

**UNIVERSIDADE FEDERAL DE MATO GROSSO
CAMPUS VÁRZEA GRANDE**

Gustavo Caetano de Souza

**ESTUDO COMPARATIVO DE ALGORITMOS DE
CONTROLE IMPLEMENTADOS EM HARDWARE DE
BAIXO CUSTO PARA CONTROLE DE TEMPERATURA EM
BANCADA DIDÁTICA**

**CUIABÁ
2019**

GUSTAVO CAETANO DE SOUZA

ESTUDO COMPARATIVO DE ALGORITMOS DE
CONTROLE IMPLEMENTADOS EM HARDWARE DE
BAIXO CUSTO PARA CONTROLE DE TEMPERATURA
EM BANCADA DIDÁTICA

Trabalho de Conclusão de Curso sub-
metido à Universidade Federal de Mato
Grosso, como requisito necessário para
obtenção do grau de Bacharel em En-
genharia de Controle e Automação

Cuiabá, Julho de 2019

UNIVERSIDADE FEDERAL DE MATO GROSSO

GUSTAVO CAETANO DE SOUZA

Esta Monografia foi julgada adequada para a obtenção do título de Bacharel em Engenharia de Controle e Automação, sendo aprovada em sua forma final pela banca examinadora:

Orientador(a): Prof. Me. Daniel Miranda
Cruz
Universidade Federal de Mato Grosso -
UFMT

Prof. Me. Cinthia Serenotti Brigante
Universidade Federal de Mato Grosso -
UFMT

Prof. Me. Bruno Silva Marció
Universidade Federal de Mato Grosso -
UFMT

Prof. Dr. Raoni Florentino da Silva Teixeira
Universidade Federal de Mato Grosso -
UFMT

Cuiabá, 20 de Agosto de 2019

Agradecimentos

Agradeço à toda minha família, amigos, professores e pessoas que ajudaram na realização desse trabalho. Sou imensamente grato pela paciência e incentivo.

Resumo

A dissertação apresentada neste documento, visa o desenvolvimento e a comparação de algoritmos clássicos de controle em hardwares de baixo custo em um protótipo de planta didática de temperatura. Para o monitoramento do processo e atuação sobre as variáveis de estado, escolheu-se sensores com preços acessíveis, proporcionando a prototipagem uma planta didática com pouco recurso financeiro. Os algoritmos de controle clássicos foram desenvolvidos baseados em modelos já estabelecidos na literatura referentes ao controlador PID 2DOF e a realimentação de estados com seguimento de referência com integradores. A implementação dessas estratégias deu-se no Arduino, onde alcançou-se resultados positivos no controle do processo selecionado frente as limitações deste hardware, visto a necessidade de uma alta taxa de amostragem para um melhor desempenho do controlador.

Palavras-chave:. Controle de temperatura, PID 2DOF, Realimentação de Estados e Arduino.

Abstract

The dissertation presented in this paper aims at the development and comparison of classical control algorithms using low cost hardware in a prototype didactic temperature process. For process monitoring and action on state variables, We opted for affordable sensors, providing a didactic plant with little financial resources. The classic control algorithms were developed based on models already established in the literature regarding the PID 2DOF controller and state feedback with reference tracking with integrators. These strategies were implemented in Arduino, where positive results were achieved in the control of the selected process due to the limitations of this hardware, given the need for a high sampling rate for better controller performance.

Keywords: Temperature control, PID 2DOF, State Feedback and Arduino.

Lista de ilustrações

Figura 1 – Rastreamento com Integradores.	19
Figura 2 – Resposta do sistema em Malha Aberta.	21
Figura 3 – Pólo do sistema em MA e restrições do projeto.	21
Figura 4 – Pólos do sistema em MF.	22
Figura 5 – Aplicação do PID projetado para o sistema com LGR.	23
Figura 6 – Comparativo entre as ações de controle.	24
Figura 7 – Ação de controle dos controladores projetados.	24
Figura 8 – Sistema em MF com os controladores.	25
Figura 9 – Ação de controle dos controladores projetados.	26
Figura 10 – Esquema de blocos para sistema discretizado.	27
Figura 11 – Transformação bilinear.	27
Figura 12 – Diagrama de Bode do processo.	28
Figura 13 – Região linear do processo.	29
Figura 14 – Plataforma Arduino.	30
Figura 15 – PIC 16F84	31
Figura 16 – Raspberry PI B	32
Figura 17 – Bancada didática	33
Figura 18 – Variáveis mais requisitadas na indústria.	34
Figura 19 – Circuito interno do LM35.	35
Figura 20 – Pinagem do LM35.	36
Figura 21 – Esquema de ligação com o microcontrolador para obter temperaturas no intervalo positivo do componente.	36
Figura 22 – Lâmpada DC utilizada para aquecimento da planta.	37
Figura 23 – Modelo de cooler utilizado para a dissipação de calor.	38
Figura 24 – Interface de acionamento e controle PWM para o cooler com alimentação de 12 Vdc.	40
Figura 25 – Interface de acionamento e controle PWM para lâmpada com alimenta- ção de 12 Vdc.	41
Figura 26 – Comparativo entre as respostas em MA e as FT estimadas.	48
Figura 27 – LGR com as restrições do projeto.	49
Figura 28 – Controle PID discreto.	50
Figura 29 – Comparativo entre PID 2DOF e PID discreto.	51
Figura 30 – Controle com Seguimento de referência com Integradores discretizado. .	52
Figura 31 – Resposta do sistema com controlador PID 2DOF atuando sobre o cooler	54
Figura 32 – Resposta do sistema com controlador PID atuando sobre o cooler . . .	55
Figura 33 – Comparativo entre PID e PID 2DOF	56

Figura 34 – Controle com seguimento de referência com integradores	57
--	----

Lista de tabelas

Tabela 1 – Comparativo de preços para aquisição de controladores PID Universais.	14
Tabela 2 – Comparativo das plataformas	32
Tabela 3 – Orçamento da bancada didática	34
Tabela 4 – Especificações da Lâmpada 64193NBU	38
Tabela 5 – Especificações do Cooler NONOIse A8025M12D	39
Tabela 6 – Desempenho do Hardware para os cálculos.	58
Tabela 7 – Ocupação dos algoritmos na memória do Hardware.	58
Tabela 8 – Desempenho do Hardware para os cálculos com 2 atuadores.	59
Tabela 9 – Ocupação dos algoritmos na memória do Hardware com 2 atuadores. .	59

Lista de abreviaturas e siglas

PID - Proporcional, Integral e Derivativo

2DOF - Dois graus de liberdade

GPIO - Entradas e Saídas

VCC/VSS - Alimentação Positiva

GND/VDD - Alimentação negativa

K_e - Ganho estático

K_i - Ganho dos integradores

DT - Duty cycle

PO - Ponto de operação

K_p - Ganho proporcional

K_i - Ganho da ação integral

K_d - Ganho da ação derivativa

SISO - Sistemas de uma entrada e uma saída

MIMO - Sistema de múltiplas entradas e múltiplas saídas

MA - Malha Aberta

MF - Malha Fechada

$t_5\%$ - Tempo de 5%

SS ou EE - State Space/Espaço de Estados

U_c - Sinal de controle do cooler

U_L - Sinal de controle da Lâmpada

Sumário

1	INTRODUÇÃO	13
1.1	Problemática	15
1.2	Metodologia	15
1.3	Objetivo	16
1.3.0.1	Objetivos Específicos	16
1.4	Organização	17
2	REVISÃO BIBLIOGRÁFICA	18
2.1	PID 2DOF	18
2.1.0.1	Controlador PID com divisão das ações de controle	18
2.2	Realimentação de Estados: Seguimento com Integradores	19
2.3	Simulações com as estratégias de controle	20
2.3.1	Projeto do controlador	20
2.3.2	Controlador PID	22
2.3.3	Controlador PID 2DOF	23
2.3.4	Controlador Seguimento com Integradores	25
2.4	Discretização	26
2.4.1	Transformação Bilinear	27
2.4.2	Tempo de amostragem	27
2.5	Wind - up	28
2.6	Hardwares de Baixo Custo	29
2.6.1	Arduino UNO REV 3	29
2.6.2	PIC 16F84	30
2.6.3	Raspberry PI 3	31
2.6.4	Comparativo entre as plataformas	32
3	INSTRUMENTAÇÃO DA BANCADA	33
3.1	Bancada	33
3.2	Sistema de medição	35
3.2.1	LM35	35
3.2.2	Esquema de ligação com o microcontrolador	36
3.3	Sistema de atuação	37
3.3.1	Lâmpada	37
3.3.2	Cooler	38
3.3.3	Drivers de acionamento	39
3.3.3.1	Cooler	39

3.3.3.2	Lâmpada	40
4	IMPLEMENTAÇÃO DOS ALGORITMOS	42
4.1	Período de amostragem	42
4.2	PID 2DOF	42
4.3	Seguimento de Referência por Integradores	44
4.4	Comunicação Serial com Matlab	46
5	PROJETOS DOS CONTROLADORES	47
5.1	Malha Aberta	47
5.1.1	LGR do Sistema de resfriamento	48
5.1.2	Discretização do Controlador	49
5.1.3	Simulação do controlador	50
5.2	Projeto em EE	51
5.3	Projeto para o sistema de aquecimento	52
6	RESULTADOS E DISCUSSÃO	53
6.1	PID 2DOF	53
6.2	PID	54
6.3	Comparativo entre PID e PID 2DOF	55
6.4	Seguimento de referência com Integradores	56
6.5	Performance do Hardware	57
6.5.1	Avaliando o hardware para um sistema SISO	57
6.5.2	Avaliando o hardware para sistemas MIMO	58
6.5.3	Limitações do Hardware	59
7	CONCLUSÕES E TRABALHOS FUTUROS	61
	REFERÊNCIAS	63

1 Introdução

O ser humano, desde o princípio da história, sempre tentou descrever os fenômenos que ocorriam ao seu redor. Uma das maneiras que o homem encontrou para descrever esses fenômenos foi a modelagem matemática, em que são criadas equações que descrevem o processo em forma de modelo [1].

Modelos matemáticos são criados para descrever o comportamento real de um determinado processo, desde o movimento de uma bola ao ser chutada, até o comportamento de reações químicas em uma caldeira. Com essa descrição pode-se realizar um projeto de sistemas de controle automático, todavia, forçando o sistema a ter um comportamento desejado pelo projetista. O controle automático tem papel fundamental no avanço da engenharia e da ciência, ele está presente nos diversos campos e vai desde o controle de veículos espaciais, direcionamento de mísseis a máquinas industriais utilizadas diariamente, como as de controle de pressão, temperatura, umidade, etc [2]

Com o passar dos anos, surgiu-se a necessidade de controlar sistemas mais complexos, com mais de uma variável, sendo assim, para modelar, isto é, definir uma função transferência para esses sistemas de controle moderno poderiam ser inter-relacionadas de uma maneira complexa. Contudo, o método de modelagem de sistemas em espaço de estados para análise e síntese de sistemas de controle é mais adequado para esses tipos de sistemas.

Atualmente na indústria, encontra-se a estrutura do PID e a técnica de realimentação de estados em CLP (Controladores lógicos programáveis) e SDCC's (Sistemas digitais de controle distribuído) respectivamente. Os CLP's são controladores microprocessados especiais que tem funções programadas pelo usuário, que substituem relés e dispositivos de passo eletromecânicos. De modo geral, o CLP implementa e controla uma lógica digital/analógica de processos sequenciais e malhas de controle analógicos para processos contínuos como o PID, como é encontrado nos dispositivos fabricados pela Siemens, Rockwell, Schneider, ABB entre outros. Já os SDCC's são sistemas distribuídos, onde seu objetivo foi substituir os sistemas de controle centralizados. Contudo, sua principal finalidade é descarregar o computador central, distribuindo as funções de controle básica, que necessitam de processamento rápido e repetitivo, desta forma, estes equipamentos tornam-se mais confiáveis, e detêm uma maior capacidade de atuação corretiva no caso de falha e execução de malhas de controle de modo mais rápido, entretanto, para a instalação destes equipamentos, é necessário um grande investimento [3]- [4]. Na Tabela 1 abaixo, é possível identificar o investimento necessário para a aquisição de controladores PID universais com modelos variados, onde os preços foram encontrados em sites de venda,

bem como: Mercado Livre, Submarino e etc.

Tabela 1 – Comparativo de preços para aquisição de controladores PID Universais.

PID	Fabricante	Modelo	Valor (R\$)
1	MYPIN	T4AIRR	480,00
2	Novus	N1100	520,00
3	Novus	N480d	614,80
4	Novus	N2000	780,00
5	Novus	N1200	790,00
6	ABB	CM10	2600,00

Decorrente da evolução tecnológica em cima dos microprocessadores, surgiram os microcontroladores, que diferem dos processadores por possuírem os periféricos embutidos em um mesmo corpo, como memórias, conversor analógico digital (A/D) e demais integrados. Derivado da sua versatilidade, estes equipamentos são introduzidos na indústria, em eletrodomésticos entre outros. Com a evolução tecnológica, e a queda de preços de componentes eletrônicos, os microcontroladores tornaram-se populares pelos baixo investimento necessário para a aquisição de um exemplar [5]- [6].

Para o controle de sistemas SISO, utiliza-se de métodos clássicos como o PID 2DOF, onde este, segundo [7] possui melhor performance quando comparado ao PID clássico, pois, consegue controlar o mesmo sistema com um menor esforço do controlador, sendo assim, proporcionando um melhor desempenho.

Nos últimos anos, houve um aumento no interesse por técnicas de controle multivariável de processos [8]. Um exemplo desse tipo de processo é o controle de variáveis em tanques. O controle de variáveis de nível, temperatura e vazão de tanques é algo muito interessante para fins de estudo, uma vez que é bastante comum em escala industrial, como nos ramos químico, petroquímico, de celulose e de alimentos [9].

Em [1] é possível verificar o uso da técnica de realimentação de estados para o controle de velocidade de um motor de corrente contínua, entretanto, desenvolveu-se um modelo para aproximar os resultados o mais próximo do real possível. Já em [10] é possível verificar o uso das técnicas empregadas para realizar a realimentação de estados para o controle de nível e temperatura de um reservatório, todavia, esta tese, demonstra as técnicas de representação em espaço de estados para o sistema completo além de demonstrar o passo a passo da linearização do sistema.

Baseando-se nestes trabalhos, neste documento propõem-se avaliar a implementação de algoritmos de controle, tais como: PID 2DOF com filtro derivativo e realimentação de estados com seguimento de referência com integradores em hardwares de baixo custo, avaliando a viabilidade desta execução bem como a performance do hardware frente os algoritmos de controle.

1.1 Problemática

Ao longo dos últimos anos, viu-se uma popularização dos microcontroladores e do open hardware, sendo estes eventos, interligados com a queda dos custos dos componentes eletrônicos, possibilitando a construção de pequenas plataformas com cotações mais acessíveis. No entanto, com estes fenômenos, notou-se uma grande ocupação dos microcontroladores no ramo tecnológico, onde estão cada vez mais presentes no cotidiano das pessoas, agregando a este, conforto e facilidades. Todavia, os dispositivos têm representado grandes impactos nos sistemas produtivos e no que tange inovações tecnológicas [6]- [11].

A automação tornou-se cada vez maior com o fenômeno da indústria 4.0, onde há interconexões entre todos os equipamentos, permitindo o crescente aumento da qualidade e produtividade, entretanto, surge paralelamente, maiores exigências dentro do processo para manter um padrão, fazendo-se necessário o uso de controladores para cumprir com tal [12], [13].

