



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO, DE CIÊNCIAS EXATAS E EDUCAÇÃO
DEPARTAMENTO DE ENG. DE CONTROLE, AUTOMAÇÃO E COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Bruno Siewerdt

Automação de um Sistema de Aquecimento de Água por Acumulação

Blumenau
2024

Bruno Siewerdt

Automação de um Sistema de Aquecimento de Água por Acumulação

Trabalho de Conclusão de Curso de Graduação em Engenharia de Controle e Automação do Centro Tecnológico, de Ciências Exatas e Educação da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Ciro André Pitz, Dr.

Blumenau

2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Siewerdt, Bruno

Automação de um Sistema de Aquecimento de Água por
Acumulação / Bruno Siewerdt ; orientador, Ciro André Pitz,
2024.

48 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Blumenau,
Graduação em Engenharia de Controle e Automação, Blumenau,
2024.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Controle PID.
3. Automação. I. Pitz, Ciro André . II. Universidade
Federal de Santa Catarina. Graduação em Engenharia de
Controle e Automação. III. Título.

Bruno Siewerdt

Automação de um Sistema de Aquecimento de Água por Acumulação

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Engenheiro de Controle e Automação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação.

Blumenau, 02 de Dezembro de 2024.

Banca Examinadora:

Prof. Ciro André Pitz, Dr.
Universidade Federal de Santa Catarina

Prof. Guilherme Brasil Pintarelli, Dr.
Universidade Federal de Santa Catarina

Prof. Marcos Vinicius Matsuo, Dr.
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus pais, que sempre me incentivaram e colocaram minha educação em primeiro lugar.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de aquecimento automatizado para um *boiler* residencial, baseado em um controlador proporcional, integral e derivativo (PID) com sintonia automática, integrado a um sistema de monitoramento remoto. O sistema desenvolvido permite a leitura contínua da temperatura da água no *boiler*, garantindo que o aquecimento ocorra de maneira eficiente e otimizada. A sintonia automática do controlador PID ajusta os parâmetros de controle, assegurando uma resposta rápida e precisa às variações de temperatura, o que resulta em economia de energia e maior conforto para o usuário. Além disso, o projeto incorpora a funcionalidade de monitoramento remoto, permitindo que as temperaturas do *boiler* sejam acessadas a qualquer momento por meio de um aplicativo no celular. Essa integração com a nuvem não só facilita o controle da temperatura, mas também possibilita a configuração de horários programados para o acionamento do *boiler*, otimizando ainda mais o consumo de energia e oferecendo maior conveniência ao usuário. Resultados obtidos com o protótipo desenvolvido confirmam a eficácia do sistema proposto.

Palavras-chave: *Boiler* Residencial; Controle PID; Sistema IoT

ABSTRACT

This paper presents the development of an automated heating system for a residential boiler, based on a proportional–integral–derivative (PID) controller with automatic tuning, integrated with a remote monitoring system. The proposed system allows continuous temperature readings of the water in the boiler, ensuring that the heating process occurs in an efficient and optimized manner. The automatic tuning of the PID controller adjusts the control parameters, ensuring a quick and precise response to temperature variations, resulting in energy savings and increased comfort for the user. Furthermore, the project incorporates remote monitoring functionality, allowing the boiler temperatures to be accessed at any time through a mobile application. This cloud integration not only facilitates temperature control but also enables the scheduling of boiler activation times, further optimizing energy consumption and offering greater convenience to the user. Results obtained with the developed prototype confirm the effectiveness of the proposed system.

Keywords: IoT System; Residential Boiler; PID Controller.

