

**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. Rodolfo Varraschim Rocha
Universidade Federal de Mato Grosso -
UFMT

Prof. Dr. Fabricio Parra Santilio
Universidade Federal de Mato Grosso -
UFMT

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

Cuiabá, XX de Julho 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 destes modelos deu-se no Arduino, onde alcançou-se resultados positivos 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 in this document, aim for the development and comparison of different classic control algorithms in low cost hardware in a prototype of a temperature didactic workbench. For the process monitoring and the actuation over the state variables, ~~chosen~~ ~~sensors with accessive prices~~, providing a didactic plant prototyping with a little financial resources. The classical control algorithms were developed based on models already established in the literature referring the PID 2DOF and state feedback with reference ~~follow-up~~ with integrators. The implementation of these models were in Arduino, where the algorithms got positive results despite the limitations of the hardware, however, there is necessity of a high sample rating for a better controller performance.

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

Lista de ilustrações

Figura 1 – Comparativo entre o Controlador PID tradicional e o 2DOF para seguir a referência e rejeitar perturbação em um sistema de segunda ordem. . .	19
Figura 2 – Comparativo entre os esforços realizados pelos controladores 2DOF e 1DOF.	19
Figura 3 – Rastreamento com Integradores.	20
Figura 4 – Sistema com realimentação de estados e rastreamento de referência . .	22
Figura 5 – Esforços do controlador para manter o sistema na referência estabelecida	22
Figura 6 – Região linear do processo.	23
Figura 7 – Plataforma Arduino.	24
Figura 8 – PIC 16F84	25
Figura 9 – Raspberry PI B	26
Figura 10 – Bancada didática	27
Figura 11 – Variáveis mais requisitadas na indústria.	28
Figura 12 – Circuito interno do LM35.	29
Figura 13 – Pinagem do LM35.	30
Figura 14 – Esquema de ligação com o microcontrolador para obter temperaturas no intervalo positivo do componente.	30
Figura 15 – Lâmpada DC utilizada para aquecimento da planta.	31
Figura 16 – Modelo de cooler utilizado para a dissipação de calor.	32
Figura 17 – Interface de acionamento e controle PWM para o cooler com alimentação de 12 Vdc.	34
Figura 18 – Interface de acionamento e controle PWM para lâmpada com alimentação de 12 Vdc.	35
Figura 19 – Comparativo entre as respostas em MA e as FT estimadas.	42
Figura 20 – Resposta do sistema com controlador PID 2DOF atuando sobre o cooler	44
Figura 21 – Resposta do sistema com controlador PID atuando sobre o cooler . . .	45
Figura 22 – Comparativo entre PID e PID 2DOF	46
Figura 23 – Atuando em dois atuadores com o controle PID 2DOF	47
Figura 24 – Controle com seguimento de referência com integradores	48
Figura 25 – Comparativo entre PID 2DOF e seguimento de referência com integradores	49
Figura 26 – Controle com seguimento de referência com integradores com dois atuadores	50
Figura 27 – Comparativo entre SS e PID 2DOF	51

Lista de tabelas

Tabela 1 – Comparativo de preços para aquisição de controladores PID Universais.	13
Tabela 2 – Comparativo das plataformas	26
Tabela 3 – Orçamento da bancada didática	28
Tabela 4 – Especificações da Lâmpada 64193NBU	32
Tabela 5 – Especificações do Cooler NONOIse A8025M12D	33
Tabela 6 – Desempenho do Hardware para os cálculos.	52
Tabela 7 – Ocupação dos algoritmos na memória do Hardware.	52
Tabela 8 – Desempenho do Hardware para os cálculos com 2 atuadores.	53
Tabela 9 – Ocupação dos algoritmos na memória do Hardware com 2 atuadores. .	54

Lista de abreviaturas e siglas

2DOF - Two degree of freedom

CC - Corrente Contínua

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 - State Space (Espaço de Estados)

U_c - Sinal de controle do cooler

U_L - Sinal de controle da Lâmpada

Sumário

1	INTRODUÇÃO	12
1.1	Problemática	14
1.2	Metodologia	14
1.3	Objetivo	15
1.3.0.1	Objetivos Específicos	15
1.4	Organização	16
2	REVISÃO BIBLIOGRÁFICA	17
2.1	PID 2DOF	17
2.1.1	Controlador PID com divisão das ações de controle	17
2.1.2	Simulação PID e PID 2DOF	18
2.2	Realimentação de Estados: Seguimento com Integradores	20
2.2.1	Simulação Realimentação de Estados	21
2.3	Wind - up	23
2.4	Hardwares de Baixo Custo	23
2.4.1	Arduino UNO REV 3	24
2.4.2	PIC 16F84	24
2.4.3	Raspberry PI 3	25
2.4.4	Comparativo entre as plataformas	26
3	INSTRUMENTAÇÃO DA BANCADA	27
3.1	Bancada	27
3.2	Sistema de medição	29
3.2.1	LM35	29
3.2.2	Esquema de ligação com o microcontrolador	30
3.3	Sistema de atuação	31
3.3.1	Lâmpada	31
3.3.2	Cooler	32
3.3.3	Drivers de acionamento	33
3.3.3.1	Cooler	33
3.3.3.2	Lâmpada	34
4	IMPLEMENTAÇÃO DOS ALGORITMOS	36
4.1	PID 2DOF	36
4.2	Seguimento de Referência por Integradores	38
4.3	Comunicação Serial com Matlab	40

5	RESULTADOS E DISCUSSÃO	41
5.1	Malha Aberta	41
5.2	Ajuste dos controladores	43
5.3	PID 2DOF	43
5.4	PID	45
5.5	Comparativo entre PID e PID 2DOF	46
5.6	PID 2DOF com dois atuadores	47
5.7	Seguimento de referência com Integradores	48
5.8	Comparativo entre PID 2DOF e seguimento de referência com integradores	49
5.9	Seguimento de referência com Integradores com dois atuadores	50
5.10	Comparativo entre as estratégias de controle com dois atuadores	51
5.11	Performance do Hardware	51
5.11.1	Avaliando o hardware com um atuador	52
5.11.2	Avaliando o hardware com dois atuadores	53
5.11.3	Limitações do Hardware	54
6	CONCLUSÕES E TRABALHOS FUTUROS	55
	REFERÊNCIAS	57

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 analógicas 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	Novus	N1100	942,00
2	Novus	N2000	1185,00
3	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 funções, pode-se descrever o sistema em **espaço de estados**. Esses **métodos de descrição do processo**, foram utilizados para **orçar** os parâmetros dos controladores por meio de um script no Matlab, sendo assim, a partir destes ganhos estimados, realizou-se ajustes nos parâmetros base para 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: Resultados e Discussão.

Capítulo 6: 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, destaca-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].

2.1.1 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 ao setpoint desejado, 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.



2.1.2 Simulação PID e PID 2DOF

Para verificar o funcionamento do controlador, será realizado uma simulação, onde será aplicado uma referência, e em um determinado instante t uma perturbação.

Na simulação, definiu-se um processo aleatório com a função transferência apresentada na Equação 2.2. Nesta planta, será simulado as diferenças existentes na atuação do PID 1DOF e o 2DOF, onde, ambos serão ajustados com os mesmo valores, todavia, será notório que o esforço para estabilização na referência do PID 2DOF será menor, quando comparado ao esforço realizado pelo PID clássico. Vale ressaltar que para ajustar o controlador, utilizou-se a função `pdtune(pid)` do Matlab, onde está retorna os valores de ajuste para a FT em questão, vale ressaltar que este método também será utilizado como base para ajuste do PID 2DOF no sistema real.

$$G(s) = \frac{1}{s^2 + 0.5s + 0.1} \quad (2.2)$$

Na simulação aplicou-se uma referência de 10 unidades, sendo assim, deseja-se que o sistema saia de um ponto qualquer, alcançando o ponto ordenado. Contudo, ajustou-se os controladores com os mesmo ganhos de K_p , K_i e K_d , sendo assim, notou-se que o PID 1DOF com estes ajustes possui overshoots, enquanto o 2DOF não apresenta estas características com os mesmos parâmetros, entretanto, além das constantes K_p , K_i e K_d , o PID 2DOF também foi ajustado com os valores de $b = 0.425$ e $c = 0.05$. Na Figura 1, é possível identificar os comportamentos relatados acima.