Tendo em vista a necessidade de reduzir custos para estes processos, faz-se necessário um estudo que possibilite avaliar a viabilidade destes algoritmos quando implementados em hardwares de baixo custo, verificando a capacidade de controle destes e a dificuldade de implementação além do desempenho do hardware frente a estas técnicas de controle.

1.2 Metodologia

Nos primeiros ensaios na planta de temperatura, aplicou-se degraus em malha aberta (MA), onde estimou-se as funções transferências (FT) para os dois atuadores (cooler e lâmpada), e a partir destas FT's, pôde-se descrever o sistema em EE, tanto contínuo quanto discreto. Esses modelos, foram utilizados para projetar os controladores tentando obter uma melhor resposta em malha fechada (MF). Com os controladores implementados, os comparou individualmente, coletando dados para inferir sobre a viabilidade da implementação, analisando parâmetros como: Performance e dificuldade de implementação.

1.3 Objetivo

Implementar diferentes algoritmos de controle em hardware de baixo custo, avaliando sua viabilidade e dificuldade de implementação em bancada didática.

1.3.0.1 Objetivos Específicos

- Estudar os algoritmos de controle PID 2DOF e seguimento de referência por integradores além de suas características;
- Avaliar os diferentes hardwares de baixo custo disponíveis: Arduino, Raspberry e PIC.
- Selecionar o hardware específico e fazer conexão com a planta didática;
- Implementar os algoritmos de controle;
- Comparar as características básicas;
- Analisar a viabilidade de cada técnica.

1.4 Organização

O presente trabalho está organizado da seguinte forma:

Capítulo 2: Revisão bibliográfica.

Capítulo 3: Construção da bancada didática.

Capítulo 4: Implementação dos algoritmos e testes de bancada.

Capítulo 5: Projeto dos controladores.

Capítulo 6: Resultados e Discussão.

Capítulo 7: Conclusões.

2 Revisão Bibliográfica

O algoritmo de controle PID é o mais encontrado na indústria, isto é, presente em mais de 95% das malhas de controle industriais segundo [14], entretanto, estudos delatam limitações destes controladores, quando aplicados em malhas MIMO, portanto, toma-se como alternativa a técnica de espaço de estados para representação do sistema, e realimenta-se estes estados para o controle das malhas em questão [15]. Nesta seção, destacam-se dois controladores clássicos que serão utilizados no decorrer da dissertação, onde são apresentados os conceitos destes e exemplos de aplicações para maior detalhamento da técnica de ambos.

2.1 PID 2DOF

Controladores com dois graus de liberdades (2DOF) incluem pesos nos termos proporcional e derivativo. Um controlador PID 2DOF é capaz de rejeitar perturbações com reduções significativas no overshoot inerente da atividade [16].

2.1.0.1 Controlador PID com divisão das ações de controle

Uma alternativa para representar um controlador com dois graus de liberdade é o PID com ponderação na referência. Este pode ser representado no domínio contínuo com a Equação 2.1:

$$u(t) = K_p((br(t) - y(t)) + K_i \int_0^t (r(t) - y(t))dt + K_d \frac{d}{dt}(cr(t) - y(t)) \quad (2.1)$$

A equação 2.1 pode ser dividida em três etapas, sendo a proporcional, onde multiplica-se o erro entre o estado e a referência $(r(t) - y(t))$ por um ganho K_p , tem-se também na etapa intermediária a ação integral, sendo esta, a integral do erro computado na atividade proporcional, resultando em um desvio nulo em relação a sinais do tipo degrau, e por fim observa-se a atuação derivativa, onde tem-se uma predição do estado futuro por meio da derivada do erro. No PID 2DOF, nota-se a adição de dois parâmetros, dos quais, modificam a equação responsável pelo controle, sendo estes: b na ação proporcional e o c na derivativa.

Segundo [14], os pesos b e c, assumem valores de 0 até 1, tornando-se fatores de ponderação da ação proporcional e derivativa do controlador, entretanto, se ambos assumirem o valor 1, tem-se o controlador PID tradicional, nada obstante, assumindo o valor 0 as ações P e D são efetivas somente na realimentação.

Para comprovar as afirmações de [14] e [17], na próxima seção será simulado a Equação 2.1 utilizando-se do Matlab e Simulink, comprovando assim, que a atuação do controlador PID 2DOF realiza menos esforços para levar o sistema até a referência.

2.2 Realimentação de Estados: Seguimento com Integradores

Realimentação de estados (seguimento com integradores) segundo [17], é um alternativa de controle, do qual garante erro nulo no regime permanente para as entradas do tipo degrau. Esta técnica de controle consiste na utilização de integradores como é ilustrado na Figura 1.

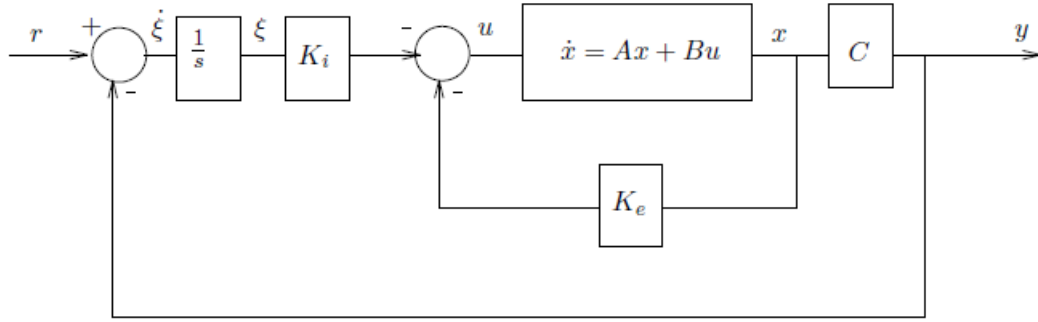


Figura 1 – Rastreamento com Integradores.

Nada obstante, pode-se representar o diagrama da Figura 1 de forma matricial (Equação 2.2), todavia, nota-se um aumento das matrizes em relação a realimentação de estados comum, entretanto, este efeito é justificado pela adição dos integradores ao diagrama, sendo assim, há mais estados, dos quais, precisam ser representados matematicamente.

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix}}_{\dot{x}_a} = \underbrace{\begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}}_{A_a} \underbrace{\begin{bmatrix} x \\ \xi \end{bmatrix}}_{x_a} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_a} u + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_{E_a} r, \quad u = - \underbrace{\begin{bmatrix} K_e & K_i \end{bmatrix}}_{K_a} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix}}_{\dot{x}_a} \quad (2.2)$$

Contudo, há uma maneira simplificada de representar o diagrama da Figura 1, utilizando as equações de estado 2.3 e 2.4, onde \dot{x} é o estado do sistema, ξ representa o estado dos integradores e K_i e K_e são os ganhos dos integradores e da realimentação de estados respectivamente, sendo estes calculados para que em regime permanente, o sistema apresente erro nulo da referência r para a saída y .

$$\dot{x} = Ax(t) + Bu(t), \quad u(t) = -K_e x(t) - K_i \xi(t) \quad (2.3)$$

$$\dot{\xi} = r(t) - y(t) = r(t) - Cx(t) \quad (2.4)$$

Em suma, segundo [2], o sistema em regime permanente, é descrito com a Equação 2.5 de controle $u(t)$ apresentada abaixo. Quando a Equação 2.5 é analisada, nota-se que em regime permanente, o valor de $\dot{\xi} = 0$, resultando em $\bar{r} = \bar{y}$, isto é, esse resultado indica que o erro de rastreamento é nulo em regime permanente.

$$\bar{u} = -K_a \bar{x}_a = Ka(A_a - B_a K_a)^{-1} E_a \bar{r} \quad (2.5)$$

2.3 Simulações com as estratégias de controle

Para demonstrar o funcionamento dos controladores, fez-se um projeto de controle com lugar das raízes, onde foi possível determinar os parâmetros dos controladores e encontrar os polos desejados em malha fechada, isto é, foi possível determinar o comportamento desejado do sistema quando a malha é fechada.

2.3.1 Projeto do controlador

Para esta simulação, optou-se por realizar o projeto em cima da Equação 2.6, onde em malha aberta nota-se um settling time de 1210 s, como é mostrado na Figura 2. Visto isso, definiu-se como objetivos de controle, que o tempo em MF fosse 10x menor que o tempo em MA, além de um sobressinal de até 15% e erro nulo a entradas do tipo degrau.

$$G(s) = \frac{4.07}{308.7s + 1} \quad (2.6)$$

Para o processo em questão, nota-se que o tempo de resposta deste é muito lento, implicando que seu único polo aproxime-se muito de zero, como pode ser visualizado na Figura 3. Nesta figura também pode-se visualizar as linhas que representam os objetivos de controle, ou seja, o settling time e o overshoot (sobressinal) desejado, desta forma, deseja-se que os pólos em malha fechada fiquem dentro das limitações, isto é, em MF os pólos devem ser alocados a esquerda da linha vertical da limitação de tempo desejado além de estarem dentro das linhas diagonais de overshoot máximo permitido, seguindo os objetivos de projeto.

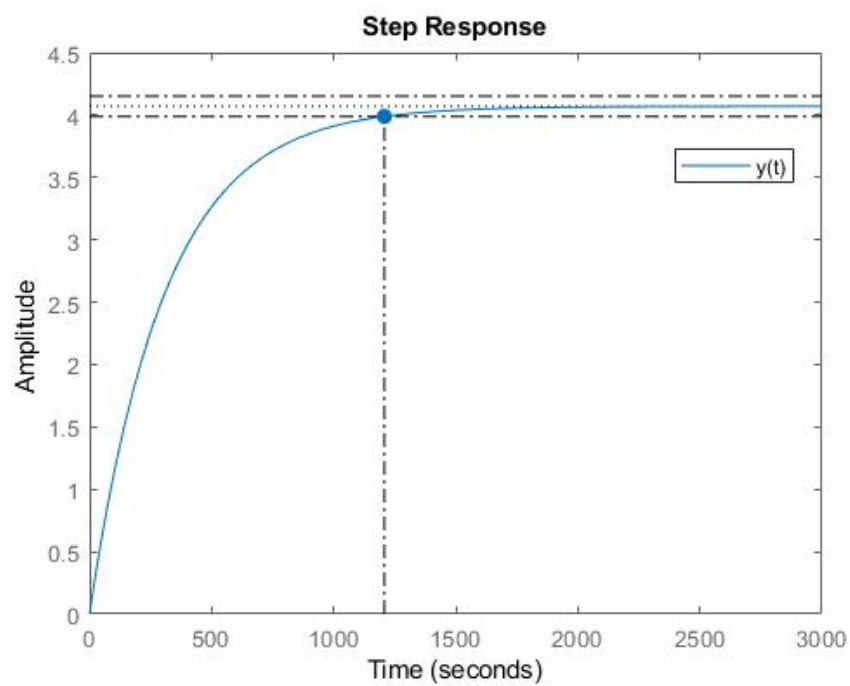


Figura 2 – Resposta do sistema em Malha Aberta.

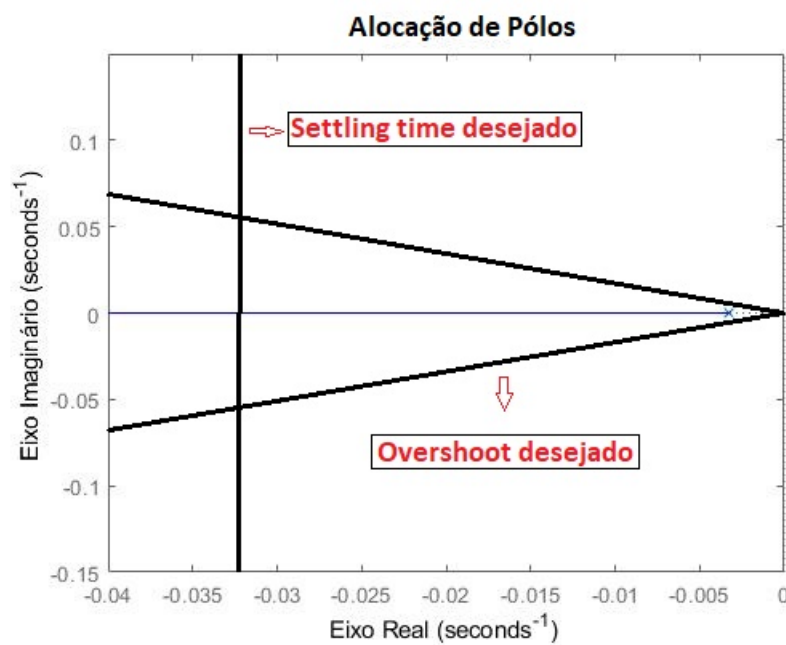


Figura 3 – Pólo do sistema em MA e restrições do projeto.

2.3.2 Controlador PID

A partir dos dados que podem ser visualizados nas Figuras 2 e 3, pode-se resumir os objetivos de projeto nos tópicos abaixo:

- Tempo de 5% desejado em MF: $\frac{t_{Ma}}{10} = 121 \text{ s}$
- Sobressinal máximo permitido: 15%
- Erro nulo ao degrau (Ação integral)

Com estas restrições de projeto e sabendo que a equação do controlador (Equação 2.7), pode-se projetar o controle por meio de lugar das raízes (LGR), de tal forma que os objetivos sejam alcançados.

$$C(s) = K_p + \frac{K_i}{s} + K_d \cdot s \quad (2.7)$$

Com a Equação 2.7, sabe-se que será necessário adicionar dois zeros e um pólo ao sistema, para que este, em MF, desenvolva o comportamento desejado, sendo assim, uma possível solução utilizando o LGR é o controlador representado na Equação 2.8 abaixo:

$$C(s) = 7.03 + \frac{0.182}{s} + 25.1 \cdot s = \frac{25.1 \cdot s^2 + 7.03 \cdot s + 0.182}{s} \quad (2.8)$$

Com este controlador, observa-se os seguintes pólos complexos em malha fechada: $-0.0360 + 0.0225i$ e $-0.0360 - 0.0225i$, onde estes são representados na Figura 4 para facilitar a visualização do sistema como um todo.

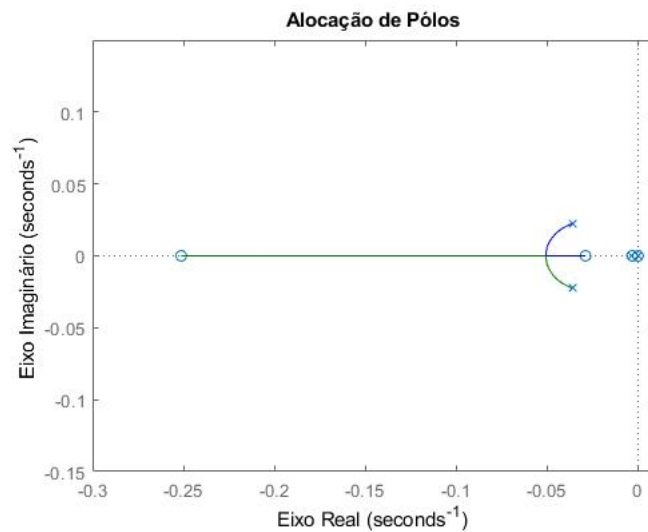


Figura 4 – Pólos do sistema em MF.