LISTA DE FIGURAS

Figura 1 – Esquema de Ligação de um Boiler para aquecimento solar da água.	12
Figura 2 – Densidade da água pura em função da temperatura.	13
Figura 3 – Resistência dos principais RTDs em função da temperatura.	14
Figura 4 – Comparação do modelo de Callendar-Van Dusen e da aproximação de primeiro grau.	15
Figura 5 – Diagrama de blocos do controlador PID.	16
Figura 6 – Esquema do Hardware desenvolvido.	19
Figura 7 – Esquemático do circuito para leitura dos Pt100.	20
Figura 8 – Esquemático do circuito de <i>Zerocrossing</i>	22
Figura 9 – Sinais do circuito de <i>Zerocrossing</i>	23
Figura 10 – Esquemático do circuito <i>dimmer</i>	24
Figura 11 – Sinais aplicados na resistência com diferentes potências desejadas.	25
Figura 12 – Tensão RMS gerada pelo <i>dimmer</i> em função do ângulo de atraso.	26
Figura 13 – Aproximação da tensão RMS gerada pelo <i>dimmer</i> em função do ângulo de atraso.	27
Figura 14 – Processamento do sinal de controle para aplicação na planta.	27
Figura 15 – Protótipo do sistema desenvolvido.	28
Figura 16 – Diagrama de blocos do sistema de controle desenvolvido.	29
Figura 17 – Resposta ao degrau do sistema em malha fechada.	30
Figura 18 – Campo do <i>WebServer</i> relacionado a configurações gerais.	32
Figura 19 – Campo do <i>WebServer</i> relacionado a calibração dos sensores.	33
Figura 20 – Campo do <i>WebServer</i> relacionado a calibração dos sensores preenchido.	34
Figura 21 – <i>Dashboard</i> desenvolvido para o sistema.	35
Figura 22 – Saída do sistema com ruído.	37
Figura 23 – Saída do sistema em malha fechada.	37
Figura 24 – Sinal de controle do sistema.	38
Figura 25 – Estrutura desenvolvida para testar o sistema.	39
Figura 26 – Saída do sistema durante o algoritmo de MQR.	40
Figura 27 – Saída do sistema no primeiro teste.	41
Figura 28 – Sinal de controle do sistema no primeiro teste.	41
Figura 29 – Saída do sistema no segundo teste.	42
Figura 30 – Sinal de controle do sistema no segundo teste.	42
Figura 31 – <i>Dashboard</i> do usuário mostrando os dados de aquecimento.	43
Figura 32 – Leitura, Saída em malha fechada e Sinal de controle para simulação com muito ruído na leitura ($\pm 40\%$ do valor de regime permanente).	44
Figura 33 – Leitura, Saída em malha fechada e Sinal de controle para simulação instável.	45

SUMÁRIO

1	INTRODUÇÃO	9
1.1	SISTEMAS DE AQUECIMENTO DE ÁGUA POR ACUMULAÇÃO	9
1.2	OBJETIVOS	9
1.2.1	Objetivo Geral	10
1.2.2	Objetivos Específicos	10
1.3	ESTRUTURA DO TRABALHO	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	AQUECIMENTO DE ÁGUA POR ACUMULAÇÃO	11
2.1.1	Estrutura Física	11
2.1.2	Princípio de aquecimento	12
2.1.3	Atuador de aquecimento	13
2.2	TRANSDUTOR DE TEMPERATURA	13
2.2.1	Princípio de Funcionamento	13
2.2.2	Pt100	14
2.3	FUNDAMENTOS DE CONTROLE	15
2.3.1	Controle PID	15
2.3.2	Mínimos Quadrados Recursivos	17
2.4	WEBSERVER	18
2.4.1	WebSocket	18
2.4.2	SPIFFS	18
3	DESENVOLVIMENTO	19
3.1	HARDWARE	19
3.1.1	Leitura do Pt100	20
<i>3.1.1.1</i>	<i>Calibração</i>	<i>21</i>
3.1.2	<i>Dimmer</i> de Potência	22
3.1.3	Protótipo	28
3.2	SISTEMA DE CONTROLE	28
3.3	WEBSERVER	31
3.4	PLATAFORMA IOT	34
4	RESULTADOS	36
4.1	SIMULAÇÃO	36
4.2	TESTE EM BANCADA	38
4.3	ANÁLISE DOS RESULTADOS	43
5	CONCLUSÃO	46
	REFERÊNCIAS	47