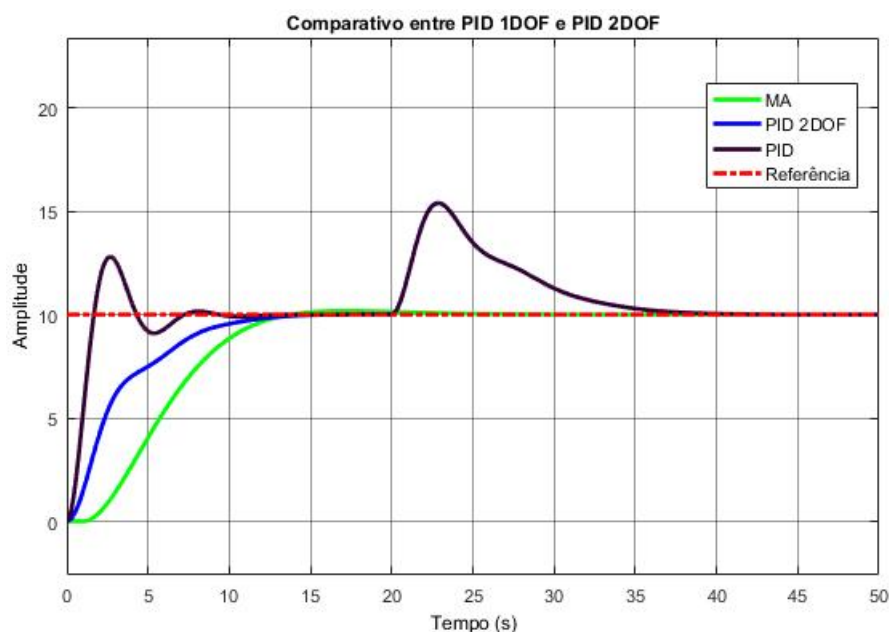


Figura 1 – Comparativo entre o Controlador PID tradicional e o 2DOF para seguir a referência e rejeitar perturbação em um sistema de segunda ordem.

Por fim, na Figura 2, pode ser verificado os esforços dos controladores, onde no primeiro instante do período de simulação, é desejado alcançar a referência estabelecida, e posteriormente, mante-lo na referência, rejeitando toda e qualquer perturbação, assim como demonstrado na Figura 1.

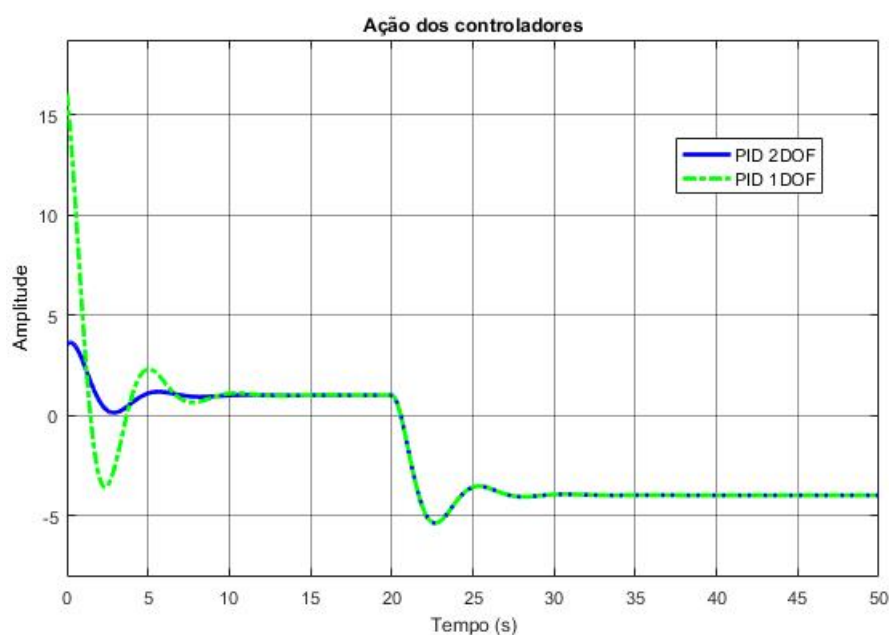


Figura 2 – Comparativo entre os esforços realizados pelos controladores 2DOF e 1DOF.

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 3.

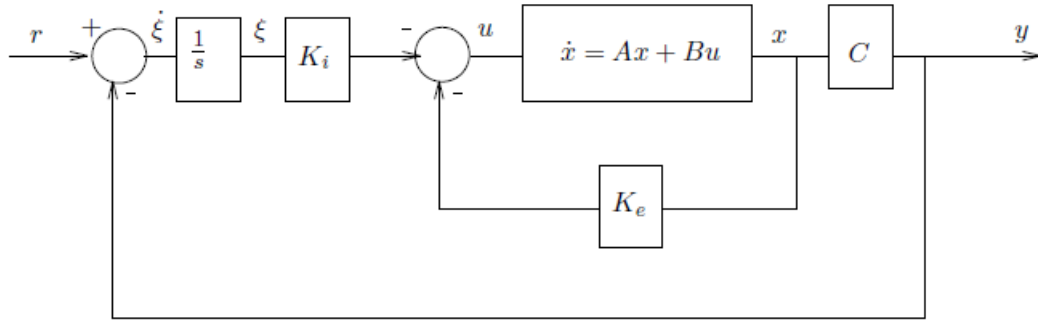


Figura 3 – Rastreamento com Integradores.

Nada obstante, pode-se representar o diagrama da Figura 3 de forma matricial (Equação 2.3), 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.

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix}}_{\dot{x}_a} = \underbrace{\begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}}_{A_a} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix}}_{\dot{x}_a} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_a} u + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_{E_a} r, \quad u = - \underbrace{\begin{bmatrix} K_e & K_i \end{bmatrix}}_{K_a} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix}}_{\dot{x}_a} \quad (2.3)$$

Contudo, há uma maneira simplificada de representar o diagrama da Figura 3, utilizando as equações de estado 2.4 e 2.5, 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 .

$$\dot{x} = Ax(t) + Bu(t), \quad u(t) = -K_e x(t) - K_i \xi(t) \quad (2.4)$$

$$\dot{\xi} = r(t) - y(t) = r(t) - Cx(t) \quad (2.5)$$

Em suma, segundo [2], o sistema em regime permanente, é descrito com a Equação 2.6 de controle $u(t)$ apresentada abaixo. Quando a Equação 2.6 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.

$$\bar{u} = -K_a \bar{x}_a = K_a (A_a - B_a K_a)^{-1} E_a \bar{r} \quad (2.6)$$

Na próxima seção, será apresentado uma simulação implementando esta técnica de controle no processo simulado anteriormente com os controladores PID, entretanto, para o ajuste dos ganhos desta técnica de controle, utilizou-se a função `place()` do matlab, entretanto para a aceleração do sistema, coleta-se as raízes do sistema com a função `eig()` e multiplica-se estes valores encontrados por uma aceleração desejada, tendo como resultante os valores dos pólos desejados em malha fechada para o projeto desejado.

2.2.1 Simulação Realimentação de Estados

Para verificar o funcionamento do controlador por realimentação de estados, foi realizado uma simulação, onde utilizou-se o mesmo processo (Equação 2.2) da simulação realizada com os controladores PID's.

Nessa simulação estabeleceu-se como referência um degrau unitário, portanto, partindo-se de um ponto qualquer, o sistema controlado deve alcançar essa referência. No pseudocódigo ?? presente nos anexos listados no final do documento, é possível visualizar a estratégia de implementação deste nas ferramentas Matlab e Simulink.