Com a equação do controlador 2.8 e a equação do sistema 2.6 em mãos, pode-se simular o processo em malha fechada com o auxílio do Simulink, desta forma, aplicou-se além de um degrau na entrada, uma perturbação em um determinado instante de tempo t para ver como funciona a rejeição do sistema. O resultado pode ser analisado na Figura 5.

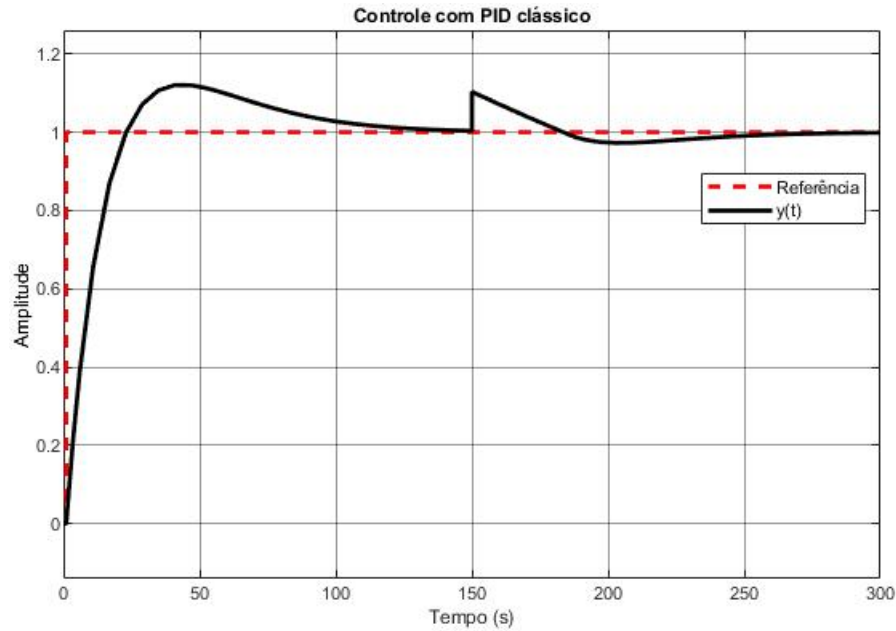


Figura 5 – Aplicação do PID projetado para o sistema com LGR.

2.3.3 Controlador PID 2DOF

Utilizando o mesmo projeto de controle realizado para o PID, apenas adicionou-se os parâmetros b e c a Equação 2.8, sendo assim, simulou-se o sistema com os mesmos parâmetros encontrados, entretanto, acrescentou-se três variações de PID 2DOF, sendo a primeira variação com $b = c = 0.2$, a segunda $b = c = 0.5$ e a última $b = c = 0.9$. Essas variações são importantes para demonstrar como que a mudança dos parâmetros b e c tiram a agressividade do controlador, desta forma atenuando os sobressinais como relatado por [14], todavia, para rejeição de perturbações, o comportamento permanece o mesmo.

Nas Figuras 6 e 7, pode-se analisar a ação das estratégias de controle, onde comprovam as afirmações de [14].

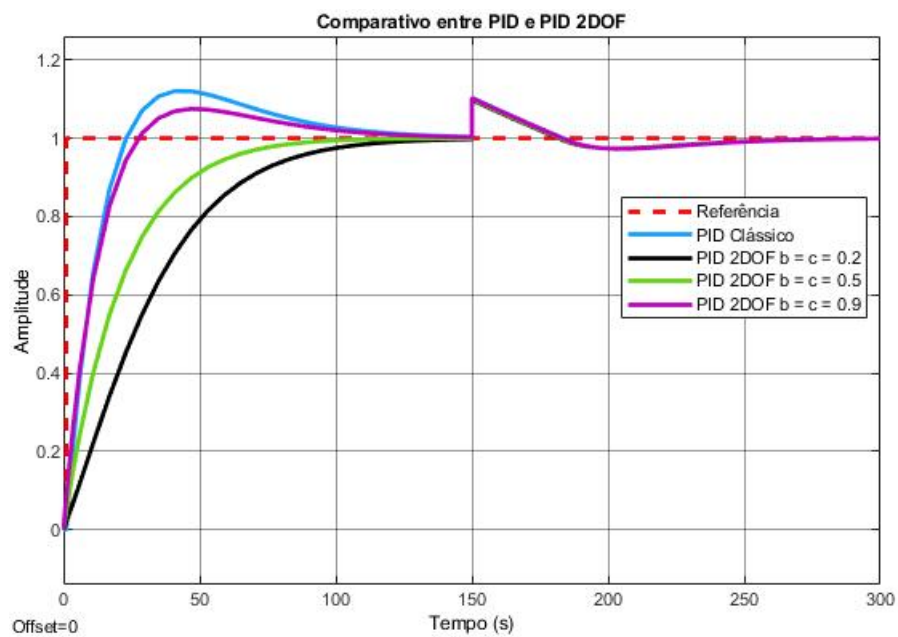


Figura 6 – Comparativo entre as ações de controle.

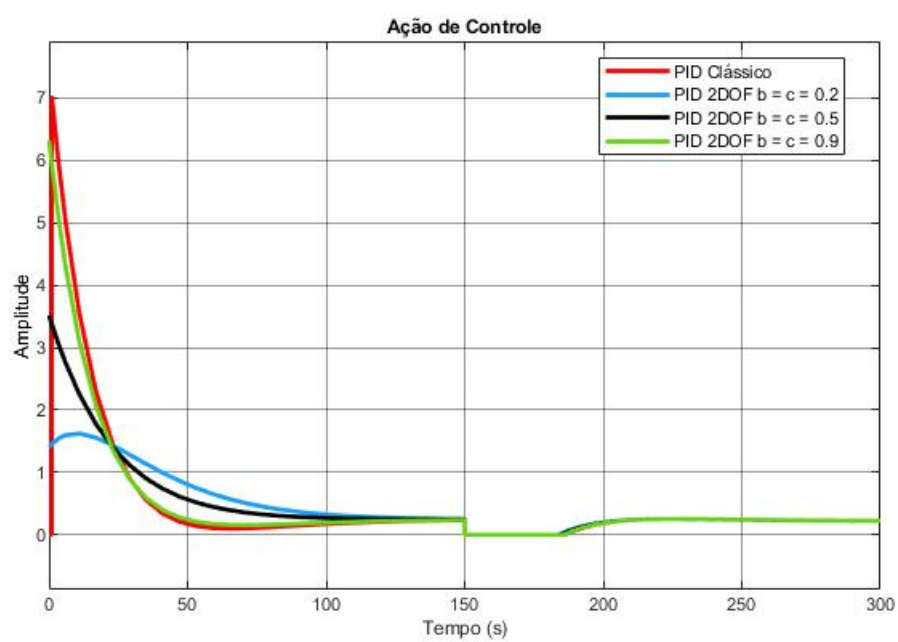


Figura 7 – Ação de controle dos controladores projetados.

2.3.4 Controlador Seguimento com Integradores

Para esta estratégia, adotou-se os mesmos objetivos fixados nas estratégias anteriores, todavia, coletou-se os polos em malha fechada do PID clássico e inseriu-se como polos desejados desta estratégia. Em suma, assim como na estratégia anterior, estabeleceu-se como referência um degrau unitário, onde a simulação pode ser dividida em dois períodos, o primeiro para alcançar a referência e o segundo período para rejeitar a perturbação inserida no sistema.

Para a análise comparativa das estratégias, selecionou-se apenas o PID 2DOF com parâmetros intermediários, mantendo apenas 3 curvas na imagem, além da referência. Nas Figuras 8 e 9, pode-se visualizar os comportamentos dos sistemas com os controladores e as ações de controle em MF.

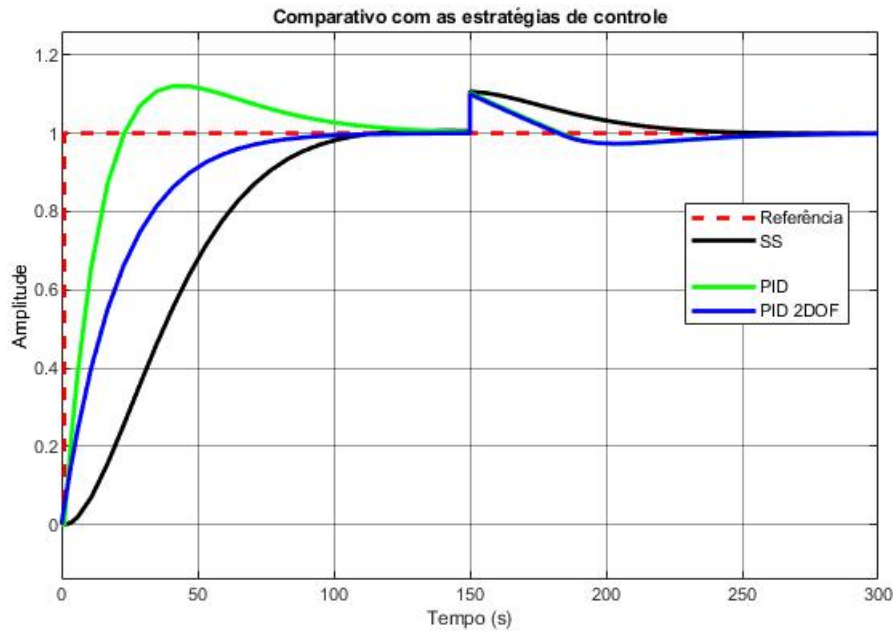


Figura 8 – Sistema em MF com os controladores.

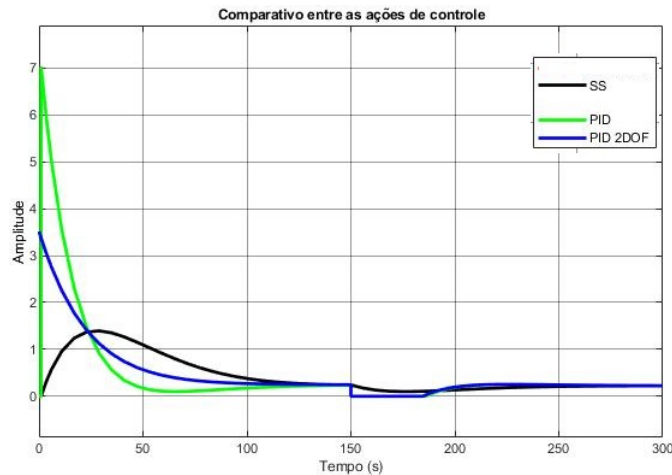


Figura 9 – Ação de controle dos controladores projetados.

Em suma, com as Figuras 8 e 9, nota-se um comportamento semelhante com as diferentes técnicas de controle, todavia, para um sistema SISO, toma-se como mais viável o controlador PID, onde há apenas três parâmetros de ajuste e um projeto mais simplificado, e se necessário utiliza-se o PID com dois graus de liberdade para tirar a agressividade do controlador, adicionando-se dois novos parâmetros. No entanto, se o sistema passar de SISO para MIMO, a estratégia com espaço de estados torna-se mais viável, onde a mesma sequência de projeto, pode ser utilizada para n variáveis, ao contrário do PID, que a cada variável que deseja-se controlar, é necessário a adição de uma nova malha de controle, acrescentando sempre mais três parâmetros.

2.4 Discretização

Microcomputadores e microprocessadores digitais são largamente utilizados na indústria atual, para fins de controle dos processos. No entanto, um grande número de sistemas industriais são de natureza analógica. Sempre que um microcomputador faz parte de um sistema analógico a presença de conversores A/D (analógico para digital) e D/A (digital para analógico) se faz necessária [18]. A representação do sistema com a discretização, pode ser observada na Figura ?? abaixo:

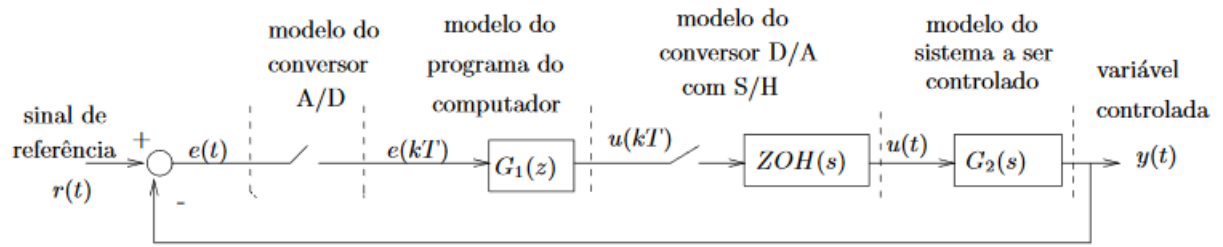


Figura 10 – Esquema de blocos para sistema discretizado.

2.4.1 Transformação Bilinear

Há diversos métodos para discretizar sistemas contínuos, entretanto, neste documento, optou-se pela transformação bilinear (transformação Tustin). A Transformação Bilinear é um tipo de mapeamento que mapeia pontos de um plano no outro. Os pontos sobre o eixo imaginário do Plano S são mapeados sobre círculo de raio unitário do Plano Z, a região à esquerda do eixo imaginário no Plano S é mapeada dentro do círculo de raio unitário no Plano Z [19], como é mostrado na Figura 11 abaixo:

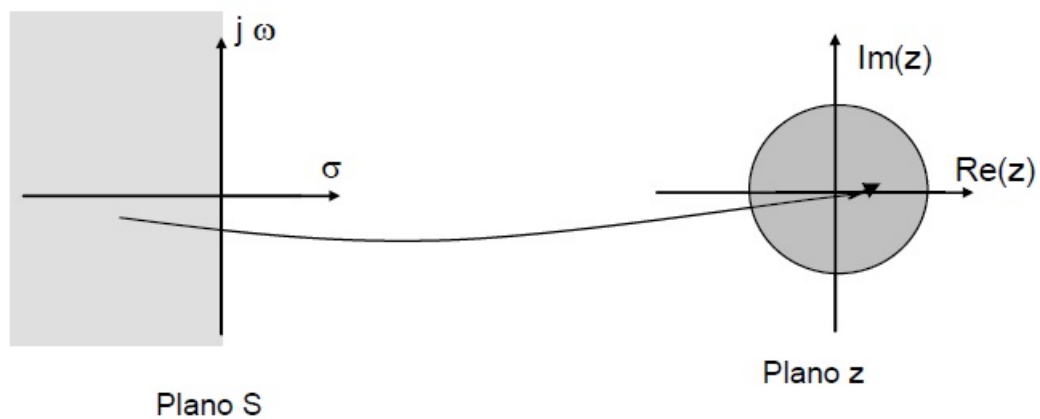


Figura 11 – Transformação bilinear.

2.4.2 Tempo de amostragem

A melhor escolha do período de amostragem em sistemas de controle é um compromisso entre vários fatores normalmente contraditórios. Normalmente a performance de um controlador digital melhora com o aumento da frequência de amostragem mas o custo do dispositivo é proporcional. A diminuição da frequência de amostragem significa mais tempo disponível para o cálculo do sinal de controle em tempo real, o que possibilita a utilização de computadores mais lentos e portanto mais baratos [18] - [20].

Vários fatores afetam a performance de controladores digitais e para que o sistema apresente uma performance mínima aceitável se faz necessário uma frequência de amostragem mínima tal que todos os sinais presentes no sistema de controle, e em especial as variáveis controladas, respeitem o Teorema da Amostragem no pior caso de excitação do sistema [18] - [20].