1 INTRODUÇÃO

1.1 SISTEMAS DE AQUECIMENTO DE ÁGUA POR ACUMULAÇÃO

Sistemas de aquecimento de água por acumulação (SOLESTROL, 2022b) são uma solução comum no contexto residencial. Esse tipo de sistema permite o aquecimento da água através da radiação solar. Essa água é acumulada em um reservatório térmico, comumente chamado de *boiler*. O sistema é feito de modo a não precisar de um atuador para a circulação de água, e conta com uma resistência elétrica, para os períodos de baixa incidência solar.

Como vantagem de sistemas de aquecimento de água por acumulação, destaca-se a economia de energia. Por outro lado, o aquecimento por acumulação permite que, em dias com alta incidência solar, não seja necessário utilizar energia elétrica da concessionária de energia para aquecer a água. Em contraponto, sistemas que utilizam *boiler* precisam esquentar grandes quantidades de água, uma vez que os reservatórios tem em média 500 L (ELETROSOL, 2022). Parte dessa água pode não ser utilizada, tendo sido gasta energia desnecessariamente para seu aquecimento. Além disso, em grande parte dos sistemas comercializados (SOLESTROL, 2022a), o acionamento da resistência elétrica para o aquecimento secundário precisa ser programado manualmente. Portanto, mudanças abruptas nas condições climáticas podem fazer o sistema não operar da maneira otimizada. Não esquentando a água quando necessário, ou aquecendo-a mesmo sem necessidade.

Para evitar os problemas citados anteriormente, propõe-se projetar e implementar um sistema de automação aplicado a uma solução comercial existente de aquecimento de água por acumulação. A principal funcionalidade do sistema proposto é adquirir os dados de temperatura, processá-los e controlar a resistência elétrica. Especificamente, o objetivo do sistema proposto é evitar que a resistência elétrica seja acionada quando a água já estiver quente, bem como permitir a programação dos horários de aquecimento da água por meio de um aplicativo em um *smartphone*. Além disso, o acionamento da resistência, que no sistema comercial era discreto (ON/OFF) passa a ser contínuo, permitindo a aplicação de um sistema de controle PID (Proporcional Integral Derivativo) para o aquecimento, o que pode resultar em controle mais preciso da temperatura desejada. Como resultado, espera-se que o novo sistema de automação proporcione maior conforto e praticidade para o usuário.

1.2 OBJETIVOS

Nas seções a seguir estão descritos o objetivo geral e os objetivos específicos deste TCC.

1.2.1 Objetivo Geral

Esse trabalho tem como objetivo o desenvolvimento de um sistema para automatizar e facilitar o aquecimento de água em um *boiler* residencial. O sistema terá conexão com a internet e permitirá a configuração de *timers* para o aquecimento através de uma plataforma IoT. O projeto visa controlar a temperatura da água, que também é aquecida pelo sol, de modo a garantir a temperatura correta nos horários programados, e não esquentar quanto não há necessidade, a fim de manter maior eficiência energética.

1.2.2 Objetivos Específicos

Os objetivos específicos desse trabalho estão listados a seguir:

- Criar um circuito para leitura de sensores Pt100 utilizando um ESP32;
- Desenvolver um *dimmer* de potência controlado pelo ESP32;
- Criar um sistema capaz de se conectar a internet e enviar/receber dados de telemetria e atributos;
- Desenvolver um algoritmo para sintonia automática de um controlador PID.

