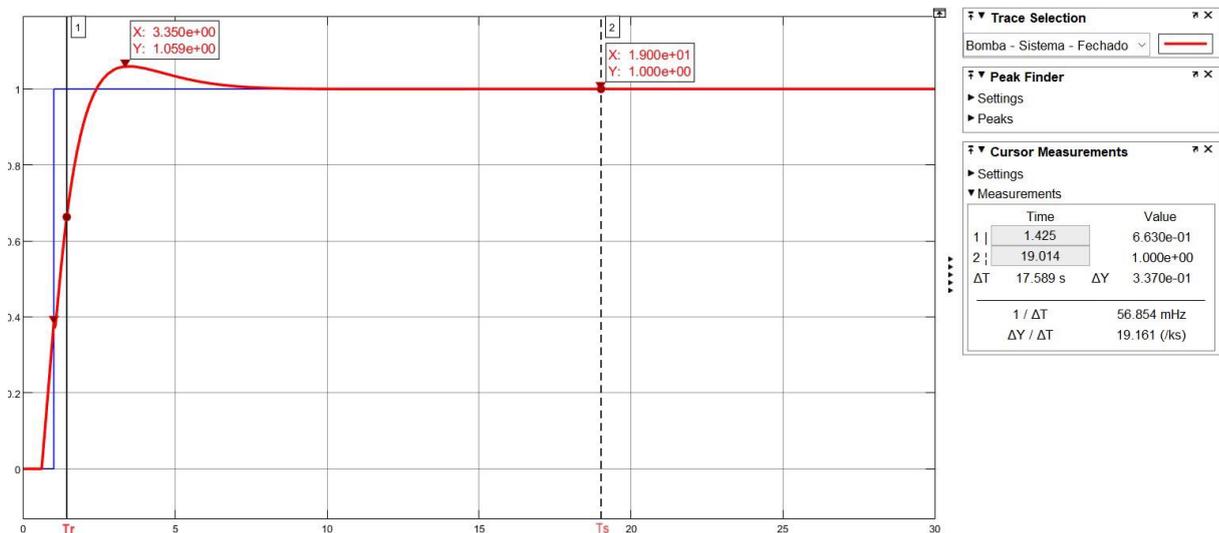
Método	Kp	Ti	Td
Ziegler-Nichols	5,491	0,160	0,003
Cohen-Coon	0,303	0,198	0,029

Fonte: Autora, 2024.

Cada um dos modelos teve seus coeficientes utilizados em um controlador PID que, por sua vez, foi adicionado ao sistema simulado anteriormente de forma a configurar a malha como operada sob retroalimentação. Assim, ao submeter esta nova simulação a um distúrbio degrau, foi possível analisar a resposta do sistema já controlado pelo controlador sintonizado.

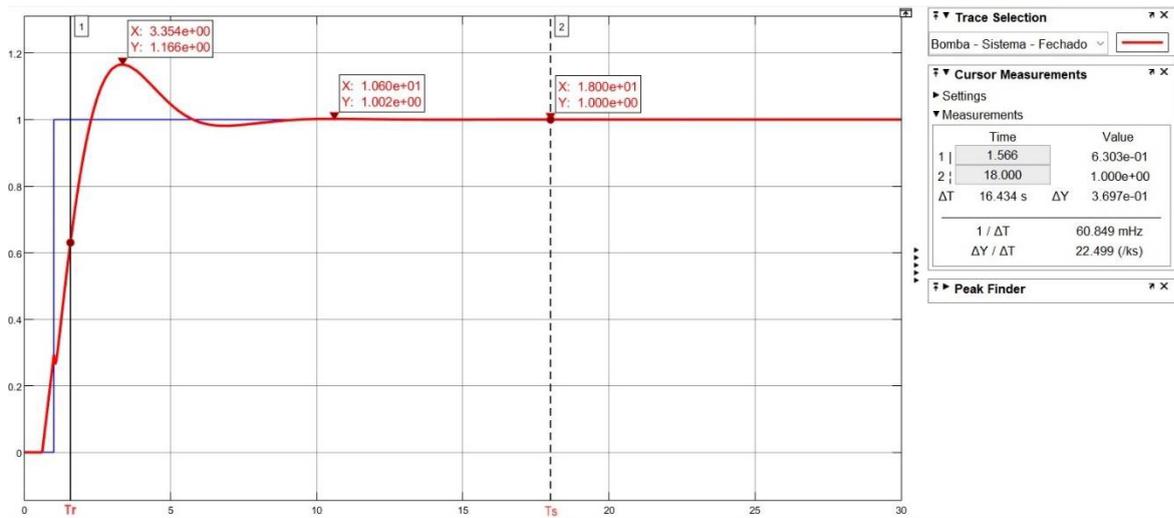
As respostas apresentadas pelo sistema simplificado controlado pelos coeficientes de Ziegler-Nichols e Cohen-Coon podem ser observadas, respectivamente, nas Figuras 33 e 34.