Tipicamente, a frequência de amostragem é escolhida de 15 à 20 vezes maior que a frequência de banda passante (ω_b), isto é, $15 \cdot \omega_b \leq \omega_a \leq 20 \cdot \omega_b$ [18] - [20]. Desta forma para o sistema apresentado na Equação 2.6, pelo seu diagrama de Bode apresentado na Figura 12, pode-se definir a largura em frequência da banda passante, desta forma descobrindo um bom período de amostragem para o sistema.

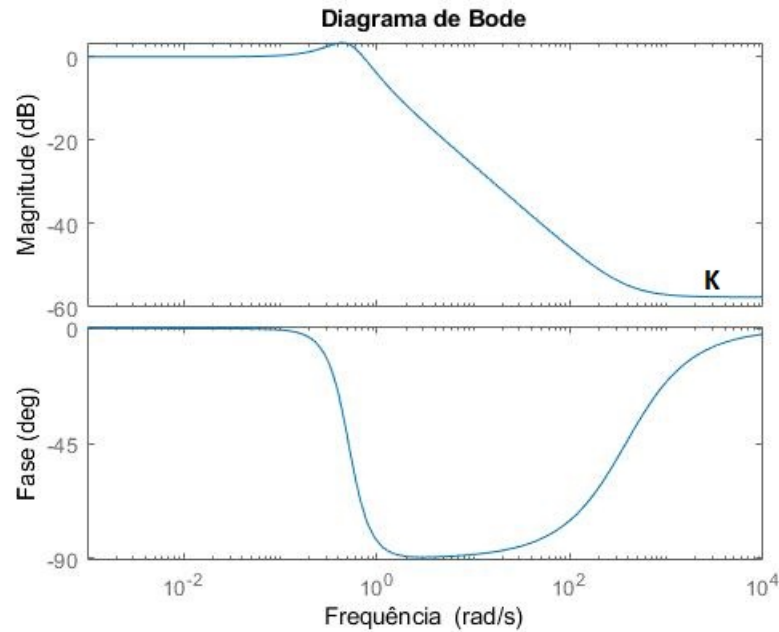


Figura 12 – Diagrama de Bode do processo.

Segundo [21], pode-se definir a frequência ω_b por $\omega_b = 0.707 \cdot K$, onde K é o valor máximo assumido no diagrama de magnitude, portanto, para o sistema em questão, tem-se como $\omega_b = 0.707 \cdot 57.6 = 40.73 \text{ rad/s}$.

2.5 Wind - up

Na prática, todos os atuadores apresentam limites. Quando o valor da variável de controle atinge o limite máximo ou mínimo do atuador ocorre a saturação do sinal de controle. Este fato faz com que a malha de realimentação seja de certa forma quebrada, pois o atuador permanecerá no seu limite independentemente da saída do processo. Contudo, se um controlador com ação integral é utilizado, o erro continuará a ser integrado e o termo integral tende a se tornar muito grande (wind-up). Neste caso, para que o controlador volte

a trabalhar na região linear, isto é, saia da saturação, é necessário que o termo integral seja decrementado [22].

Para decrementar o termo integral, pode-se tomar como exemplo as implementações práticas de [23], onde é fornecido um algoritmo em C++ para Arduino como uma possível solução para este inconveniente, sendo assim, o acúmulo da integral fica restrito e os atuadores não são saturados, portanto, o sistema permanecerá na região linear, como é mostrado na Figura 13 abaixo:

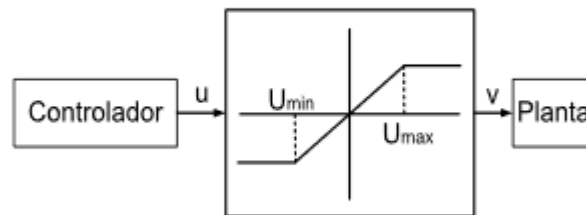


Figura 13 – Região linear do processo.

2.6 Hardwares de Baixo Custo

Um microcontrolador é um sistema microprocessado com diversos periféricos (funcionalidades) disponíveis em um único chip, ou seja, é um bloco com memórias de programa, memória de dados, RAM, temporizadores e circuitos de clock. Ao agrupar todas essas funcionalidades, cria-se a possibilidade de desenvolvimento rápido de sistemas eletrônicos com emprego de um pequeno número de componentes [24].

No mercado, encontra-se diversas opções de microcontroladores, tais como: Arduinos, Raspberry PI, PIC's, Node MCU, Teensy 3, MSP 430, entre outros, todavia, no recorte abaixo, será apresentado três microcontroladores mais populares no cenário nacional, onde encontra-se um vasto suporte para problemas que possam surgir nas implementações.

2.6.1 Arduino UNO REV 3

Arduino é uma plataforma eletrônica open-source baseada em um hardware e software de fácil utilização. Esta placa permite leitura de entradas (sensores, botões, entre outros), além de permitir o controle de suas saídas para acionamento de atuadores. Com o passar dos anos, realizaram-se diversos projetos utilizando a plataforma, tornando a solução de problemas mais simples perante a difusão da placa em diversos ramos da ciência [25].

O cérebro desta placa é o microcontrolador ATMEGA 328P da família AVR. O ATMEGA 328P é um microcontrolador compacto que apresenta a maioria das características da família AVR, entretanto, apresenta uma memória maior que os demais, contendo uma pinagem idêntica [24].

O Arduino Uno Rev 3 apresenta um conjunto de 13 slots de entradas/saídas digitais, e um conjunto de 6 entradas de leitura analógicas, entretanto, em relação ao slot digital, 6 destas podem ser utilizadas como saídas para controle PWM de alguma carga como é demonstrado na Figura 14



Figura 14 – Plataforma Arduino.

2.6.2 PIC 16F84

O PIC 16F84 pertence a uma classe de microcontroladores de 8 bits, com uma arquitetura RISC (Computador com um conjunto reduzido de instruções). O PIC16F84, é adequado para diversas aplicações, como a indústria automível, sensores remotos, fechaduras eléctricas e dispositivos de segurança. É também um dispositivo ideal para cartões inteligentes, bem como para dispositivos alimentados por baterias, por causa do seu baixo consumo. O baixo custo, consumo e a facilidade de manuseamento e flexibilidade, fazem com que o PIC16F84 seja bem requisitado para soluções de problemas eletrônicos [26]. Na Figura 15 é possível visualizar o microcontrolador em questão.



Figura 15 – PIC 16F84 .

2.6.3 Raspberry PI 3

A Raspberry PI 3 é uma plataforma que conta com o microprocessador BCM 2837 (Módulo de Controle da Carroceria), possui 27 GPIO's e como diferencial das demais plataformas, está suporta sistemas operacionais, como:

- Ubuntu
- Windows 10 IoT
- Raspbian
- OSMC
- Arch Linux ARM
- SUSE 64 bits

Por ser uma plataforma mais robusta, e com mais funções que as demais, apresenta-se um preço mais elevado, sendo este, compatível com as funcionalidades oferecidas, todavia, permanece na faixa de hardwares de baixo custo. Outro ponto diferencial do Raspberry é que pode-se programá-lo em C e Python, além de oferecer conexões via protocolo Ethernet sem precisar de módulos para tal, como é perceptível as conexões dispostas na placa (Figura 16) [27].



Figura 16 – Raspberry PI B .

2.6.4 Comparativo entre as plataformas

A partir da análise destas plataformas, apurou-se os dados e os agrupou em uma tabela, sendo esta, de fim comparativo, influenciando na escolha do hardware para a execução do projeto.

Tabela 2 – Comparativo das plataformas

Plataforma	Clock	GPIO	Alimentação	SRAM	Preço
Arduino	16 MHz	14	7 - 12 V	2 KB	R\$ 25.75
Raspberry	1.4 GHz	40	5 V	1 GB	R\$ 170.00
Pic16f84 (Kit)	10 MHz	13	5 V	264 KB	R\$ 70.00

Com o intuito de desenvolver uma bancada de baixo custo, optou-se pela utilização da plataforma Arduino, sendo esta capaz de receber os algoritmos utilizados para controle, além de deter a melhor acessibilidade econômica. Esta plataforma também conta com uma grande comunidade que trabalha em conjunto, auxiliando no desenvolvimento de novas aplicações e na solução de problemas relacionados a implementações.

3 Instrumentação da bancada

O projeto deu-se entorno de uma bancada didática, onde adaptou-se uma estrutura com materiais de baixo custo, tanto medição de estado (temperatura) como para atuação no resfriamento da estufa por meio de dois atuadores (cooler e lâmpada). Em suma, com estes medidores e atuadores, torna-se possível a construção de uma futura bancada didática de baixo custo, aproveitando-se do modelo estrutural proposto por este trabalho.

3.1 Bancada

A bancada utilizada representa uma estufa, onde o controle da temperatura da-se por meio da dissipação de calor com um cooler e com a fonte de calor, isto é, uma lâmpada DC. Na Figura 17 abaixo, há a representação do sistema ao qual implantou-se os controladores, vale ressaltar que esta estrutura encontrava-se fechada durante os ensaios.

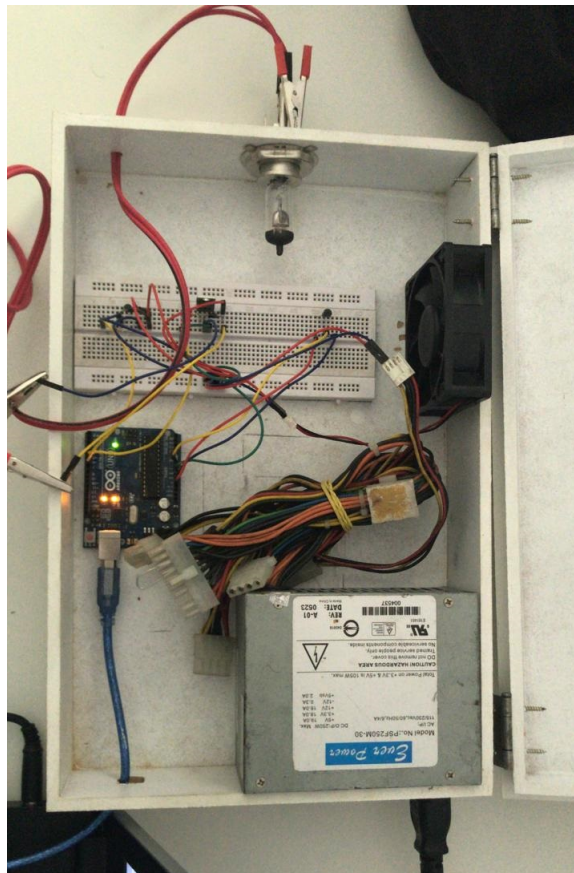


Figura 17 – Bancada didática

Nesta bancada, deseja-se controlar a temperatura, sendo esta, uma das mais presentes/requisitadas na indústria segundo [28], como é demonstrado pela Figura 18

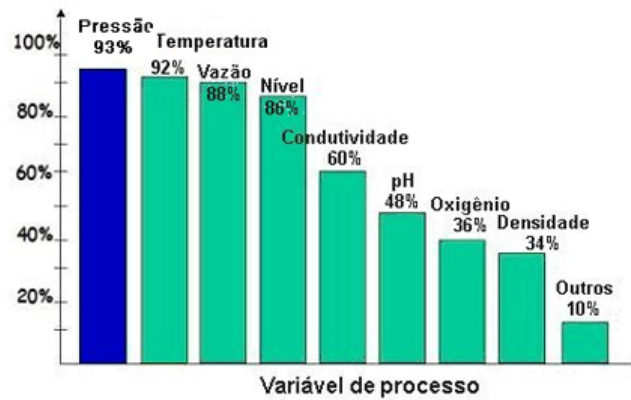


Figura 18 – Variáveis mais requisitadas na indústria.

Baseando-se nesta necessidade apresentada na Figura [28], propõem-se o estudo de controladores aplicados em hardware de baixo custo para a estabilidade desta variável.

Não só o hardware onde foram implementados as diferentes estratégias de controle são de baixo custo, mas, todo o sistema de atuação e medição enquadram-se neste modelo. Na Tabela 3, é apresentada os custos totais para a confecção da bancada.

Tabela 3 – Orçamento da bancada didática

Dispositivo	Preço (R\$)
LM35	0,50
Drives de Acionamento	10,00
Fonte de Alimentação	50,00
Cooler	10,00
Estrutura	20,00
Arduino UNO	25,00
Total	115,00

3.2 Sistema de medição

Para a medição da temperatura, utilizou o sensor LM35, sendo este descrito abaixo, onde é detalhado o princípio de funcionamento, a estrutura do sensor e o esquema de ligação deste.

3.2.1 LM35

O transdutor de temperatura LM35, projetado pela Texas Instruments, é preciso e possui uma resposta de tensão relativa à temperatura de forma linear que pode ser alimentado com até 40 V, apresentando uma variação de 10 mV por grau celsius de temperatura. Segundo seu fabricante, sua vantagem é a não necessidade de calibração externa para se obter exatidão, além disso, possui uma precisão de até 0,25 °C dentro da faixa de temperatura de -55 °C à 150 °C [29]. O circuito interno do transdutor apresentado na Figura 19 é encontrado no datasheet do componente, entretanto, na Figura 20 é possível identificar a pinagem do componente, onde +Vs é a entrada de tensão e V_{out} o sinal de temperatura.

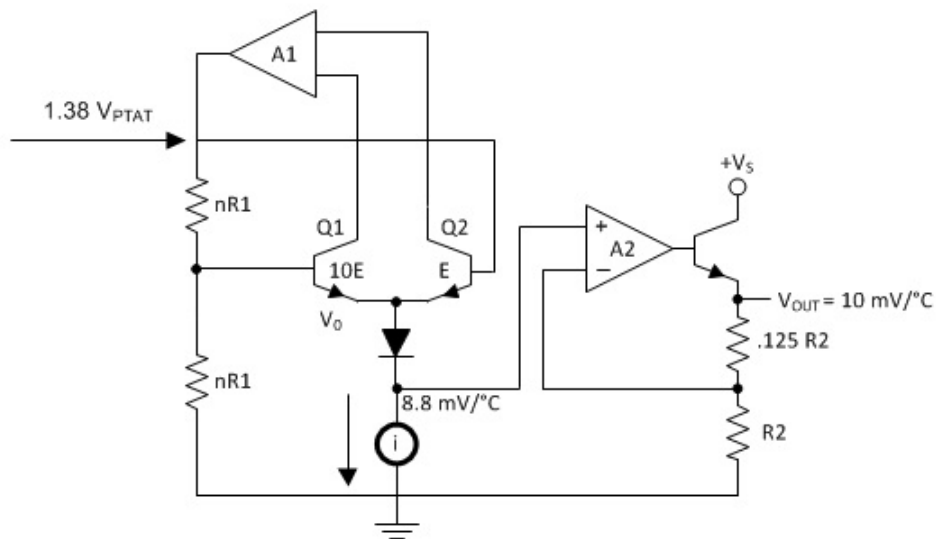


Figura 19 – Circuito interno do LM35.

3.3 Sistema de atuação

Para controlar a temperatura no sistema, optou-se por atuar de dois maneiras em cima do estado desejado, sendo esta atuação por vezes acontecendo por meio de uma fonte de calor (lâmpada DC) outrora por um dissipador de calor (cooler). A seguir serão apresentados as principais características dos atuadores e seus respectivos drives de acionamento.