A simulação realizada, pode ser dividida em dois períodos distintos, sendo o primeiro, o tempo que o sistema controlado necessita para assentar-se na referência pré estabelecida, posteriormente, aplicou-se perturbações e o segundo período de tempo é referente as rejeições destas perturbações aplicadas. Na Figura 4 é possível visualizar os períodos divididos em dois espaços, como foi mencionado acima.

Por fim, na Figura 5, pode ser verificado o esforço do controlador, onde no primeiro instante do período de simulação é desejado alcançar a referência estabelecida, e posteriormente, mante-lo na referência, rejeitando toda e qualquer perturbação, assim como demonstrado na Figura 4.



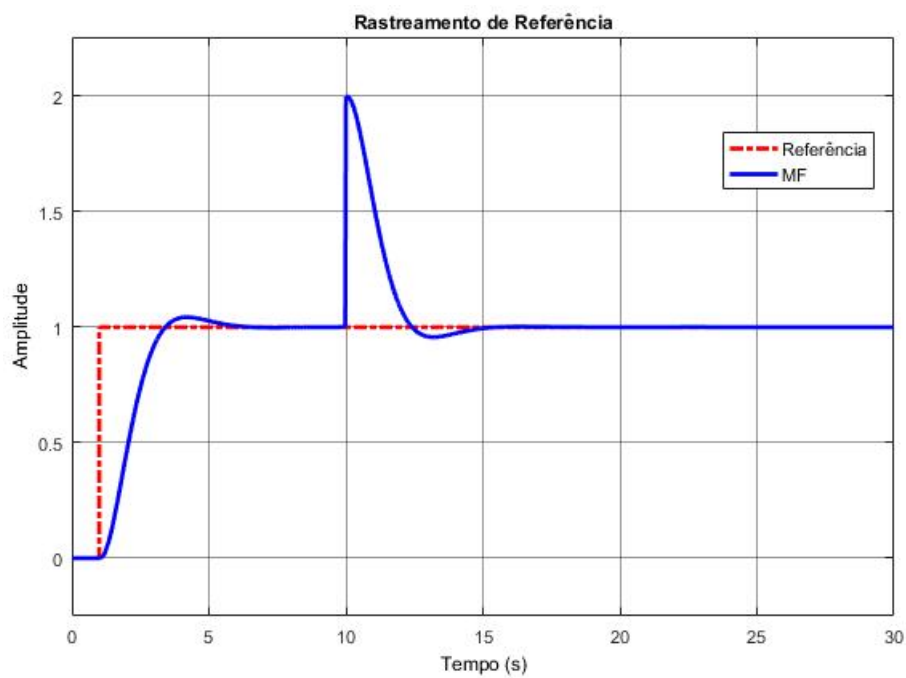


Figura 4 – Sistema com realimentação de estados e rastreamento de referência

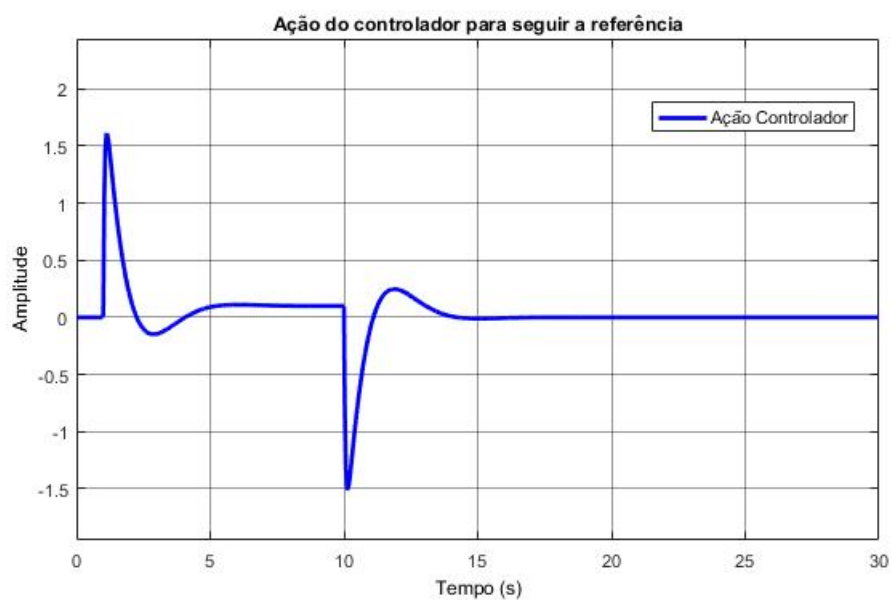


Figura 5 – Esforços do controlador para manter o sistema na referência estabelecida

2.3 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 [18].

Para decrementar o termo integral, pode-se tomar como exemplo as implementações práticas de [19], 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 6 abaixo:

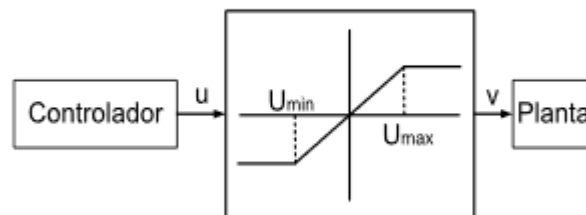


Figura 6 – Região linear do processo.

2.4 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 [20].

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.

2.4.1 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 [21].

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 [20].

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 7



Figura 7 – Plataforma Arduino.

2.4.2 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 elétricas 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 [22]. Na Figura 8 é possível visualizar o microcontrolador em questão.



Figura 8 – PIC 16F84 .

2.4.3 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 9) [23].



Figura 9 – Raspberry PI B .

2.4.4 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 10 abaixo, há a representação do sistema ao qual implantou-se os controladores, vale ressaltar que esta estrutura encontrava-se fechada durante os ensaios.

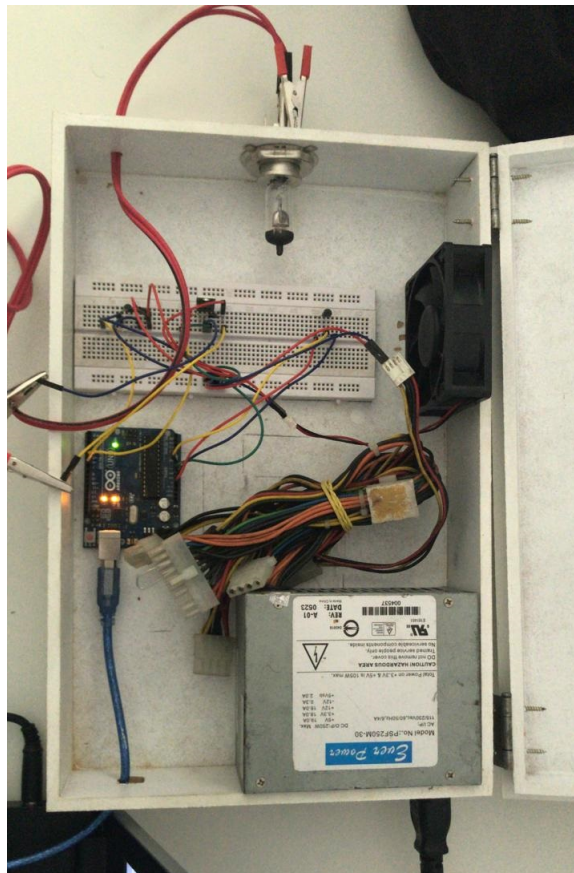


Figura 10 – Bancada didática

Nesta bancada, deseja-se controlar a temperatura, sendo esta, uma das mais presentes/requisitadas na indústria segundo [24], como é demonstrado pela Figura 11

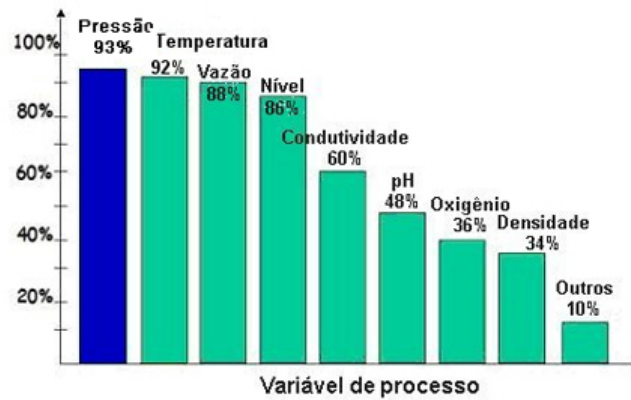


Figura 11 – Variáveis mais requisitadas na indústria.