Figura 33 - Resposta ao degrau do sistema simplificado sintonizado Ziegler-Nichols (malha aberta)



Fonte: Autora, 2024.

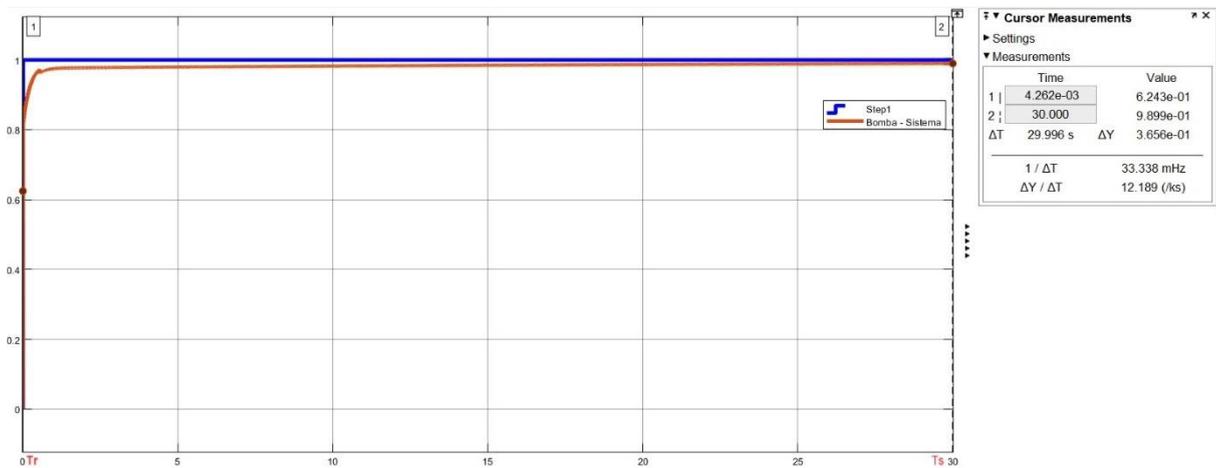
Figura 34 - Resposta ao degrau do sistema simplificado sintonizado por Cohen-Coon



Fonte: Autora, 2024.

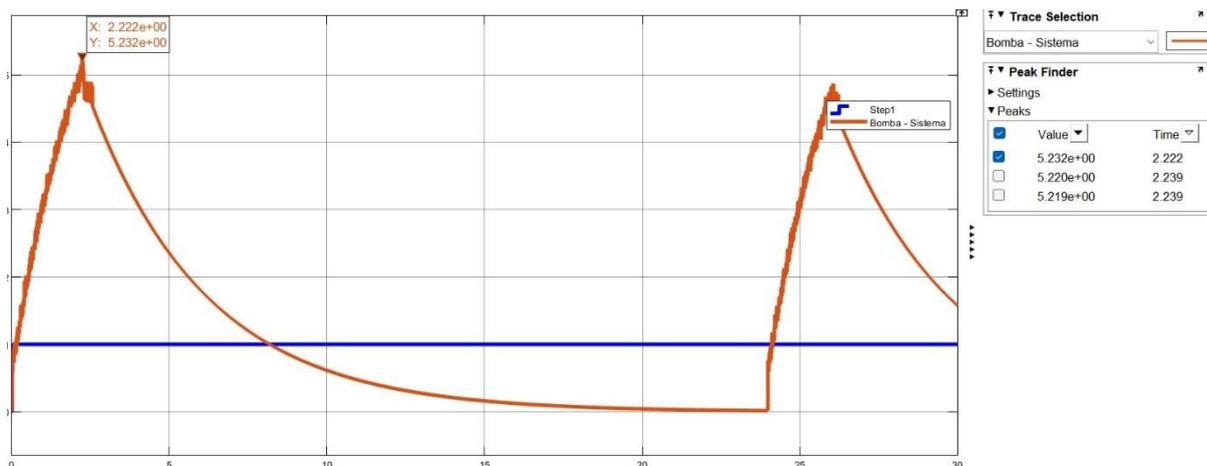
Já as respostas apresentadas pelo sistema completo controlado pelos coeficientes dos mesmos métodos citados acima podem ser observadas, respectivamente, nas Figuras 35 e 36.