3.3.1 Lâmpada

Para o sistema de aquecimento, fez-se uso de uma lâmpada utilizada em motocicletas. A escolha desta deu-se pela facilidade da construção de um driver de acionamento e pelo seu baixo custo. Este atuador contém duas opções: Baixa iluminação e alta iluminação, onde optou-se por utilizar a primeira opção por conta desta atribuir as condições necessárias exigidas pelo sistema além de resultar em um consumo de corrente menor que a outra opção, sendo assim, não necessitando de fontes de alimentações mais robustas. Na Figura 22 é possível identificar as características físicas dos atuador.



Figura 22 – Lâmpada DC utilizada para aquecimento da planta.

A intensidade da fonte de calor é determinada por meio dos três terminais do componente, sendo o terminal central o GND, o extremo esquerdo a opção de baixa intensidade luminosa e o extremo direito a opção de alta intensidade luminosa. Vale salientar que independente da intensidade, a tensão de alimentação permanece a mesma, sendo a corrente o fator flutuante, variando de 2 A até aproximadamente 4 A.

Abaixo na Tabela 4 encontra-se as especificações da lâmpada, sendo estas, retiradas do datasheet do atuador, onde foram úteis para o desenvolvimento do driver.

Tabela 4 – Especificações da Lâmpada 64193NBU

Características	Unidades
Potência	60 W
Tensão de Alimentação	12 Vdc
Corrente Nominal	2 A
Fluxo Luminoso	1600/1000 lm

3.3.2 Cooler

Para a dissipação da massa de calor presente na estufa, utilizou-se um cooler de atuação simples. Essa predileção deu-se por conta das duas dimensões encaixarem perfeitamente no invólucro, um baixo consumo de corrente e pelo seu baixo custo de aquisição. Na Figura 23 é possível identificar um componente do mesmo fabricante semelhante ao utilizado na planta.



Figura 23 – Modelo de cooler utilizado para a dissipação de calor.

Na Tabela 5 estão as especificações fornecidas pelo fabricante, sendo estas, fundamentais para a confecção da interface de acionamento do atuador.

Tabela 5 – Especificações do Cooler NONOIse A8025M12D

Características	Unidades
Tensão de Alimentação	12 Vdc
Corrente Nominal	0.130 A

3.3.3 Drivers de acionamento

Para os drivers de acionamento dos atuadores, baseou-se nos modelos apresentados por [30], onde o sistema de sua autoria possui características semelhantes a bancada didática desenvolvida neste trabalho, entretanto, algumas modificações foram realizadas decorrente da necessidade de cada atuador, afim de minimizar gastos e deixar o sistema eficiente.

3.3.3.1 Cooler

Para o acionamento do cooler, baseou-se no diagrama utilizado por [30], onde a amplificação do sinal de corrente é feita por um transistor BC 548, sendo este um transistor NPN que permite uma corrente de 500 mA em seu coletor, sendo esta corrente aproximadamente 3.85 vezes superior a corrente máxima de trabalho da ventilação, sendo assim, podemos constatar que a interface de acionamento e controle irá trabalhar sem superaquecimento em seus componentes eletrônicos.

Na Figura 24, é possível visualizar o esquema elétrico utilizado no driver para o controle PWM da carga.

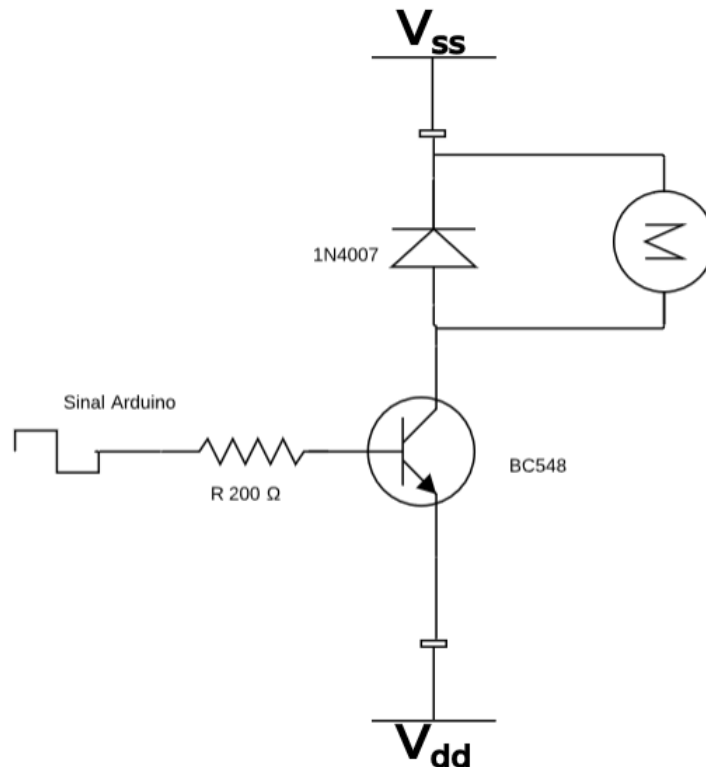


Figura 24 – Interface de acionamento e controle PWM para o cooler com alimentação de 12 Vdc.

3.3.3.2 Lâmpada

Assim como a interface de potência do cooler, o driver para a lâmpada foi baseado no modelo proposto por [30], entretanto, aplicou-se uma modificação no sistema original, buscando uma maior eficiência no sistema.

No driver proposto por [30], o MOSFET utilizado para realizar o controle da carga suporta uma corrente de dreno (I_d) de aproximadamente 110 A segundo o fabricante, sendo esta corrente 27.5 vezes maior que a corrente máxima da carga a ser controlada. Visando isto, na interface realizada neste trabalho, o IRF 3305 foi substituído por um IRF 540, onde a sua corrente de dreno gira em torno de 20 A até 28 A, 5 vezes maior que a corrente máxima da lâmpada, portanto, com esta expansão do limite de trabalho, não há sobreaquecimentos no sistemas mesmo sem utilizar dissipador no componente.

Na Figura 25, pode-se visualizar o esquema elétrico realizado para o acionamento e controle da carga, onde a alimentação do sistema, assim como no cooler é de 12 Vdc.

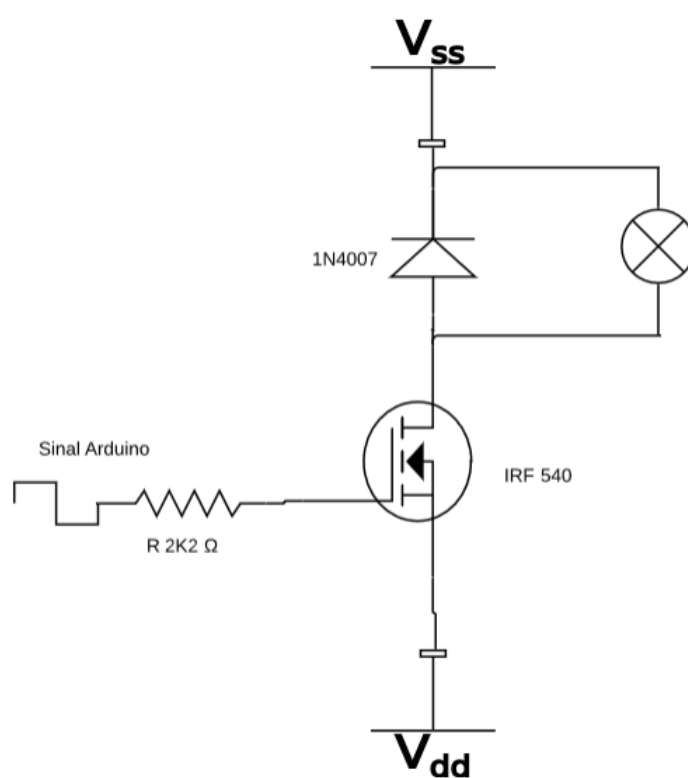


Figura 25 – Interface de acionamento e controle PWM para lâmpada com alimentação de 12 Vdc.

4 Implementação dos Algoritmos

As estratégias de controle foram implementadas na plataforma Arduino UNO, onde desenvolveu uma classe para auxiliar nas funções principais fornecidas pela IDE do hardware, sendo assim, a partir da chamada do construtor, antes do *void setup()*, bastava apenas chamar os métodos públicos da classe, sendo estes métodos necessários para inicializar um Setpoint, posteriormente adicionar uma entrada no sistema e por fim calcular a saída do controlador.

4.1 Período de amostragem

Para os algoritmos implementados, definiu-se como período de amostragem, um tempo de 25 *ms*, este tempo, foi obtido a partir de ensaios práticos com os algoritmos de controle desenvolvidos e a interface de visualização de dados, onde observou-se que o tempo mínimo para execução de um ciclo de cálculo é de aproximadamente 24.7 *ms*, assim, aproximando-se o valor, chega-se a 25 *ms*, onde, este foi utilizado para que o hardware trabalhe na sua capacidade máxima.

4.2 PID 2DOF

Para o desenvolvimento do algoritmo desta estratégia de controle, implementou-se a Equação 2.1, onde dividiu-se a atuação em métodos dentro de uma classe PID 2DOF. Vale ressaltar que para a aplicação prática, é necessário a adição de windups na ação integral, sendo esta estratégia utilizada para evitar a saturação do controlador, fazendo com que o sistema saia da região linear. Abaixo, é demonstrado o Pseudocódigo da implementação no Arduino, esta implementação 1, pode ser dividido em duas frentes, sendo estas: Inicialização das constantes do controlador e cálculo da ação de controle. A inicialização das constantes de entrada são realizadas no *void setup()*, outrora, os cálculos são realizados de forma cíclica no *void loop()*, desta forma a ação de controle é efetuada de forma cíclica a cada 25 *ms*.

Algoritmo 1: Classe PID 2DOF

Entrada: Ganhos: K_p , K_i , K_d , b , c , T_s **Saída:** Controle $u(t)$ **início**

Inicializa os ganhos do controlador

Define o setpoint desejado

Define o período de amostragem

if $T_s > (0.025)$ **then**

Realiza uma leitura do sensor

Filtra os dados coletados do sensor com janela móvel

Calcula o desvio entre a leitura e o Setpoint

Calcula o erro para a ação proporcional

$$ErroP = Referncia \cdot b - Temperatura$$

 Calcula a ação proporcional (P), multiplicando o ErroP pela constante K_p Calcula a diferença de tempo entre uma execução de máquina (dT) Calcula a ação integral aplicando um somatório: $I += Erro \cdot dT$ Calcula a ação diferencial (D): $\frac{UltimaAmostra - Amostra \cdot c}{dT}$

Atualiza a última amostra

 Calcula a ação de controle (u), somando as três ações calculadas previamente

$$u = P + I + D$$

Ação anti wind-up

Implementação de saturação virtual de acordo com as limitações do hardware

Escreve o valor de controle na saída PWM referente aos atuadores

else | Aguarda até o T_s ser maior que o período de amostragem **end****fim**

4.3 Seguimento de Referência por Integradores

Para está estratégia de controle, utilizou-se a biblioteca *BasicLinearAlgebra.h*, proporcionando uma facilidade maior para trabalhar com matrizes, possibilitando que o algoritmo consiga transitar entre sistemas SISO e MIMO, apenas modificando as matrizes de ganhos da estratégia de controle.

Baseando-se no diagrama da Figura 1, nota-se que a equação de controle é dada pela Equação 4.1:

$$u(t) = -K_i \cdot \int_0^t Erro.dt - K_e \cdot x(t) \quad (4.1)$$

Para realizar todas as operações entre as matrizes, utilizou-se de métodos disponíveis na biblioteca utilizada, sendo estes métodos descritos no github do desenvolvedor.

Contudo, para implementar essa equação no microcontrolador, é necessário encontrar os estados do sistema. Segundo [31], o estado $x(t)$ pode ser relacionado com a própria saída $y(t)$ para sistemas de primeira ordem, todavia, para sistemas onde não tem-se acesso a todos os estados, pode-se implementar um método para estimar os estados necessários, complementando o Algoritmo 2 em questão.

Assim como no Algoritmo 1, os cálculos são realizados em intervalos de 25 *ms*, sendo assim, mantendo um padrão entre os algoritmos desenvolvidos, todavia, este tempo pode ser alterado de acordo com a necessidade da planta.

Da mesma forma que o Algoritmo 1, o Algoritmo 2 pode ser dividido em duas frentes também, sendo estas: Inicialização das matrizes do controlador e cálculo da ação de controle. A inicialização das matrizes de entrada são realizadas no *void setup()*, outrora, os cálculos são realizados de forma cíclica no *void loop()*, desta forma a ação de controle é efetuada de forma cíclica a cada 25 *ms*. Abaixo em 2 é apresentado o pseudocódigo utilizado para a construção do Algoritmo utilizado para efetuar o controle por seguimento

de referência por integradores.

Algoritmo 2: Classe Espaço de Estados

Entrada: Ganhos: k_I , k_E , x

Saída: Controle $u(t)$

início

 Inicializa a biblioteca BasicLinearAlgebra.h

 Define as matrizes de ganhos k_E e k_I e as matrizes de estados x e controle u (

 Essas matrizes são expandidas/comprimidas de acordo com o tamanho do sistema, isto é, SISO ou MIMO)

 Define o Setpoint e o período de amostragem

if $T_s > (0.025)$ **then**

 Realiza uma leitura do sensor

 Filtra os dados coletados do sensor com janela móvel

 Calcula o desvio entre a leitura e o Setpoint

 Calcula a diferença de tempo entre uma execução de máquina (dT)

 Calcula a ação integral aplicando um somatório: $I += Erro \cdot dT$

 Salva a leitura do(s) sensor(es) dentro da matriz de estados (x)

 Calcula a ação de controle $u = -K_i \cdot I - K_e \cdot x$

 Ação anti wind-up

 Implementação de saturação virtual de acordo com as limitações do hardware

 Escreve o valor de controle na saída PWM referente aos atuadores

else

 | Aguarda até o T_s ser maior que o período de amostragem

end

fim

4.4 Comunicação Serial com Matlab

Para realizar a leitura dos dados, utilizou-se o Matlab versão estudante, onde fez-se uso de pacotes de comunicação serial para adquirir os dados resultantes de cálculos realizados no hardware de baixo custo.

Utilizando o Matlab para visualizar os dados de controle, atrasamos um pouco o processo de cálculo, entretanto, para esta planta que pretende controlar temperatura (uma variável lenta), o tempo de amostragem necessário não é alto, sendo assim, não impactando negativamente no controle. A seguir, é apresentado o pseudocódigo 3, do qual, retrata o procedimento necessário para comunicar o software com o hardware, nesta comunicação o Matlab coleta todas as amostras e salva em vetores, com o fim do experimento estes dados são plotados e podem ser analisados.

Algoritmo 3: Aplicação dos métodos

início

- Identifica o Hardware
- Exclui qualquer comunicação já existente entre o hardware e o software em questão
- Inicializa uma nova comunicação
- Faz a leitura dos dados da serial
- Tratar os dados da serial, convertendo para as unidades desejadas
- Salvar dados em um vetor
- Plotar vetor com dados salvos

fim

5 Projetos dos controladores

Para os primeiros testes, fixou-se a lâmpada em 100% da sua capacidade de trabalho, onde o controle de temperatura é feito por meio do cooler. Baseado nisto, neste primeiro instante, é de interesse projetar um controlador para a Equação 5.2. Este projeto será realizado em tempo contínuo, posteriormente discretizado para implementação no sistema físico.