Baseando-se nesta necessidade apresentada na Figura [24], 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 [25]. O circuito interno do transdutor apresentado na Figura 12 é encontrado no datasheet do componente, entretanto, na Figura 13 é possível identificar a pinagem do componente, onde +Vs é a entrada de tensão e V_{out} o sinal de temperatura.

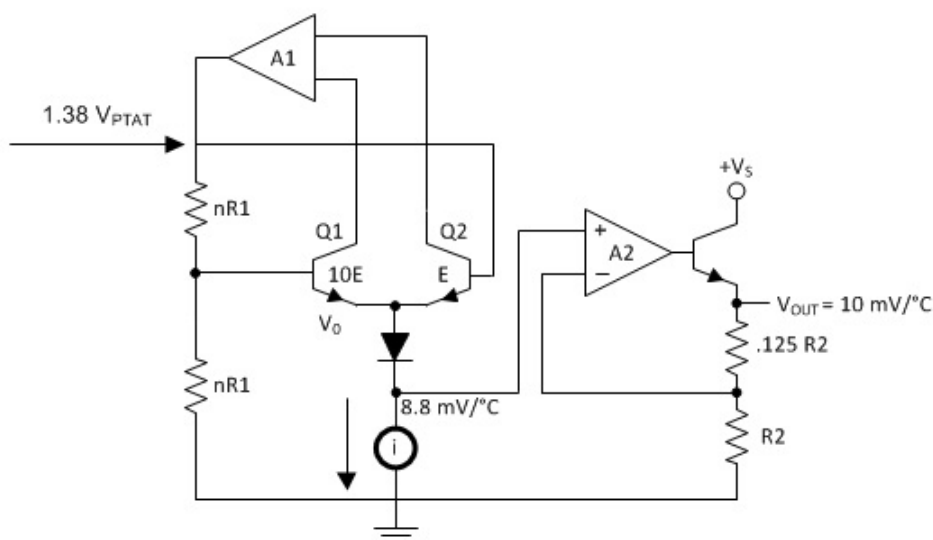


Figura 12 – Circuito interno do LM35.

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 15 é possível identificar as características físicas dos atuador.



Figura 15 – 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 16 é possível identificar um componente do mesmo fabricante semelhante ao utilizado na planta.



Figura 16 – 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 [26], 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 [26], 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 17, é possível visualizar o esquema elétrico utilizado no driver para o controle PWM da carga.

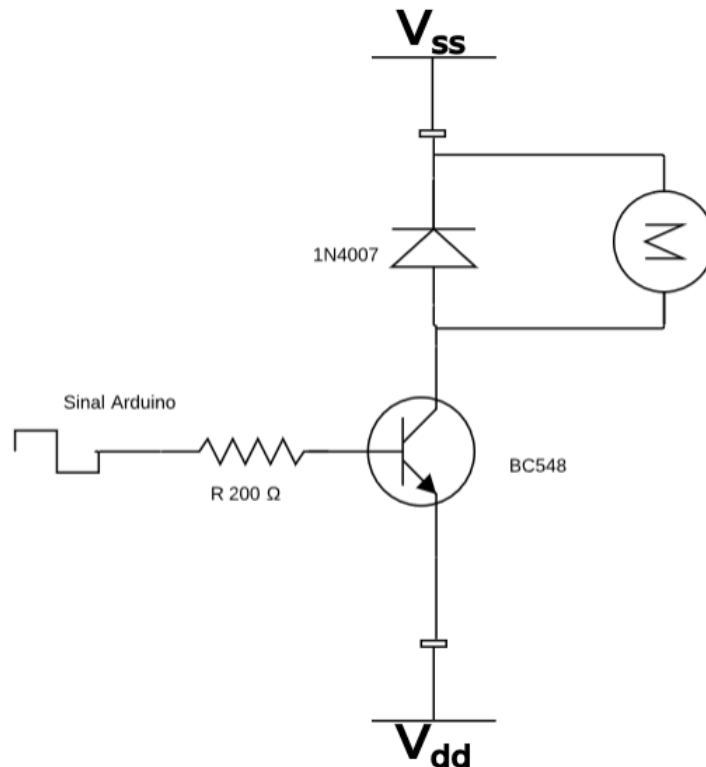


Figura 17 – Interface de acionamento e controle PWM para o cooler com alimentação de 12 Vdc.

3.3.3.2 Lâmpada

Assim como a interface de potência do cooler, o driver para a lâmpada foi baseado no modelo proposto por [26], entretanto, aplicou-se uma modificação no sistema original, buscando uma maior eficiência no sistema.

No driver proposto por [26], 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 18, 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.

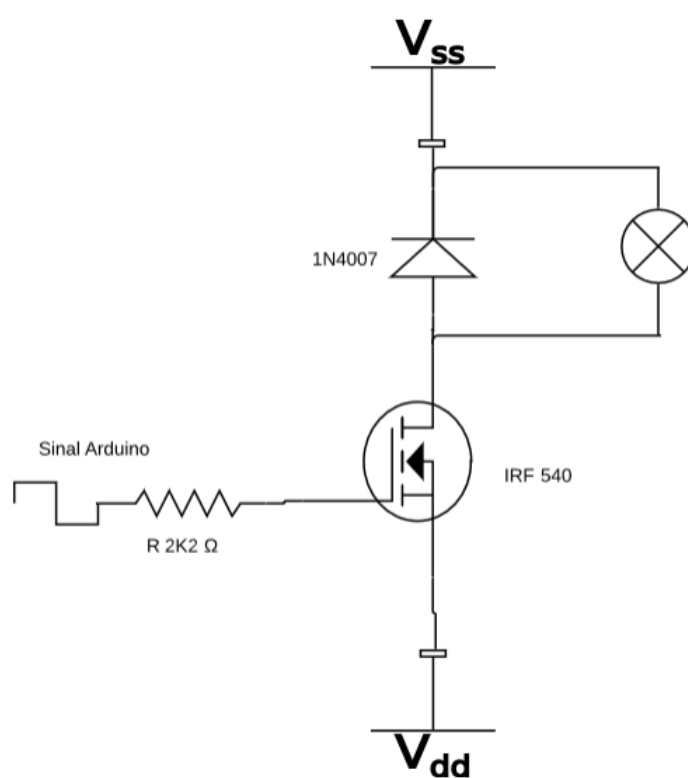


Figura 18 – Interface de acionamento e controle PWM para lâmpada com alimentação de 12 Vdc.

4 Implementação dos Algoritmos

As estratégias de controle foram implementadas na plataforma Arduino, 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.

A seguir, serão apresentados as estratégias utilizadas para o desenvolvimento dos algoritmos, descrevendo tanto as classes quanto a utilização dos métodos.



4.1 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, onde definiu-se o período de amostragem (T_s) de 25 ms.