Figura 35 - Resposta ao degrau do sistema completo sintonizado Ziegler-Nichols (malha aberta)



Fonte: Autora, 2024.

Figura 36 - Resposta ao degrau do sistema completo sintonizado por Cohen-Coon



Fonte: Autora, 2024.

Comparando os comportamentos das respostas da representação simplificada do sistema dados pelos dois métodos, é possível ver que – considerando uma estratégia de controle que visa a diminuição de *overshoot* e a velocidade na resposta - o desempenho do controlador dado pelo método de Ziegler-Nichols em malha aberta foi superior ao dado por Cohen-Coon, visto que, apesar de ambos alcançarem valores de erro estacionário muito pequenos, o primeiro método exibiu menores valores de *overshoot* (1,059 em comparação à 1,66) e tempo de resposta (1,425 min em comparação com 1,566 min) tendo apenas um tempo de acomodação mais lento (1 min de diferença).

No caso da representação do sistema completo, o método de Ziegler-Nichols novamente obteve melhor desempenho, entretanto o mais notório é que a resposta por Cohen-Coon apresentou um comportamento bastante peculiar com valores de erro muito altos.

Esses resultados podem ser explicados pelo fato de que o método de Cohen-Coon foi idealizado como uma evolução para o desempenho do método de Ziegler-Nichols em malha aberta para sistemas com altos valores de tempo morto. Logo, como o sistema de interesse deste trabalho não apresenta considerável tempo morto, este método de sintonia não é adequado para o mesmo.

Entretanto, um outro ponto considerável é que ambos os métodos clássicos foram idealizados para acomodar a dinâmica de atuadores analógicos (por exemplo, atuadores pneumáticos) e, portanto, resultam em comportamentos pouco previsíveis para o controle digital. Nesse caso, as relações deixam de representar considerações físicas e se tornam manipulações algébricas de pouca confiabilidade.

Nesse caso, até mesmo o bom desempenho do sistema sob o controle por Ziegler-Nichols, é mais facilmente explicado pelo fato de que as relações tabeladas neste método geram valores maiores para sistemas cujas respostas ao degrau são curvas sigmóides de menor declive.

Enfim, após os procedimentos sob malha aberta, o sistema foi disposto sob malha fechada e foi realizado o procedimento descrito na Figura 29 acerca do método de sintonia de Ziegler-Nichols em malha fechada.

Entretanto, como já foi abordado nesta seção, o sistema modelado neste trabalho não tende a oscilações. Assim, mesmo utilizando valores de ganho proporcional muito altos, não houve oscilações estáveis (com amplitude constante) na resposta do sistema. Dessa forma, foi possível afirmar que o método de Ziegler-Nichols em malha fechada é também incompatível com o sistema de interesse.

Deste modo, o último método a ser utilizado para a sintonia do sistema foi o método de tentativa e erro. Nesse método, o sistema foi operado em malha fechada e a sua resposta ao degrau unitário foi observada para valores de ganhos diferentes.