5.1 Malha Aberta

Nos testes em malha aberta, o sistema foi submetido a degraus aplicados em diferentes atuadores e em diferentes condições, sendo estes importantes para avaliarmos o tempo de resposta natural do sistema a ser controlado.

No primeiro teste, manteve-se o cooler em uma atuação de 50%, e aplicou-se um degrau na lâmpada, e em cima da resposta do sistema, pode-se extimar uma possível função transferência, baseando-se nas estratégias de identificação fornecidas por [14]. Está função transferência é apresentada na Equação 5.1.

$$G(s) = \frac{4.07}{308.7s + 1} \quad (5.1)$$

No segundo teste, ligou-se a lâmpada em aproximadamente 50% da sua capacidade máxima de trabalho, posteriormente aplicou-se um degrau no cooler, onde avaliou-se o comportamento do sistema, resultando na função transferência 5.2.

$$G(s) = \frac{-7.08}{270.9s + 1} \quad (5.2)$$

Contudo, estas equações podem ser representadas em espaço de estados, para isso, basta utilizar a função $SS()$ do Matlab passando como parâmetro a função transferência a ser convertida. A partir desta conversão, é possível obter as equações 5.3 e 5.4 abaixo:

$$\begin{cases} \dot{x} = -0.0032 \cdot x(t) + 0.1250 \cdot u(t) \\ y = 0.1055 \cdot x(t) \end{cases} \quad (5.3)$$

$$\begin{cases} \dot{x} = -0.0037 \cdot x(t) + 0.1250 \cdot u(t) \\ y = -0.2091 \cdot x(t) \end{cases} \quad (5.4)$$

Estas equações, e estes métodos de aplicações, são utilizados como base para o ajuste do controlador, isto é, utilizando funções do Matlab, é possível descobrir parâmetros, dos quais, são utilizados no controle do processo.

Na Figura 26, é possível visualizar o comparativo entre as funções transferências obtidas por meio de experimentos e as FT estimadas por meio das sugestões de cálculos sugeridos em [14] para estimar as curvas de primeira ordem.

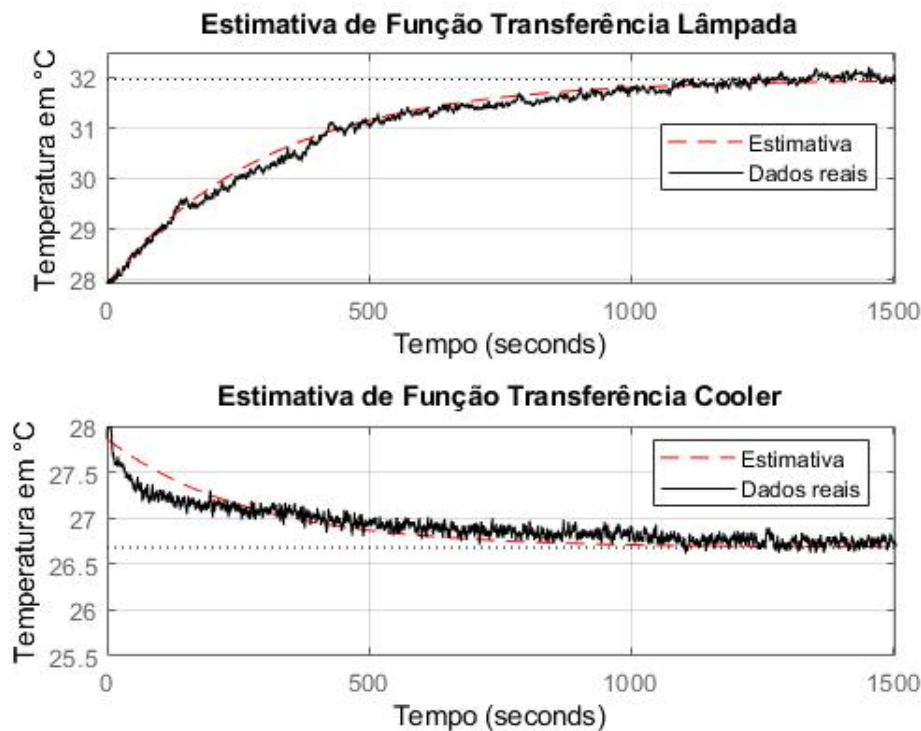


Figura 26 – Comparativo entre as respostas em MA e as FT estimadas.

5.1.1 LGR do Sistema de resfriamento

Assim como foi realizado na revisão bibliográfica, é necessário identificar os pólos do sistema, entretanto, como o sistema esboçado pela Equação 5.2 é lento, isto é, tem um $t_{5\%}$ de aproximadamente 1060 segundos, o pólo dessa FT é muito próximo de zero, portanto, é necessário cuidado para que o sistema não se desestabilize com ganhos exagerados.

Para este projeto, definiu-se que o tempo de 200 segundos e o sobressinal máximo aceitado é de 15%, assim, é possível obter o LGR da Figura 27, do qual, exemplifica as limitações desejadas no sistema.

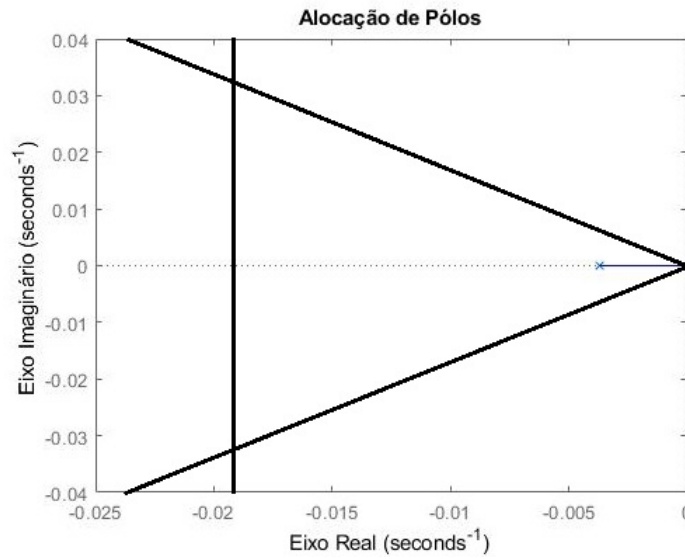


Figura 27 – LGR com as restrições do projeto.

Com a Figura 27 acima e com a especificações do projeto, pode-se definir a função do controlador como foi realizado na Equação 4. Tomando como base a mesma equação do controlador, e seguindo o mesmo passo à passo da revisão bibliográfica, uma das possíveis soluções, é a Equação 5.5 do controlador resultante, onde com esta equação, é possível obter a dinâmica em MF desejada.

$$C(s) = \frac{-45.962(s + 66.06)(s + 0.07904)}{s(s + 80)} \quad (5.5)$$

5.1.2 Discretização do Controlador

Para a discretização deste controlador e da Equação 5.2, utilizou-se o método bilinear ou Tustin. Estas equações foram discretizadas com o período de amostragem 25 ms, assim como foi explicado na seção 4, executando os cálculos na velocidade máxima do hardware escolhido.

Definido o período de amostragem, as equações resultantes da transformações são dadas abaixo, onde são respectivamente o processo discretizado e o controlador.

$$G(z) = \frac{-0.0003267z - 0.0003267}{z - 0.9999} \quad (5.6)$$

$$C(z) = \frac{-41.999(z - 0.998)(z - 0.09544)}{z(z - 1)} \quad (5.7)$$

5.1.3 Simulação do controlador

Com o sistema discretizado, simulou-se o sistema, onde este é submetido a um degrau unitário, sendo assim, como previsto, este degrau deve ser seguido com erro nulo, além de rejeitar perturbações no processo. Na Figura 28, é possível visualizar o funcionamento do sistema em MF, onde o mesmo atende todos os requisitos estabelecidos, tais como:

- Erro nulo ao degrau
- $t_{5\%} \leq 200$ segundos
- Overshoot $\leq 15\%$

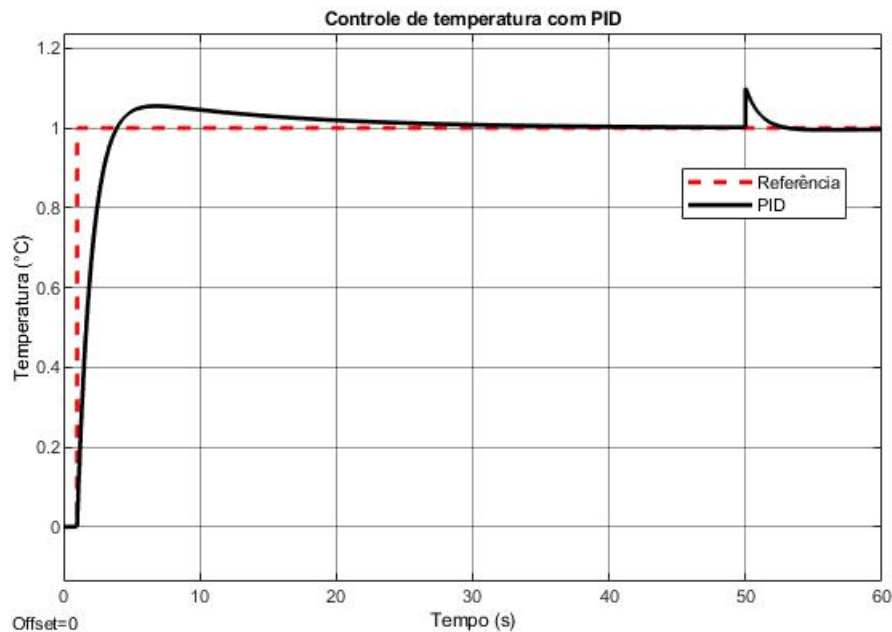


Figura 28 – Controle PID discreto.

Utilizando o mesmo ajuste, realizou-se um ensaio com o PID 2DOF, onde acrescentou-se os parâmetros atenuantes b e c iguais a 0.8, onde possibilitou a retirada do overshoot de 9.36%, e mantendo o settling time menor que 200 segundos. Na Figura 29 é possível visualizar os resultados obtidos.

Para obter os parâmetros do controlador, basta inserir a Equação 5.11 projetada, e utilizar a função pid do Matlab, onde todos os dados, inclusive o período de amostragem serão mostrados.

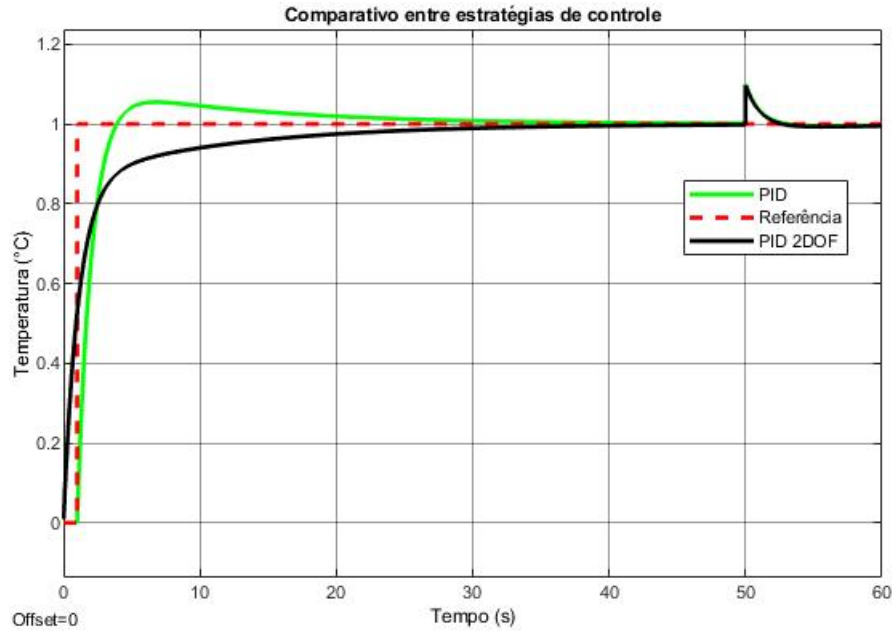


Figura 29 – Comparativo entre PID 2DOF e PID discreto.

5.2 Projeto em EE

Inicialmente, discretizou-se a representação em espaço de estados no tempo contínuo (Equação 5.4, sendo esta discretização realizada sob o mesmo período de amostragem fixado nas transformações bilineares anteriores. Realizando a transformação, as matrizes modificam-se, desta forma a representação do sistema, pode ser expressa na Equação 5.8.

$$\begin{cases} \dot{x}(k+1) = 0.9999 \cdot \dot{x}(k) + 0.0221 \cdot u(k) \\ y(k) = -0.2091 \cdot \dot{x}(k) \end{cases} \quad (5.8)$$

Com esta equação em mãos, no primeiro momento, projetou-se os pólos desejados, para que o comportamento do sistema em malha fechada fosse muito próximo dos resultados obtidos via simulação com o PID 2DOFs, entretanto, resultou em ganhos muito altos para o integrador, desta forma, implicando em um chaveamento no atuador entre ligado/desligado para o controle da temperatura nos experimentos práticos. Para contornar este problema, atrasou o sistemas com pólos mais lentos em malha fechada, enquadrando-se dentro das especificações de projeto, desta forma, foi fixado como pólos desejados em malha fechada os seguintes valores: -0.053 e -0.05 .

Com estes dados em mãos e com auxílio do Simulink, simulou-se o processo em tempo discreto, desta forma, a Figura 30 abaixo representa os resultados obtidos via simulação.

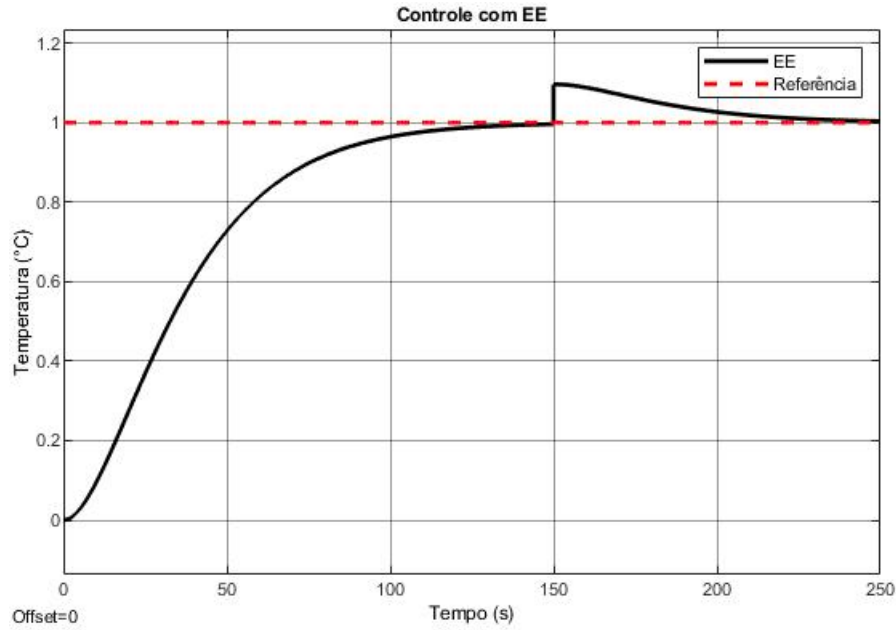


Figura 30 – Controle com Seguimento de referência com Integradores discretizado.