1.3 ESTRUTURA DO TRABALHO

O presente trabalho foi dividido nas seguintes partes:

- **Capítulo 2.** - Fundamentação teórica - Nesse capítulo, são abordados os conhecimentos necessários para a implementação do sistema proposto. São tratados temas como a estrutura física do sistema, as tecnologias empregadas e a fundamentação matemática necessária para o desenvolvimento.
- **Capítulo 3.** - Desenvolvimento - Nesse capítulo, é abordado o processo de desenvolvimento do sistema proposto, explicando detalhadamente o seu funcionamento.
- **Capítulo 4.** - Resultados - Nesse capítulo, os resultados do sistema desenvolvido são mostrados e analisados.
- **Capítulo 5.** - Conclusão - Nesse capítulo é feita uma análise do sistema como um todo, a fim de verificar se os objetivos propostos foram cumpridos.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 AQUECIMENTO DE ÁGUA POR ACUMULAÇÃO

2.1.1 Estrutura Física

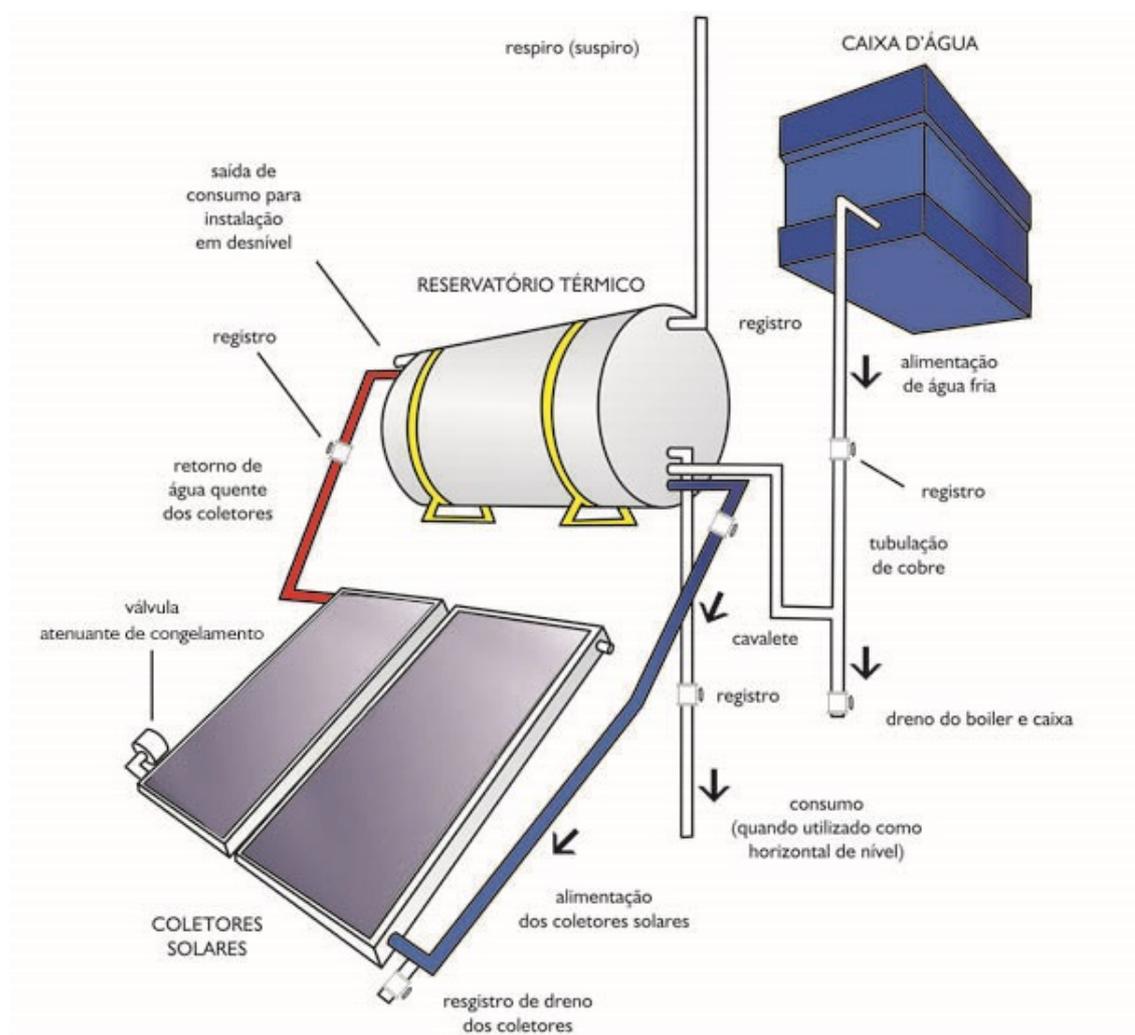
Conforme ilustrado na Figura 1, a estrutura física de um sistema de aquecimento de água por acumulação consiste em 3 componentes principais: o reservatório térmico, a caixa d'água e os coletores solares. Há também tubulações de cobre que interligam essas estruturas.