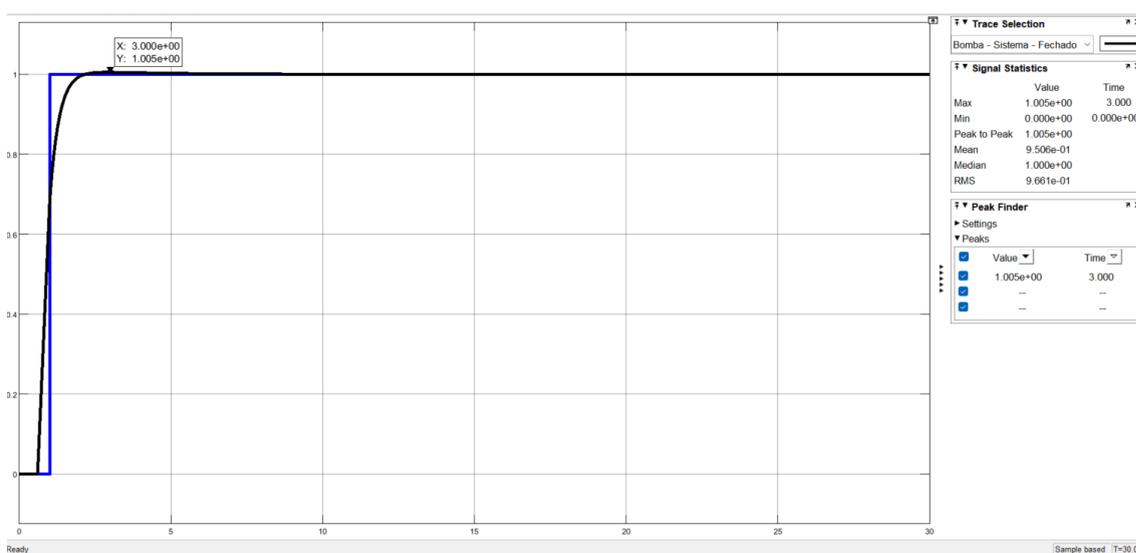
O melhor desempenho para o sistema simplificado foi encontrado para os coeficientes descritos na Tabela 07 e a resposta do sistema correspondente aos mesmos pode ser observada na Figura 37.

Tabela 7 - Coeficientes das ações de um PID dados por Tentativa e Erro para o sistema simplificado

Método	K _p	T _i	T _d
Tentativa e Erro	8	2,67	0

Fonte: Autora, 2024.

Figura 37 - Resposta ao degrau do sistema simplificado sintonizado por Tentativa e Erro



Fonte: Autora, 2024.

Já para o sistema completo, o melhor desempenho (apresentado na Figura 38) durante a operação no simulador foi encontrada para os parâmetros descritos na Tabela 08.

Tabela 8 - Coeficientes das ações de um PID dados por Tentativa e Erro para o sistema completo

Método	Kp	Ti	Td
Tentativa e Erro	6	1,3	0,001

Fonte: Autora, 2024.

Figura 38 - Resposta ao degrau do sistema completo sintonizado por Tentativa e Erro



Fonte: Autora, 2024.

É possível ver que o desempenho do sistema sob um controlador PI ou PID com pouca ação derivativa foi superior até mesmo ao método de sintonia mais adaptado que foi utilizado (Ziegler-Nichols em malha aberta). Isto pode ser explicado pelo rápido tempo de resposta do sistema, o que o torna menos propenso a *overshoot*.

Dessa forma, as ações proporcional e integral se mostram suficientes para uma estratégia de controle simples.

7. CONCLUSÃO

No presente trabalho foi proposto o desenvolvimento de um controlador PID na plataforma Arduino para controle de vazão de um tanque. Visando atingir o objetivo proposto, foi desenvolvido um sistema de monitoramento e controle abrangente feito por três componentes majoritários.

Inicialmente, foi desenvolvido um sistema supervisor cuja função principal era a de ser uma interface gráfica amigável para possibilitar tanto o monitoramento das variáveis de processo quanto a alteração do valor da vazão. Esse sistema apresentou três telas principais as quais esperavam e respondiam a comandos do operador. Em seguida, foi desenvolvido um conjunto de algoritmos auxiliares o qual foi responsável pela intermediação entre a comunicação da interface gráfica com os algoritmos de controle feitos em Arduino.

Por fim, foi desenvolvido o conjunto de algoritmos produzidos em Arduino cujas funções principais eram de comunicação com a porta serial para leitura e transmissão de dados do sistema físico além do controle da vazão através da manipulação direta da tensão de uma bomba. Nele, todas as funções criadas visaram garantir dois fluxos de informação sem interrupções: a comunicação dos dados dos sensores e a leitura do novo valor desejado para vazão, sua conversão para um valor equivalente de tensão (utilizando o controlador PID) e a alimentação dessa nova tensão à bomba através de um circuito Buck.