5.3 Projeto para o sistema de aquecimento

Para este atuador, pode-se tomar como FT correspondente a Equação 5.1, onde é a mesma função da revisão bibliográfica, entretanto, o projeto será diferente. Para este sistema, deseja-se um $t_{5\%} \leq 150s$ e um overshoot máximo de 15%. Seguindo os passos realizados na revisão bibliográfica, uma possível solução para o sistema é dada pela equação do controlador 5.9 abaixo:

$$C(s) = \frac{-0.1(s + 379.9)(s + 0.07896)}{s} \quad (5.9)$$

Utilizando a transformação de tustin para transformar as Equações 5.1 e 5.9 com o mesmo tempo de amostragem fixado anteriormente, obtém as Equações abaixo:

$$G(z) = \frac{0.00016479(z + 1)}{(z - 1)} \quad (5.10)$$

$$C(z) = \frac{-46.038(z - 0.998)(z + 0.6521)}{(z + 1)(z - 1)} \quad (5.11)$$

Diferentemente dos processos anteriores, onde o controlador resultante iria controlar a planta sozinho, neste caso ele irá trabalhar em conjunto com o controlador projetado na Equação 5.11, isto é, haverá duas malhas de controle, uma responsável pela atuação da lâmpada e outra responsável pela atuação do cooler e ambas controlando a temperatura da bancada didática.

6 Resultados e Discussão

A seguir, serão apresentados alguns testes realizados com as estratégias de controle, onde os algoritmos foram submetidos a experimentos com mudanças de setpoints constantes e submetidos a algumas perturbações, podendo assim, avaliar a capacidade de seguir a referência desejada, bem como, avaliar o rendimento do hardware quando o mesmo roda os algoritmos e envia os dados para o Matlab, sendo este software a interface de visualização.

6.1 PID 2DOF

O primeiro teste para executar o controlador PID 2DOF, utilizou-se apenas o cooler, isto é, a lâmpada sempre na atuação máxima e o controle de temperatura deu-se por meio da dissipação de calor. Nota-se que utilizando esta estratégia, o intervalo de controle é em torno de 27 à 40 graus Celsius, sendo o inferior o limite máximo de resfriamento possível com a lâmpada atuando em sua capacidade máxima de trabalho.

Para o ensaio, forneceu-se alguns degraus para o sistema e algumas perturbações (diminuindo a capacidade de trabalho da lâmpada), tornando possível avaliar a capacidade deste seguir a referência e obter um tempo de 5% menor que em malha aberta. Na Figura 31, é apresentado os resultados desta aplicação.

Os parâmetros utilizados, foram os calculados previamente na etapa de projeto do controlador PID, sendo eles:

- $K_p = -38$
- $K_i = -3$
- $K_d = -0.1$
- $b = c = 0.8$

Em relação a Figura 31, pode-se visualizar por meio do sinal de controle azul, que as perturbações ocorrem nos tempos de aproximadamente 250 s, 700 s e 180 s. Essas perturbações, são variações na atuação da lâmpada, isto é, durante um intervalo de tempo, o sistema de aquecimento sai da capacidade máxima de atuação e passa a assumir outros valores, impactando o processo durante o intervalo do tempo que o fenômeno ocorre.

Neste experimento, nota-se que o settling time prático diferenciou-se do valor simulado, isto deve-se as limitações do atuador de resfriamento, todavia, apesar das diferenças, o controlador permanece dentro das especificações de projeto, com o $t_{5\%} \leq 200s$.

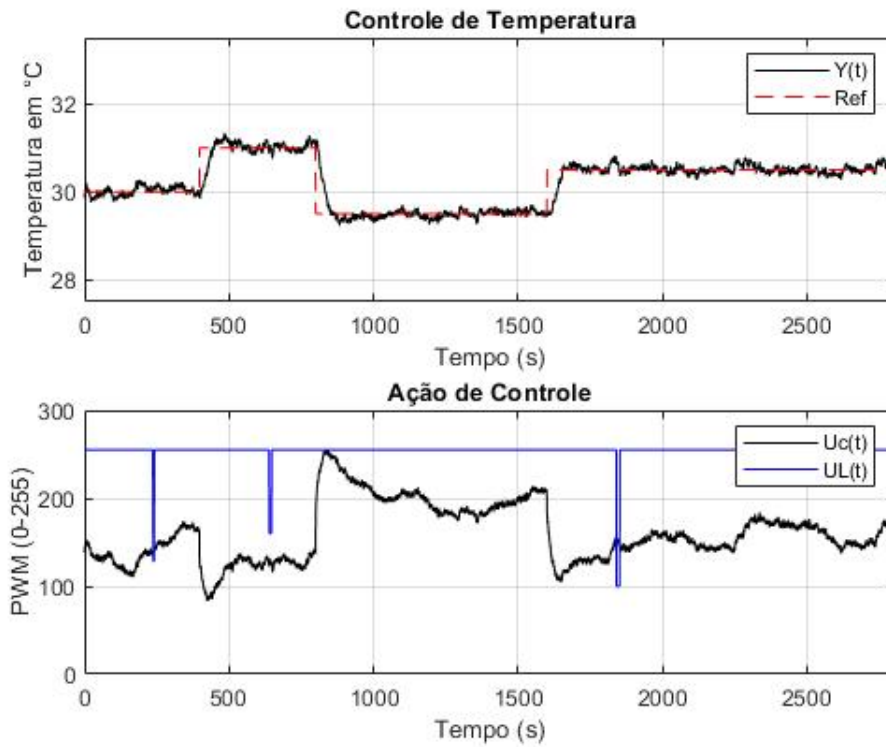


Figura 31 – Resposta do sistema com controlador PID 2DOF atuando sobre o cooler

6.2 PID

No segundo ensaio, alterou-se os parâmetros b e c para 1, assumindo um PID clássico, entretanto, os parâmetros K_p , K_i e K_d permaneceram os mesmo do ensaio do PID 2DOF. Neste ensaio, notou-se um maior esforço do controlador para alcançar e manter a temperatura na referência, isto é, o controle ficou mais agressivo, resultando em maiores oscilações além de muitas saturações na ação de controle. Estes resultados podem ser visualizados na Figura 32.

Analisando a Figura 32, nota-se uma oscilação no início da simulação, isto ocorreu pois com a alteração dos parâmetros b e c para 1, o controlador passou a necessitar de um esforço maior para alcançar a referência, implicando nestas oscilações, como é explicado na revisão bibliográfica o motivo destas diferenças entre as estratégias.

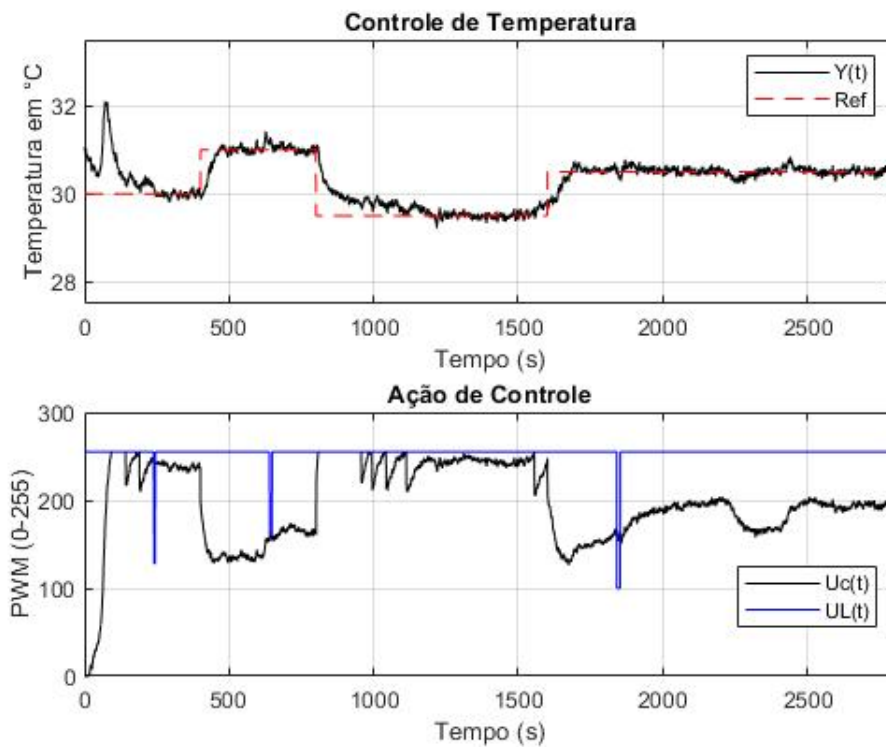


Figura 32 – Resposta do sistema com controlador PID atuando sobre o cooler

6.3 Comparativo entre PID e PID 2DOF

Posteriormente aos ensaios 1 e 2, comparou-se os dados coletados, onde pode-se comprovar as afirmações da revisão bibliográfica referente a ambos controladores. Na Figura 33, nota-se que durante todo o período do ensaio, o sinal de controle do PID 2DOF é inferior ao sinal de controle do PID, sendo assim, para o mesmo experimento a estratégia com dois graus de liberdade não saturou o controlador em nenhum momento, contrapartida a estratégia com um grau de liberdade saturou em diversos momentos, tanto para o resfriamento quando para o aquecimento.

Em relação aos tempos de 5%, nota-se que ambos são semelhantes, todavia, o PID clássico em alguns momentos tem um atraso ligeiramente maior para alcançar a referência, entretanto, para a rejeição de perturbação, ambos rejeitam com o mesmo comportamento.

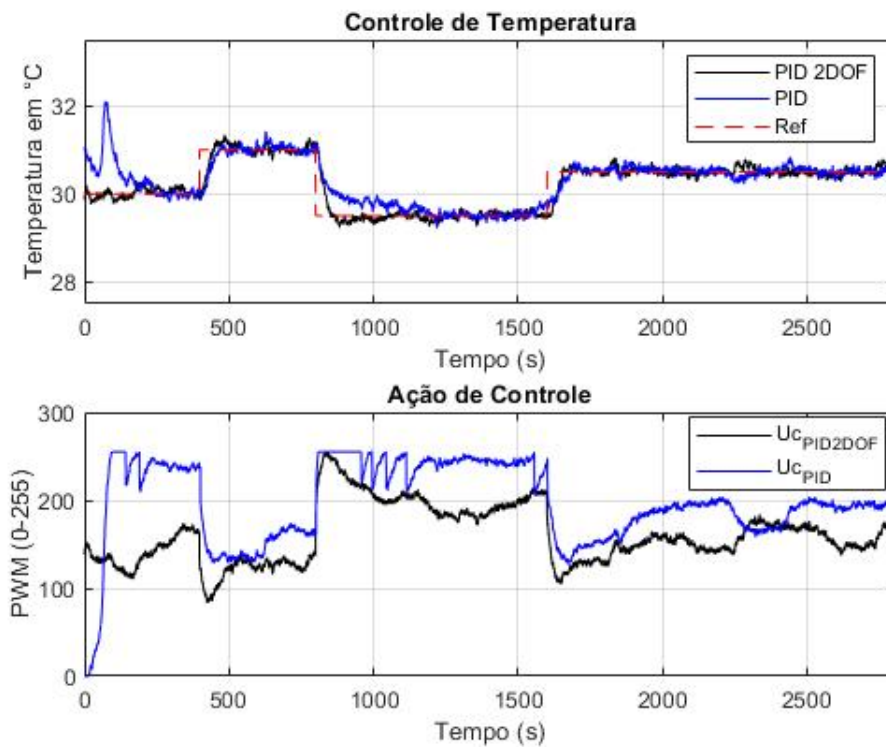


Figura 33 – Comparativo entre PID e PID 2DOF

6.4 Seguimento de referência com Integradores

No primeiro ensaio, efetuou-se o controle apenas utilizando o cooler, sendo assim, os limites de atuação permaneceram os mesmos dos ensaios feitos com o PID 2DOF. Utilizando esta estratégia de controle, nota-se um rendimento semelhante ao PID 2DOF, isto é, tempo de 5% de aproximadamente 50 segundos, entretanto, o controle tornou-se mais agressivo, implicando em um maior trabalho do controlador para manter a variável estável na referência. Os resultados referentes a estas análises, estão presentes na Figura 34. Os parâmetros resultantes da etapa de projeto foram os seguintes:

- $K_i = 4.0562$
- $K_e = 49.9131$

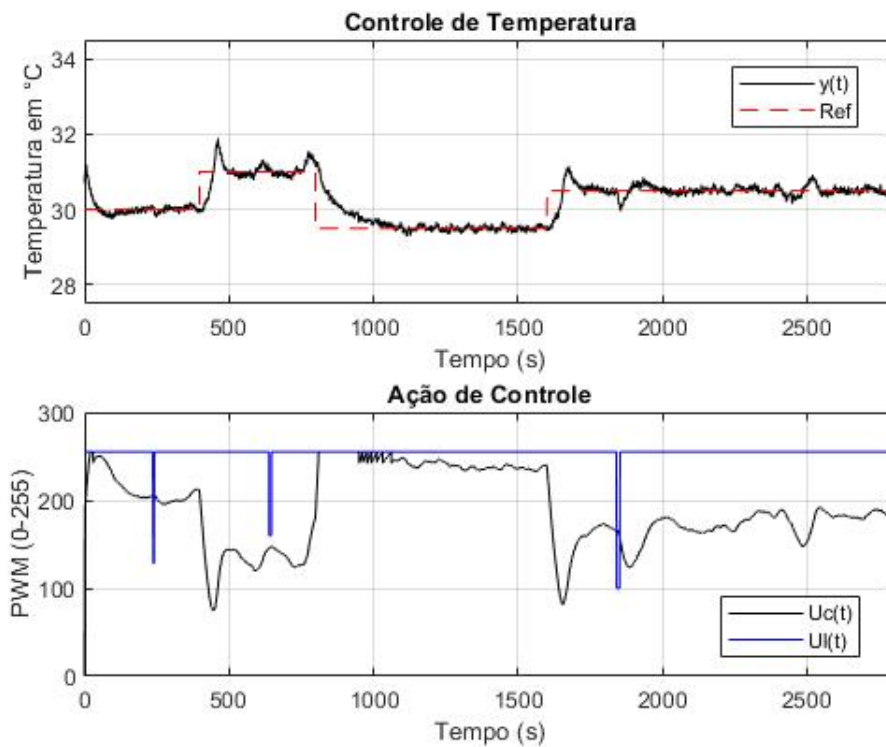


Figura 34 – Controle com seguimento de referência com integradores

6.5 Performance do Hardware

Para avaliar o desempenho do hardware de baixo custo na atividade proposta, dividiu-se a análise em duas frentes, sendo a primeira avaliando o tempo médio para a execução dos cálculos necessários para efetuar o controle da planta, e a segunda analisando a memória necessária para armazenar estes modelos.

6.5.1 Avaliando o hardware para um sistema SISO

No primeiro instante, executou-se os algoritmos sem período de amostragem, sendo este passo necessário para definir o tempo de execução mínimo necessário para processar um ciclo de cálculos, incluindo a comunicação com Matlab. Esse método foi executado para todas estratégias de controle. Na Tabela 6, pode-se visualizar os tempos necessários para processar um ciclo de código, isto é, coletar uma amostra do sensor, calcular o controle e enviar os dados para o Matlab.

Na Tabela 6, nota-se que o controle por seguimento de referência por integradores assume o maior período de amostragem, isto deve-se as operações matemáticas mais complexas envolvendo matrizes, sendo necessário a utilização de uma biblioteca externa para auxiliar no cálculo, desta forma, é natural que este tempo seja superior.

Tabela 6 – Desempenho do Hardware para os cálculos.