O Algoritmo 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

if $Erro < 0.2$ **AND** $Erro > -0.2$ **then** **if** $(255 - aux) > 255$ **OR** $(255 - aux) < 0$ **then** $I = I/1.3$ **else** $I = I$ **end** Calcula a ação de controle (u), somando as três ações calculadas previamente

$$u = P + I + D$$

 Salva o valor real do controlador dentro de uma variável auxiliar aux **if** $u > 255$ **then** $u = 255;$ **if** $u < 0$ **then** $u = 0;$ **else** $u = u;$ **end**

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**

4.2 Seguimento de Referência por Integradores

Para esta 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 3, nota-se que a equação de controle é dada pela Equação 4.1:

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

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 [27], 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 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. 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

 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 sensor dentro da matriz de estados (x)

if $Erro < 0.2$ **AND** $Erro > -0.2$ **then**

if $(255 - aux) > 255$ **OR** $(255 - aux) < 0$ **then**

$I = I/1.3$

else

$I = I$

end

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

 Salva o valor real do controlador dentro de uma variável auxiliar aux

if $u > 255$ **then**

$u = 255;$

if $u < 0$ **then**

$u = 0;$

else

$u = u;$

end

 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 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 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.

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 pelo [14]. Está função transferência é apresentada na Equação 5.1.

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

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.

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

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:

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

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

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 19, é possível visualizar o comparativo entre as funções transferências obtidas por meio de cálculos sugeridos em [14] para estimar funções de primeira ordem.

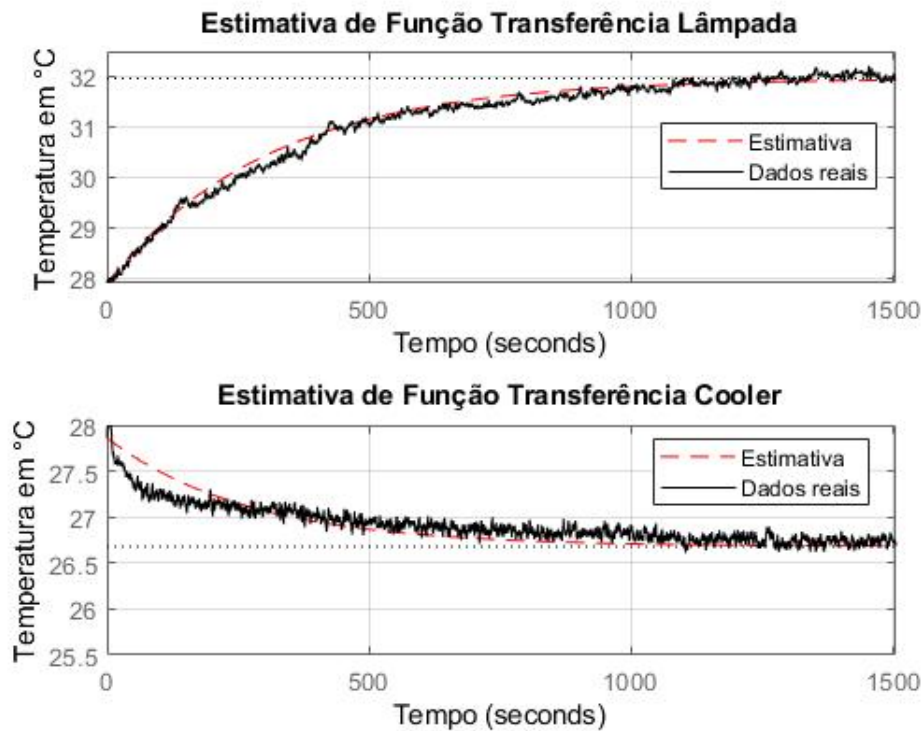



Figura 19 – Comparativo entre as respostas em MA e as FT estimadas.

Para gerar as Equações 5.1 e 5.2 discretizadas, utilizou-se o auxílio do Matlab, isto é, aplicou-se os vetores referente aos numerados e denominadores expressados no domínio da frequência, entretanto, adicionou-se o período de amostragem utilizado no microcontrolador, isto é, 25 *ms*, sendo assim, resultando nas expressões acima.

5.2 Ajuste dos controladores

Para ajustar os controladores, utilizou-se as estimativas de funções transferência, onde por meio das funções *pidtune()* e *place()* encontradas no Matlab, podemos ter como resultante alguns parâmetros, para este projeto, utilizou-se os valores resultantes apenas como base, isto é, com os ajustes sugeridos pelo Matlab, executava-se experimentos e de acordo com as respostas resultantes, ajustava-se as constantes de forma a acelerar o sistema e/ou diminuir oscilações do mesmo.

Utilizando esta estratégia de ajuste para diminuir o tempo de 5% em MA, foi possível obter os seguintes parâmetros para os controladores:

- PID 2DOF utilizando apenas o cooler para controlar a temperatura: $K_p = -38$, $K_i = -3$, $K_d = -0.1$, $b = c = 0.8$;
- PID 2DOF com dois atuadores, sendo o primeiro ajuste para o cooler e o segundo para a lâmpada respectivamente: $K_p = -38$, $K_i = -3$, $K_d = -0.1$, $b = c = 0.8$ e $K_p = 38$, $K_i = 20$, $K_d = 0.1$, $b = c = 0.8$;
- Seguimento de referência por integradores utilizando apenas o cooler para controlar a temperatura: $K_e = -6.9720$ e $K_i = -3$; 
- Seguimento de referência por integradores, sendo o primeiro ajuste para o cooler e o segundo para a lâmpada respectivamente: $K_e = -6.9720$, $K_i = -3$ e $K_e = 6.2163$, $K_i = 5.1600$.

5.3 PID 2DOF

O primeiro teste para simular 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 20, é apresentado os resultados desta aplicação.

Em relação a Figura 20, 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,

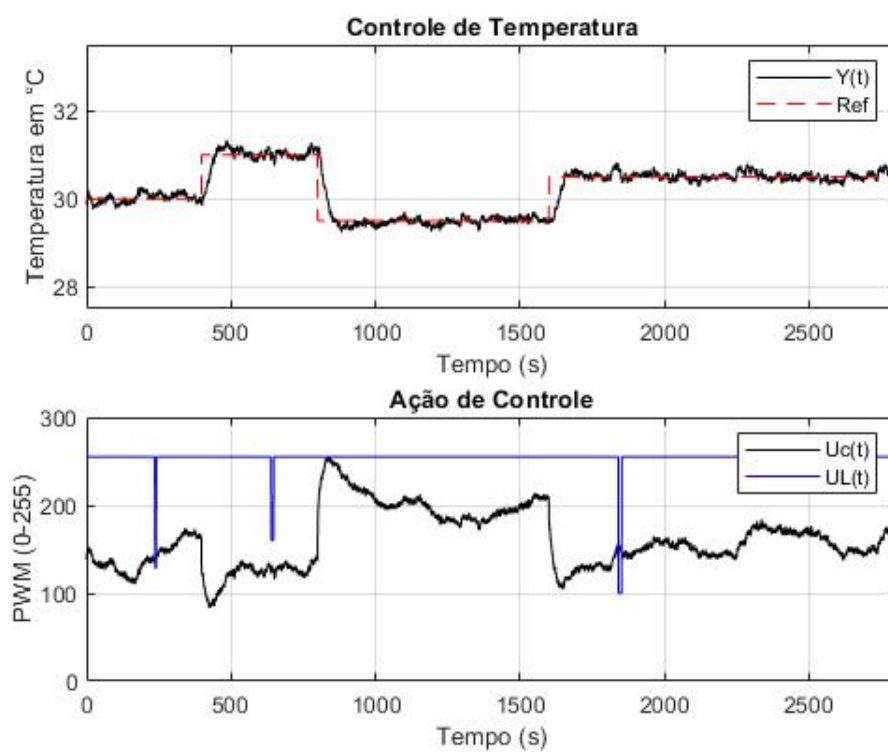


Figura 20 – Resposta do sistema com controlador PID 2DOF atuando sobre o cooler

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.

5.4 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 mesmos 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 21.