Acerca do conhecimento da dinâmica do sistema para a produção de um algoritmo de controle de bom desempenho, foram realizados procedimento de identificação de sistemas e simulação dinâmica utilizando ferramentas da plataforma MATLAB e, com esses, foi possível observar um comportamento de sistema que pôde ser modelado por uma função de transferência de primeira ordem, sem tempo morto considerável e consideravelmente estável. Finalmente, esse modelo foi submetido à métodos de sintonia clássicos e, entre eles, foi observado desempenho similar entre o método de Ziegler-Nichols em malha aberta e o método de tentativa e erro. Foi possível também observar que o sistema apresenta bom desempenho tanto sobre o controle PID quanto sobre controle PI.

Em trabalhos futuros é sugerido realizar a comparação entre o desempenho do sistema controlador pelo controlador de melhor desempenho obtido neste trabalho através de simulação e o desempenho real do sistema sob a ação do mesmo controlador. Ademais, uma abordagem bastante interessante para outros trabalhos é a implementação de conceitos de controle digital para uma modelagem mais robusta desse sistema e, conseqüentemente, a definição de um sistema de controle melhor adaptado às necessidades do mesmo.

REFERÊNCIAS

ÅSTRÖM, Karl Johan; HÄGGLUND, Tore. **PID Controllers: Theory, Design, and Tuning**. North Carolina, USA: Isa - The Instrumentation, Systems And Automation Society, 1995.

BOAL, Andrew K. On-Site Generation of Disinfectants. **The National Environmental Services Center**. West Virginia, EUA., 2009.

CAMPOS, M. C., TEIXEIRA, Herbert C. G. **Controles Típicos de Equipamentos e Processos Industriais**. São Paulo: Blucher, 2006.

CORRIPIO. **Tuning of Industrial Control Systems**. Editora ISA - Instrument Society of America, EUA., 1990.

CHIEN et al., **On the Automatic Control of Generalized Passive Systems**. Trans. ASME, 1952.

DESBOROUGH, L., MILLER, R. **Increasing Customer Value of Industrial Control Performance Monitoring—Honeywell's Experience**. 2000.

HIRZEL, Timothy. **Basics of PWM (Pulse Width Modulation)**. Disponível em: <https://docs.arduino.cc/learn/microcontrollers/analog-output/>. Acesso em: 05 jan. 2024.

ISERMANN, Rolf; MÜNCHHOF, Marco. Identification of Dynamic Systems. [S. l.]: Springer Berlin Heidelberg, 2011. DOI 10.1007/978-3-540-78879-9. Disponível em: <http://dx.doi.org/10.1007/978-3-540-78879-9>.

KASHIWAGI, Hiroshi. NONPARAMETRIC SYSTEM IDENTIFICATION. In: UNBEHAUEN, Heinz D. (ed.). **CONTROL SYSTEMS, ROBOTICS AND AUTOMATION: modeling and system identification**. 6. ed. Kumamoto, Japão: Eolss Publications, 2009. Disponível em: <https://www.eolss.net/sample-chapters/c18/E6-43-10-01.pdf>. Acesso em: 19 dez. 2024.

LATHI, Bhagwandas Pannalal; GREEN, Roger A.. **Linear systems and signals**. Nova York, EUA: Oxford University Press, 2018. 1010 p.

LEBLANC, S, COUGHANOWR, D. **Process Systems Analysis and Control**. McGraw-Hill Higher Education., 2009.

LUYBEN. **Process Modelling, Simulation and Control for Chemical Engineers**. Editora McGraw-Hill, NY, EUA. 1990.

NISE, S.. **Control Systems Engineering**. Jefferson City. John Wiley & Sons, Inc., 2011.

OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 3. ed. Rio de Janeiro: Ltc, 1997.

OSBORNE, Adam. An Introduction to Microcomputers: volume 1: basic concepts. Berkeley California USA: Osborne-McGraw Hill, 1980.

PETRY, Clóvis Antônio; BARBI, Prof. Ivo. **Introdução aos Conversores CC-CC**. Florianópolis: Universidade Federal de Santa Catarina, 2001. 17 p. Disponível em: https://professorpetry.com.br/Bases_Dados/Apostilas_Tutoriais/Introducao_Conversores_CC_CC.pdf. Acesso em: 18 jan. 2024.

SEBORG, E., MELLICHAMP, DALE, E., D. **Process Dynamics and Control**. John Wiley & Sons., 1989.