Estratégia	Número de Amostras	Tempo de Amostragem (ms)
PID	1	20.4
PID 2DOF	1	20.5
SS	1	23

Posteriormente, coletou-se a quantidade de armazenamento necessária para executar os controles na IDE do microcontrolador, onde estes dados foram alocados Tabela 7. Com estas informações, pode-se inferir quanto cada estratégia exige do hardware, isto é, o quanto de espaço é necessário para alocar cada estratégia, onde para o seguimento de referência com integradores faz-se a mesma ressalva realizada na análise da Tabela 6.

Tabela 7 – Ocupação dos algoritmos na memória do Hardware.

Estratégia	Memória de Programa (%)	Memória Dinâmica (%)
PID	17	33
PID 2DOF	17	33
SS	16	32

Esta ocupação da memória é importante, pois, quando o microcontrolador trabalha no limite de sua capacidade, os tempos de amostragem são alterados, sendo assim, o hardware necessita de mais tempo para a execução do algoritmo. Como é mostrado por [32], o tempo de amostragem é essencial para um bom controle, principalmente em sistemas onde a necessidade de atuação é extremamente veloz, exigindo-se mais do hardware em termos de velocidade de execução.

6.5.2 Avaliando o hardware para sistemas MIMO

Na segunda bateria de ensaios, simulou-se uma possível expansão do sistema, tanto com a adição de novas malhas PID 2DOF quanto aumentando as matrizes. Com isso, nota-se que o hardware passou a ser mais exigido no caso do PID 2DOF, onde para realizar os cálculos, os métodos são executados duas vezes, portanto, é natural o aumento do tempo de amostragem quando coparado com o mesmo ensaio realizado no item anterior. No caso do seguimento de referência com integradores, há somente a inicialização das matrizes necessárias, sendo assim, o período de amostragem permaneceu quase o mesmo. Na Tabela 8, pode-se visualizar os dados referentes a expansão do sistema, onde cada malha PID rendeu um acréscimo de aproximadamente 4.7 ms.

Contudo, a partir da Tabela 8, pode-se inferir que a utilização do seguimento de referência por integradores é bem quisto para sistemas com mais entradas e/ou variáveis em termos de projeto de controle, como afirma [17], não obstante, pode-se inferir também que em termos de hardware, conforme a dificuldade do sistema (com números de variáveis), no PID 2DOF faz-se necessário a adição de novas malhas para o controle da planta,

Tabela 8 – Desempenho do Hardware para os cálculos com 2 atuadores.

Estratégia	Número de Amostras	Tempo de Amostragem (ms)
PID	1	24.7
PID 2DOF	1	24.7
SS	1	23.2

onde exige-se mais do hardware quando compara-se ao seguimento de referência por integradores, do qual, a complexidade do sistema com matrizes maiores exige-se menos do microcontrolador do que a adição de novas malhas com controladores PID, vale ressaltar que para matrizes quadradas a partir de ordem 2, os valores permanecem o mesmo, quanto que para o PID a cada adição de uma malha soma-se 3% na memória dinâmica.

Por fim, assim como na Tabela 7, a Tabela 9 apresenta dados referentes a ocupação do algoritmo no hardware, onde assim como no tempo de amostragem, a adição de novas malhas de PID 2DOF para controlar a temperatura, impactaram negativamente no espaço necessário para armazenamento do algoritmo no hardware, por outro lado o seguimento de referência por integradores possui uma característica importante, apesar de ter seus índices expandidos, bem como o do PID 2DOF, estes índices permanecem constantes para sistemas cujas as matrizes são superiores a ordem 2, desta forma, enquanto a cada malha de controle acrescida no PID 2DOF é exigido um espaço de armazenamento maior, na estratégia dos seguimento de referência por integradores isto não é necessário.

Tabela 9 – Ocupação dos algoritmos na memória do Hardware com 2 atuadores.

Estratégia	Memória de Programa (%)	Memória Dinâmica (%)
PID	17	36
PID 2DOF	17	36
SS	19	39

6.5.3 Limitações do Hardware

O Arduino possui algumas limitações, onde pode-se dividi-las em três frentes, sendo elas:

- Tempo de amostragem;
- Número de entradas analógicas para os sensores/estados;
- Número de saídas PWM para acionamento de cargas;

Em relação ao tempo de amostragem, nota-se que para os algoritmos desenvolvidos, o *sample time* seja sempre maior que 25 ms, sendo assim, permanecendo acima da média de tempo necessário para se executar um ciclo de controle.

Outra limitação, é que pode-se medir apenas 6 saídas analógicas, onde estas podem corresponder aos possíveis estados da planta que deseja-se controlar. Uma alternativa que pode ser implementada no futuro de acordo com a complexidade do processo a ser controlado é um estimador de estados.

Para o controle PWM dos atuadores possui o mesmo número de entradas analógicas, todavia, se for necessário mais que 6 saídas de controle, é aconselhável a troca de hardware, visto que, diferentemente dos estados que pode-se estimar, as saídas de controle é necessário espaço no port para que os sinais sejam enviados ao driver de acionamento.

7 Conclusões e Trabalhos futuros

Com a realização deste trabalho, foi possível identificar as dificuldades da montagem de um sistema na prática, bem como realizar ajustes mecânicos para melhorar e otimizar o funcionamento da planta como um todo, ou seja, tanto a estrutura física do projeto, os hardwares utilizados quanto a comunicação com os softwares para a aquisição de dados.

Com relação as estratégias de controle, nota-se que ambas alcançaram bons resultados e com performance semelhantes, onde ambos os algoritmos ocuparam espaços semelhantes dentro do microcontrolador, além de serem capazes de realizar os cálculos e mostrarem os dados no Matlab com intervalo entre as amostras de aproximadamente 25 ms.

De modo geral, nota-se que para sistemas SISO, o PID de modo geral, é a melhor estratégia de controle, pois esta possui uma complexidade menor, entretanto, quando o número variáveis é expandido, a diferença entre o PID 2DOF e o SS diminuiu, contudo, pode-se inferir que conforme a complexidade do sistema aumenta, a melhor estratégia é utilizar SS, onde será necessário o ajuste de apenas dois parâmetros (K_i e K_e), enquanto que no PID há três parâmetros por malha de controle, além dos 2 parâmetros do PID 2DOF. Outro ponto importante a favor do SS, é a ocupação de memória, isto é, a partir de sistemas que exigem matrizes de ordem 2, esta estratégia ocupa sempre um espaço semelhante na memória do microcontrolador, enquanto o PID 2DOF necessita sempre de mais espaço, além de necessitar de mais tempo para executar um ciclo de cálculo por conta das diversas malhas que um sistema maior pode exigir.

Com relação ao hardware, nota-se que este pode ser utilizado em diversos processos, sendo que a condição mínima para a utilização do ATMEGA 328P para controlar alguma planta é que o tempo de amostragem do processo seja maior ou igual a 25 ms quando efetua-se o controle simultaneamente com a visualização dos dados via comunicação serial com Matlab, todavia, em outros desktops com melhores hardwares, esta comunicação serial entre software e hardware pode ser mais rápida.

Em trabalhos futuros, pode-se implementar estes algoritmos em diferentes hardwares, analisando a performance nestas distintas plataformas que podem ser encontradas no mercado, elegendo assim o melhor hardware de baixo custo para aplicar estas estratégias de controle.

Outro fator que pode ser otimizado na continuação do trabalho é a construção dos algoritmos, isto é, pode-se estudar estratégias para diminuir a complexidade dos mesmos, como por exemplo a escolha de um outro filtro para as amostras de estados coletados via porta analógica, onde este ocupa 19% da memória dinâmica quando a janela é de

100 amostras, sendo assim, podendo obter melhores resultados referente aos tempos de execução ou apenas diminuir a exigência com relação a memória.

Pode-se alocar em trabalhos futuros a implementação de um estimador de estados e um filtro de Kalman, expandindo os algoritmos desenvolvidos para sistemas mais complexos bem como adquirir melhores resultados referente ao controle.

Em suma, com este trabalho foi possível identificar as dificuldades em se construir sistemas físicos para a realização de experimentos, além de identificar problemas que só acontecem na prática, como por exemplo a aquisições de sinais, sendo esta impactada com ruídos, desta forma aprendeu-se como tratá-los impedindo que impactasse negativamente nos cálculos das estratégias de controle.

Referências

- 1 TALLIS, S. M. *Controle de Velocidade de um motor de Corrente Contínua por Realimentação de Estados Observados*. 57 p. Monografia (Bacharelado) — Universidade Federal de Ouro Preto, São Paulo, 2017. Citado 2 vezes nas páginas 13 e 14.
- 2 OGATA, K. *Modern Control Engineering*. 5th. ed. [S.l.]: PEARSON PRENTICE HALL, 2010. Citado 2 vezes nas páginas 13 e 20.
- 3 PUPO, M. S. *Interface Homem-Máquina para supervisão de um CLP em controle de processos através da WWW*. 113 p. Monografia (Bacharelado) — Universidade de São Paulo, São Paulo, 2002. Citado na página 13.
- 4 BRANDÃO, D. *Bloco funcional para controle FIELDBUS por variáveis de estado*. 134 p. Monografia (Mestrado) — Universidade de São Paulo, São Paulo, 2000. Citado na página 13.
- 5 ROSA, M. *Monitoramento de interrupções de fornecimento em uma rede elétrica monofásica residencial utilizando microcontroladores e supervisor SCADABR*. 52 p. Monografia (Tecnólogo) — Universidade Tecnológica Federal do Paraná, PONTA GROSSA, 2016. Citado na página 14.
- 6 SMIDIT, A. C. G. *Implementação de uma plataforma robótica controlada remotamente utilizando o Arduino*. Monografia (Bacharelado) — Universidade de São Paulo, São Carlos, 2013. Citado 2 vezes nas páginas 14 e 15.
- 7 KUMAR, M.; PATEL, V. V. Tuning of two degree of freedom pid controller for second order processes. *International Journal of Science*, India, 2015. Citado na página 14.
- 8 JOHANSSON, K. H. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, v. 8, n. 3, 2000. Citado na página 14.
- 9 GARCIA, C. *Modelagem e Simulação de Processos Industriais e de Sistemas Eletromecânicos*. São Paulo: Editora da Universidade de São Paulo, 1997. Citado na página 14.
- 10 YUCEL, T. *Modeling and Control of a heated tank system with variable liquid hold-up*. 107 p. Monografia (Mestrado) — Middle East Technical University, Turkey, 2013. Citado na página 14.
- 11 NETO, P. F. M. B. B. O.; QUEIROGA, S. L. M. Aplicabilidade dos microcontroladores em inovações tecnológicas. *CONNEPI*, Tocantins, 2012. Citado na página 15.
- 12 MAITELLI, A. L. Controladores lógicos programáveis. *Microchip Technology Inc*, 1998. Citado na página 15.
- 13 ÅSTRÖM, K. J.; HAGGLUND, T. *The Future of PID Control*. [S.l.]: Control Engineering Practice, 2001. Citado na página 15.

- 14 ÅSTRÖM, K. J.; HAGGLUND, T. *Advanced PID Control*. [S.l.]: Research Triangle Park: ISA - Instrumentation, Systems and Automation Society, 2005. Citado 5 vezes nas páginas 18, 19, 23, 47 e 48.
- 15 CUNHA, M. *Controle em Cascata de um Atuador Hidráulico: Contribuições Teóricas e Experimentais*. 177 p. Monografia (Doutorado) — Universidade Federal de Santa Catarina, Santa Catarina, 2000. Citado na página 18.
- 16 MATHWORKS. *Two-Degree-of-Freedom PID Controllers*. 2013. MathWorks PID 2DOF. Disponível em: <<https://www.mathworks.com/help/control/ug/two-degree-of-freedom-2-dof-pid-controllers.html>>. Acesso em: 21 dez. 2018. Citado na página 18.
- 17 OGATA, K. *Modern Control Engineering*. 4th. ed. [S.l.]: PEARSON PRENTICE HALL, 2002. Citado 2 vezes nas páginas 19 e 58.
- 18 OGATA, K. *Discrete-time control systems*. [S.l.]: Prentice Hall, 1995. Citado 3 vezes nas páginas 26, 27 e 28.
- 19 BRERO. *ANÁLISE DE SISTEMAS CONTÍNUOS AMOSTRADOS*. Citado na página 27.
- 20 G.F.FRANKLIN; J.D.POWELL; M.L.WORKMAN. *Digital control of Dynamic Systems*. [S.l.]: Addison- Wesley, 1990. Citado 2 vezes nas páginas 27 e 28.
- 21 HELDER. *Análise de Sistemas no Domínio da Frequência*. Análise de Sistemas no Domínio da Frequência. Disponível em: <<ftp://mecanica.ufu.br/LIVRE/Helder/FEMEC42060CSL/AulasCtrl/>>. Acesso em: 13 ago. 2019. Citado na página 28.
- 22 SILVA, J. M. G. da. *Wind-up da ação integral*. Wind-up da ação integral. Disponível em: <<http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node31.html>>. Acesso em: 21 jul. 2019. Citado na página 29.
- 23 BLOG, P. *Improving the Beginner's PID: Reset Windup*. Improving the Beginner's PID: Reset Windup. Disponível em: <<http://brettbeauregard.com/blog/2011/04/improving-the-beginnerpid-reset-windup/>>. Acesso em: 21 jul. 2019. Citado na página 29.
- 24 CHARLES, B. L.; MARCO, V. V. M. *AVR e Arduino Técnicas de Projeto*. Florianópolis: Edição dos Autores, 2012. Citado 2 vezes nas páginas 29 e 30.
- 25 ARDUINO. *Arduino Uno REV 3*. 2018. Arduino Uno REV 3. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 22 dez. 2018. Citado na página 29.
- 26 MIKROE. *Microcontroladores PIC - Online*. [S.l.]: mikroElektronika, 2003. Citado na página 30.
- 27 KOFUGI, S. T. *Raspberry PI 3 Modelo B*. 2018. Raspberry Pi 3. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/3444590/mod_resource/content/2/181-Raspberry%20Pi%203%20modelo%20B.pdf>. Acesso em: 22 dez. 2018. Citado na página 31.

- 28 SMAR. *Medição de Vazão*. 2010. Medição de Vazão. Disponível em: <<http://www.smar.com/newsletter/marketing/index40.html>>. Acesso em: 09 fev. 2019. Citado na página 34.
- 29 INSTRUMENTS, T. *LM35 0.5C Temperature Sensor with Analog Output and 30V Capability*. LM35. Disponível em: <<http://www.ti.com/product/LM35#>>. Acesso em: 21 jun. 2019. Citado 2 vezes nas páginas 35 e 36.
- 30 SANTOS, O. L. D.; JÚNIOR, J. D. S. M.; NEVES, M. R. M. *Sistema de Controle de Temperatura para uma Estufa com Monitoramento via Aplicativo*. 14 p. — Universidade Federal de Roraima, Boa Vista, 2016. Citado 2 vezes nas páginas 39 e 40.
- 31 CABRAL, E. L. L. *Transformação entre as formas espaço dos estados e função transferência*. Espaço de Estados. Acesso em: 22 jun. 2019. Citado na página 44.
- 32 SCHWARTZ, G. L.; RODRIGUES, L. F. M. *Análise de desempenho de controle embarcado em microcontrolador e em hardware-in-the-loop de um manipulador robótico*. 86 p. — Instituto Federal Fluminense, Campo dos Goytacazes - RJ, 2018. Citado na página 58.