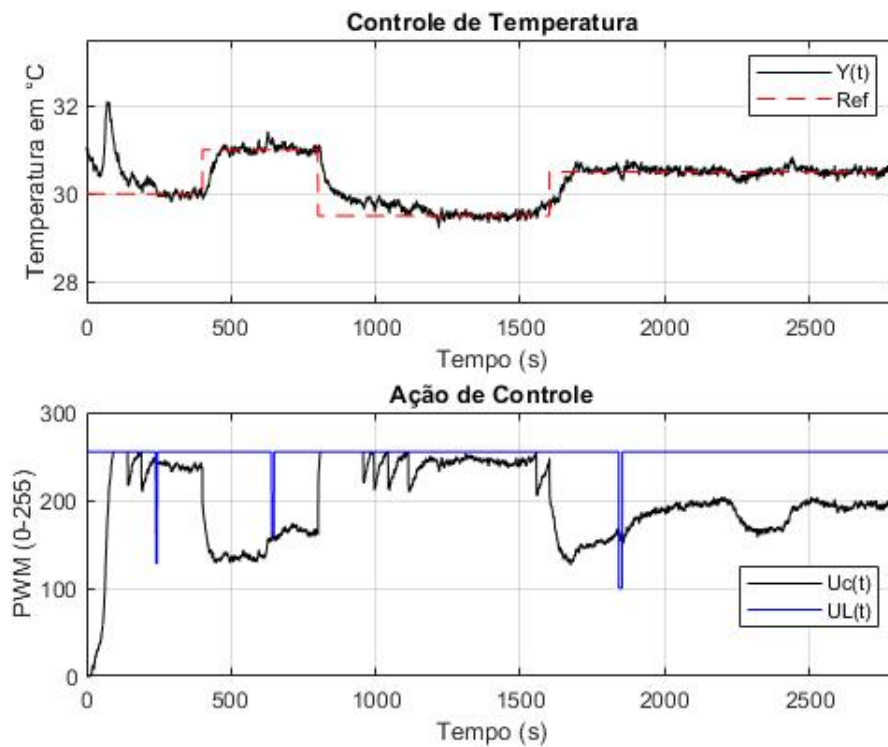


Figura 21 – Resposta do sistema com controlador PID atuando sobre o cooler

Analisando a Figura 21, 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.

5.5 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 22, 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.

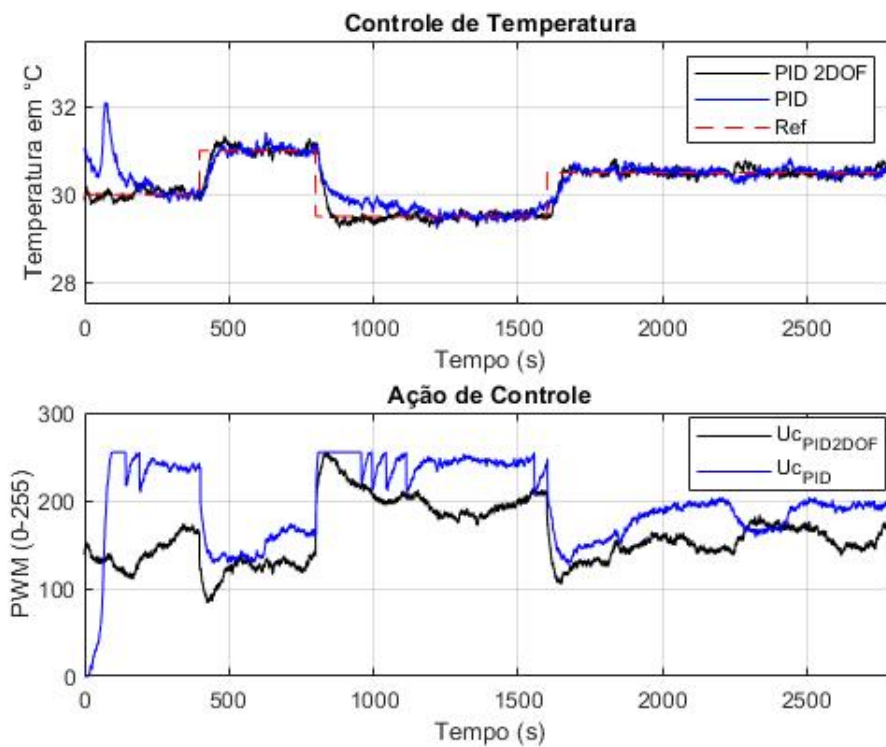


Figura 22 – Comparativo entre PID e PID 2DOF

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.

5.6 PID 2DOF com dois atuadores

Como o sistema possui dois atuadores, ajustou-se a estratégia de controle para contemplar a utilização da lâmpada, isto é, aplicou-se o controle tanto para a lâmpada quanto para cooler em malhas separadas, mas com os mesmos setpoints. Essa estratégia possibilitou a redução do intervalo de controle, isto é, com essa estratégia pode-se trabalhar com temperaturas menores do que 27 graus Celsius, levando em consideração a temperatura média do ambiente onde os experimentos foram realizados.

Na Figura 23 é retratado os resultados referentes ao controle da temperatura com dois atuadores, onde estes foram ajustados com parâmetros semelhantes ao primeiro experimento, entretanto, como era necessário acelerar a atuação da lâmpada, adicionou-se algumas unidades no parâmetro K_i .

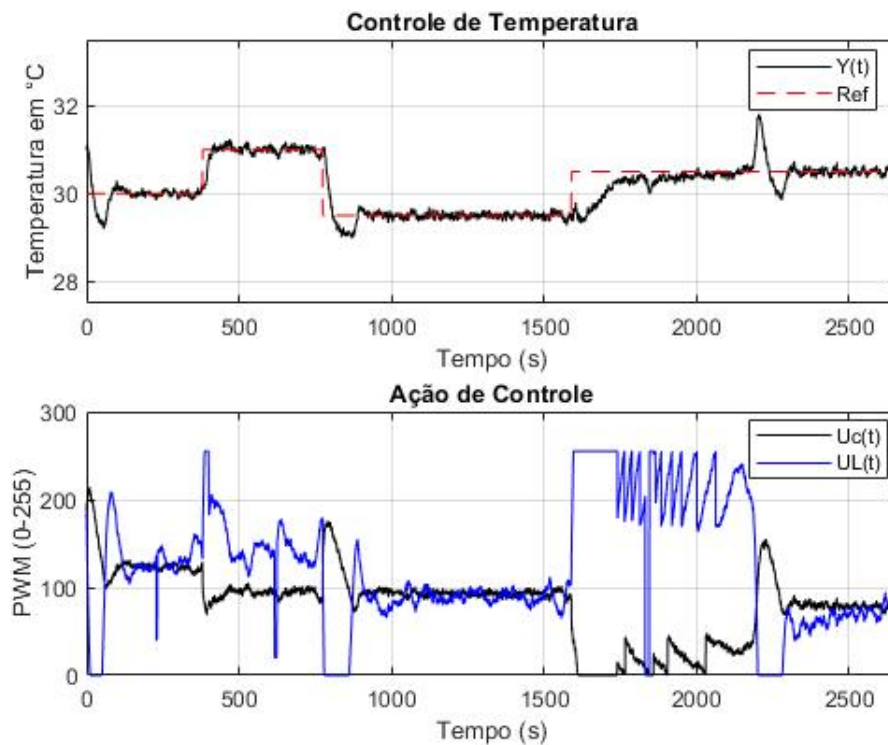


Figura 23 – Atuando em dois atuadores com o controle PID 2DOF

Com esta estratégia, pode-se inferir que o sistema torna-se mais complicado de controlar, isto é, ocorre maiores oscilações e a rejeição das mesmas perturbações aplicadas anteriormente, necessitam de um tempo maior, impactando o sistema com alguns overshoots.

5.7 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 24.