V. Viswanatha and R. V. S. Reddy, "Digital control of buck converter using arduino microcontroller for low power applications," 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon). IEEE, Agosto. 2017. doi: 10.1109/smarttechcon.2017.8358412.

ZIEGLER, J.G, NICHOLS, N. B. Optimum settings for automatic controllers, 1942.

APÊNDICES

APÊNDICE A - Captura de tela referente ao sistema de controle completo

```
control_buck_and_flow  ReadMe.adoc  ▼
1 float setpoint = 2.50; //Variavel que armazena o valor do set point padrao de tensao
2 byte saidaAnalogica = 0; //Variavel que armazena o valor do PWM (0 a 255)
3 unsigned long tempoant = 0; //Variavel que armazena um valor de tempo
4 float tensao = 0.0; //Variavel que armazena um valor de temperatura
5 //variaveis para PID
6 double outputTensaoPID;
7 double somaErro, erroAnt;
8 double kp, ki, kd;
9
10 void setup() {
11     pinMode(5, OUTPUT);
12     Serial.begin(9600);
13     analogWrite(5, saidaAnalogica);
14     tempoant = micros(); //Registra um tempo
15 }
16
17 void loop() {
18     //BLOCO PARA LEITURA DE TENSAO A CADA 1 MILISSEGUNDOS
19     unsigned long tempo = micros();
20     if(tempo > (tempoant+100)){ //compara com o tempo anterior
21         tensao = analogRead(A0)*(5.2/1023.0);
22         tensao *= 5.72;
23         tensao += 0.3;
24         readBuffer();
25         serialResponse();
26         PID_flow(tempo){
27             buckControl();
28             tempoant = tempo; //recomeca a contagem de tempo
29         }
30     }
31
32     void readBuffer(){ } //realiza a leitura do buffer serial
33
34     void buckControl(){ } //controle P do circuito Buck
35
36     void serialResponse(){ } //altera o valor de SETPOINT para o desejado pelo operador
37
38     void PID_flow(long tempo){ }
39 }
90
```

Fonte: Autora (2024).

APÊNDICE B - Captura de tela referente ao código de controle PID feito no Arduino

```
void PID_flow(long tempo){
    double deltaT = (double)(tempo - tempoant);
    double error = setpoint - Vazao;
    somaErro += (error * deltaT);
    double dErro = (error - erroAnt) / deltaT;
    tensao = kp * error + ki * somaErro + kd * dErro;
    erroAnt = error;
}
```

Fonte: Autora (2024).

APÊNDICE C - Captura de tela referente ao código de leitura do *buffer*

```

void readBuffer(){
    inputSeveral[0] = 0;
    maxChars = buffSize - 1;
    byte charCount = 0;
    byte ndx = 0;

    if(Serial.available() > 0){
        while (Serial.available() > 0) {
            if (ndx > maxChars - 1) {
                ndx = maxChars;
            }
            inputSeveral[ndx] = Serial.read();
            ndx ++;
            charCount ++;
        }
        if (ndx > maxChars) {
            ndx = maxChars;
        }
        inputSeveral[ndx] = 0;
    }
} //realiza a leitura do buffer serial

```

Fonte: Autora (2024).

APÊNDICE D - Capturas de tela referentes ao código de tratamento dos dados do *buffer*

```

void serialResponse(){
    char * partOfString;
    partOfString = strtok(inputSeveral, ",");

    strcpy(inputCsvString, partOfString);
    partOfString = strtok(NULL, ",");
    inputFloat = atof(partOfString);

    if(inputCsvString[0] == 'r'){
        Serial.print(tensao);
        Serial.print(",");
        Serial.print(corrente);
        Serial.print(",");
        Serial.print(analogRead(sensorACS712));
        Serial.print(",");
        Serial.print(calculoVazao);
        Serial.print(",");
        Serial.println(contaPulso);
    }
}

void serialResponse(){
    char * partOfString;
    partOfString = strtok(inputSeveral, ",");

    strcpy(inputCsvString, partOfString);
    partOfString = strtok(NULL, ",");
    inputFloat = atof(partOfString);

    if(inputCsvString[0] == 'w'){
        setpoint = inputFloat;
    }
} //altera o valor de SETPOINT para o desejado pelo operador

```

Fonte: Autora (2024).