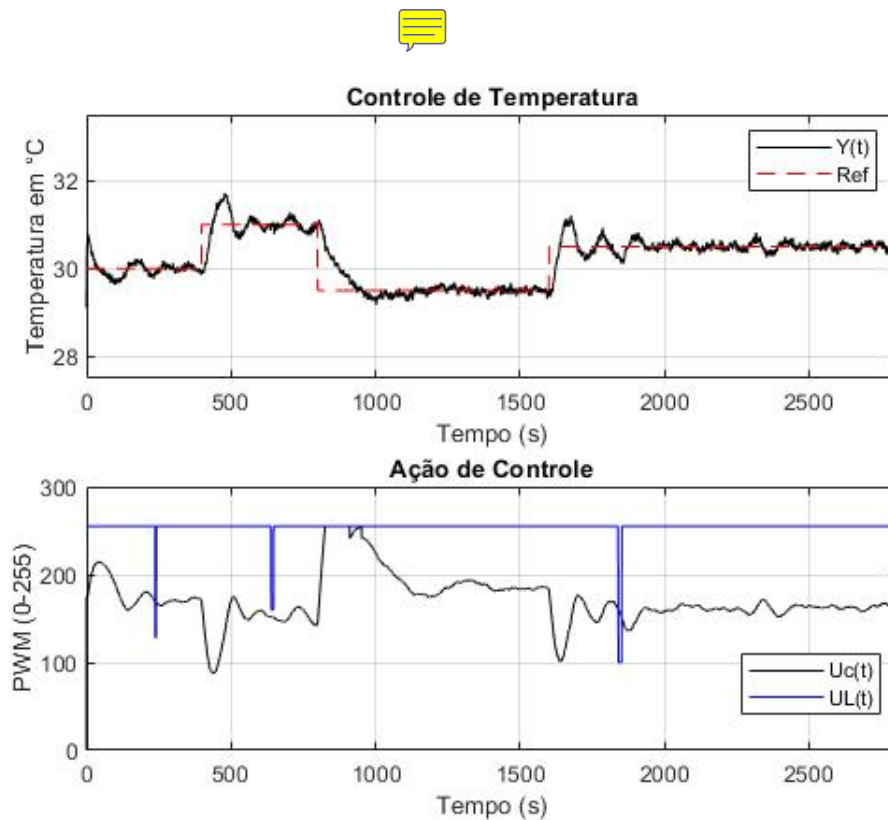


Figura 24 – Controle com seguimento de referência com integradores

5.8 Comparativo entre PID 2DOF e seguimento de referência com integradores

Neste comparativo, nota-se uma maior dificuldade na rejeição das perturbações por parte do seguimento de referência com integradores quando, todavia, os $t_{5\%}$ de ambos são semelhantes, além dos sinais de controle serem muito próximos, isto é, sem muitas saturações como é visto no *PID*.

Outro ponto a ser levado em consideração em relação a estratégia de controle seguimento de referência com integradores, é que aparecem alguns overshoots para alcançar a referência com os ajustes realizados, todavia, isto pode ser melhora com a implementação do filtro de Kalman, entretanto, nesta trabalho, será realizado somente a implementação da atividade de controle, nada obstante, estes incrementos podem ser desenvolvidos em trabalhos futuros.

Os resultados do comparativo, podem ser visualizados na Figura 25 abaixo:

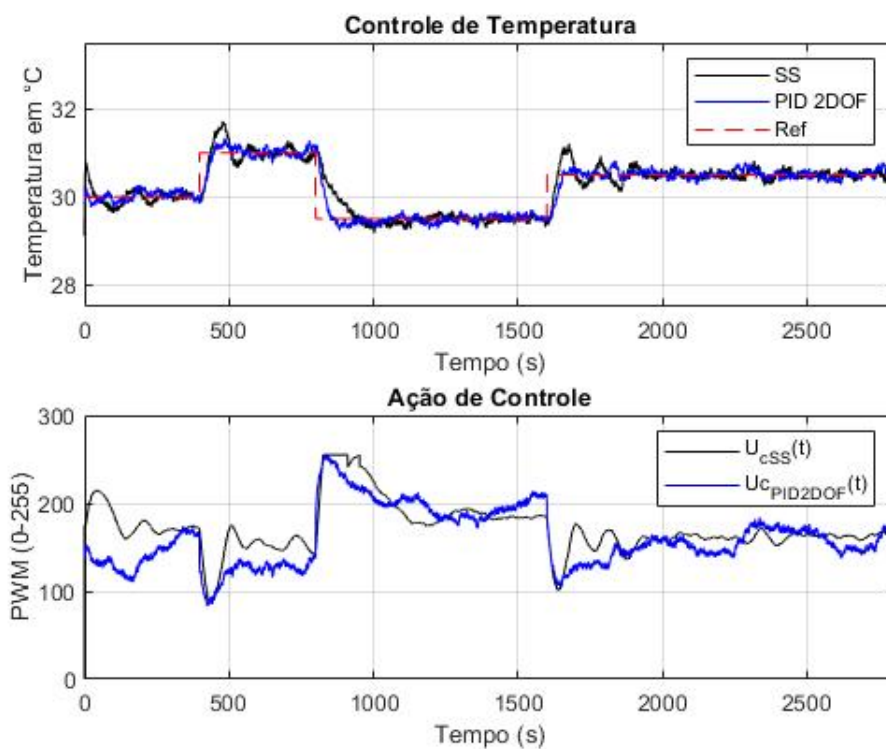


Figura 25 – Comparativo entre PID 2DOF e seguimento de referência com integradores

5.9 Seguimento de referência com Integradores com dois atuadores



Em um segundo momento, optou-se por controlar a temperatura utilizando os dois atuadores, desta forma, assim como nos experimentos anteriores com esta quantidade de atuadores, foi possível expandir os limites de atuação do sistema, porém, estes limites podem variar de acordo com a temperatura média do ambiente onde os ensaios serão realizados. Abaixo, na Figura 26, pode-se visualizar o experimento:

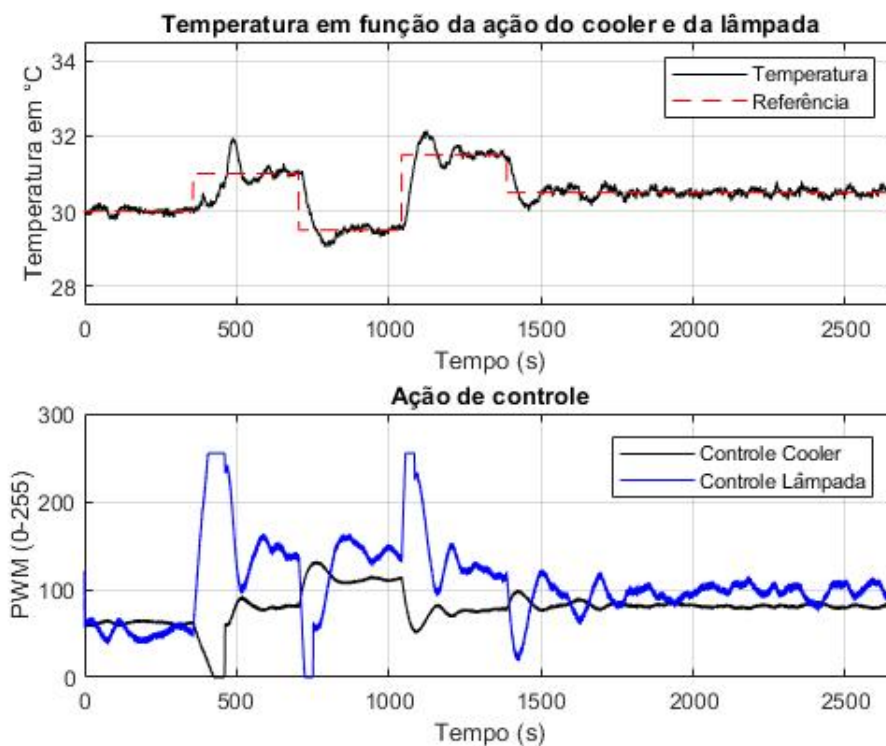


Figura 26 – Controle com seguimento de referência com integradores com dois atuadores

Para este experimento adicionou-se assim como no *PID 2DOF*, duas malhas de controle, sendo assim o sistema possui uma única entrada de setpoint para um estado (temperatura) a ser controlado.

Na Figura 26, nota-se algumas oscilações, assim como nas Figuras 24, todavia, as mesmas observações relatadas na seção anterior, aplicam-se neste caso.

5.10 Comparativo entre as estratégias de controle com dois atuadores

Afim de comparar as duas estratégias propostas neste trabalho, juntou-se os dados coletados nos experimentos com dois atuadores, onde estes foram plotados um sobre o outro resultando na Figura 27. Nesta figura, é possível identificar que ambos possuem tempos de 5% semelhantes, exceto no último degrau, todavia não houve oscilações como pode ser visto na outra estratégia. Observa-se também que para atuar em degraus que exigem o aumento da temperatura, o PID 2DOF consegue atuar com menos overshoots, entretanto, para degraus onde exigem a redução da temperatura e rejeição de perturbações, observa-se comportamentos semelhantes.

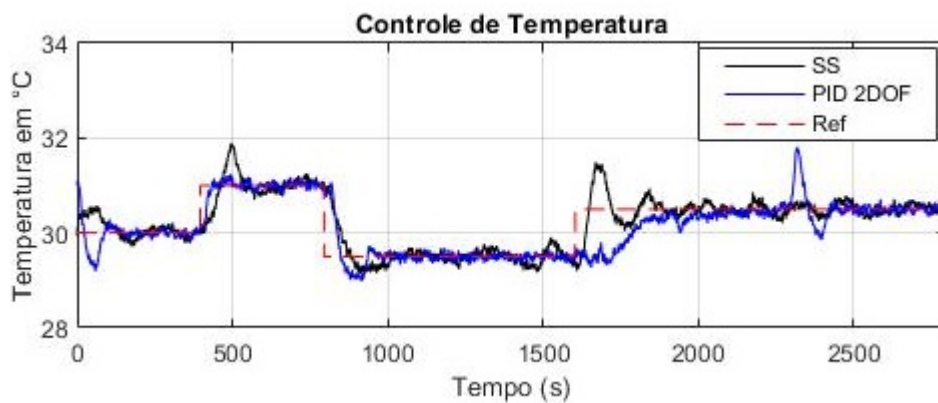


Figura 27 – Comparativo entre SS e PID 2DOF

5.11 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 controlando o processo apenas com um atuador e a segunda, controlando o processo com dois atuadores, todavia, dentro destes recortes, acrescentou-se a interferência do período de amostragem afim de estimar um intervalo que este hardware possa atuar com essas estratégias de controle.

5.11.1 Avaliando o hardware com um atuador

No primeiro instante, executou-se o algoritmo sem período de amostragem, sendo este passo necessário para definir o tempo de execução mínimo necessário para processar um ciclo do algoritmo, 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.

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

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. Para o sistema onde efetuou-se os testes, como as matrizes são mais simples por conta do sistema ser um SISO, o tempo de amostragem coletado foi de 23 *ms*, mas com o aumento das dimensões das matrizes, um maior tempo de amostragem será requisitado, portanto o limite mínimo de tempo de amostragem fica relacionado a complexidade do sistema, bem como o tamanho das matrizes que serão utilizadas para os cálculos da ação de controle.

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	16	33
PID 2DOF	17	33
SS	14	31

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 gasta mais tempo para a execução do algoritmo. Como é mostrado por [28], 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.

5.11.2 Avaliando o hardware com dois atuadores

Na segunda bateria de ensaios, utilizou-se dois atuadores para controlar a temperatura, sendo eles: cooler e lâmpada. 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. Como no seguimento de referência com integradores o método é chamado somente uma vez, isto é, depois de chamado o método inicializa-se apenas as matrizes necessárias, sendo assim, o período de amostragem permaneceu o mesmo, onde deve-se levar em consideração a complexidade do sistema, ou seja, quando maior o sistema, maior serão as matrizes, onde o hardware será mais exigido proporcionalmente. Na Tabela 8, pode-se visualizar os dados referentes aos experimentos com dois atuadores.

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

Estratégia	Número de Amostras	Tempo de Amostragem (ms)
PID	1	24.6
PID 2DOF	1	24.7
SS	1	23

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, 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.

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 via lâmpada e via cooler, impactaram negativamente no espaço necessário para armazenamento do algoritmo no hardware, por outro lado o seguimento de referência por integradores permaneceu com os mesmo índices.

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	14	31

5.11.3 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.

6 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 obtiveram bons resultados e com performance semelhantes, onde ambos os algoritmos ocuparam o 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 20 ms. 

De modo geral, nota-se que para sistemas de um estado, o PID com dois graus de liberdade, é a melhor estratégia de controle, pois esta possui uma complexidade menor, entretanto, quando passou-se a controlar a temperatura com duas variáveis, a diferença entre o PID 2DOF e o SS diminuiu, contudo, pode-se inferir que conforme adicionamos atuadores para serem controlados, a distância entre o PID 2DOF e o SS diminuem, sendo assim, a melhor estratégia será relativa as condições desejadas para o sistema em MF. Nada obstante, pode-se concluir também que o controle para um processo MIMO, é mais recomendado a estratégia SS, onde não é necessário regular dois controladores para dois sistemas que interagem entre si, mas sim, utilizar os cálculos das matrizes resultantes das funções de cálculo do Matlab.

Vale ressaltar que conforme a complexidade do sistema muda, o tempo de amostragem irá aumentar decorrente das operações matriciais que envolvem a resposta do controlador, deste modo, com um tempo de amostragem maior, os sistemas que podem ser controlados com esta estratégia utilizando este hardware ficam mais restritos.

Para o sistema onde ocorreram os experimentos, pode-se concluir que a melhor estratégia de controle seria o uso do PID 2DOF, pois o sistema é de primeira ordem e SISO, todavia, quando adiciona-se mais uma malha de controle, nota-se que em termos de exigência de hardware a estratégia utilizando seguimento de referência por integradores passa a ser melhor, pois o impacto das operações matriciais para o cálculo do controle é menor que a adição de malhas *PID* para o controle individual das variáveis desejadas.

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.

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 mesmo, 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 em cima do hardware.

Pode-se alocar em trabalhos futuros a implementação de um estimador de estados e um filtro de Kalman, onde pode-se expandir 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 12 e 13.
- 2 OGATA, K. *Modern Control Engineering*. 5th. ed. [S.l.]: PEARSON PRENTICE HALL, 2010. Citado 2 vezes nas páginas 12 e 21.
- 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 12.
- 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 12.
- 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 13.
- 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 13 e 14.
- 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 13.
- 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 13.
- 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 13.
- 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 13.
- 11 NETO, P. F. M. B. B. O.; QUEIROGA, S. L. M. Aplicabilidade dos microcontroladores em inovações tecnológicas. *CONNEPI*, Tocantins, 2012. Citado na página 14.
- 12 MAITELLI, A. L. Controladores lógicos programáveis. *Microchip Technology Inc*, 1998. Citado na página 14.
- 13 ÅSTRÖM, K. J.; HAGGLUND, T. *The Future of PID Control*. [S.l.]: Control Engineering Practice, 2001. Citado na página 14.

- 14 ÅSTRÖM, K. J.; HAGGLUND, T. *Advanced PID Control*. [S.l.]: Research Triangle Park: ISA - Instrumentation, Systems and Automation Society, 2005. Citado 4 vezes nas páginas 17, 18, 41 e 42.
- 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 17.
- 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 17.
- 17 OGATA, K. *Modern Control Engineering*. 4th. ed. [S.l.]: PEARSON PRENTICE HALL, 2002. Citado 3 vezes nas páginas 18, 20 e 53.
- 18 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 23.
- 19 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 23.
- 20 CHARLES, B. L.; MARCO, V. V. M. *AVR e Arduino Técnicas de Projeto*. Florianópolis: Edição dos Autores, 2012. Citado 2 vezes nas páginas 23 e 24.
- 21 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 24.
- 22 MIKROE. *Microcontroladores PIC - Online*. [S.l.]: mikroElektronika, 2003. Citado na página 24.
- 23 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 25.
- 24 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 28.
- 25 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 29 e 30.
- 26 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 33 e 34.
- 27 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 38.

- 28 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 53.