O reservatório térmico, também chamado de boiler, é conectado à caixa d'água e aos coletores solares. A caixa d'água é a fonte de água fria para o sistema. É importante observar que ela fica, necessariamente, acima do boiler, para que a água seja transferida em apenas um sentido e sem a necessidade de um atuador. Como a água quente tende a ficar acima da fria, é necessário conectar a tubulação proveniente da caixa d'água na parte inferior do boiler, com a presença de um cavalete (técnica chamada de sifonagem) para evitar que a água retorne para a caixa d'água.

Os coletores solares são os responsáveis por coletar a radiação solar e esquentar a água. Eles são compostos por tubulações interligadas com uma camada de vidro na frente. É importante que essa estrutura se encontre abaixo do boiler, para que a diferença de temperatura entre a água quente e a fria faça a água circular no sistema sem a presença de um atuador (processo explicado na Seção 2.1.2).

Ainda há a presença de uma resistência elétrica dentro do boiler. Essa tem a função de esquentar a água quando apenas a radiação solar não é suficiente. No formato original, antes do processo de automatização, seu acionamento é programado manualmente através de um temporizador analógico.

Figura 1 – Esquema de Ligação de um Boiler para aquecimento solar da água.



Fonte: (SOLESTROL, 2022b)

2.1.2 Princípio de aquecimento

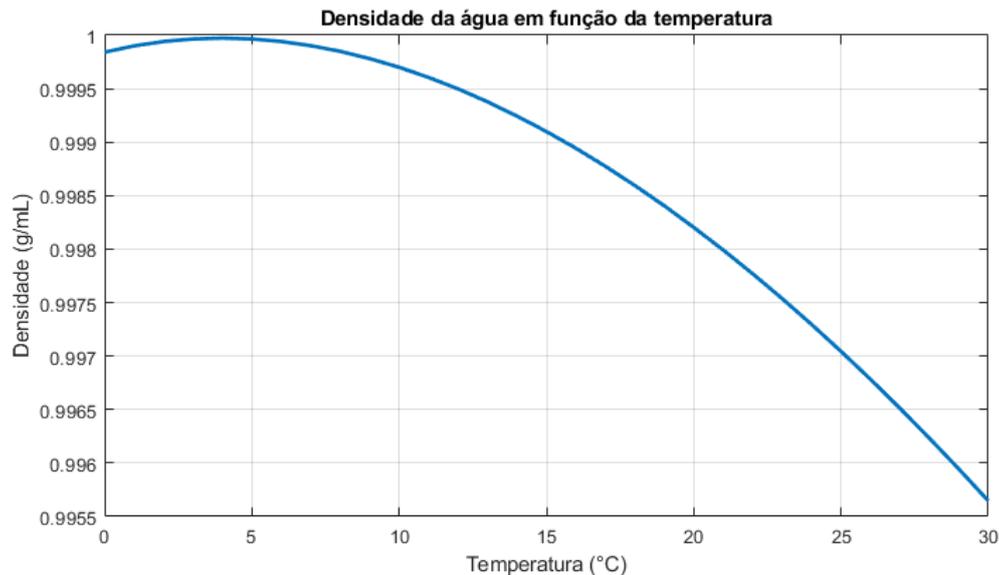
A Figura 2 mostra a densidade da água pura para temperaturas entre 0°C e 30°C. Essas duas propriedades são inversamente proporcionais a partir de 4°C, ou seja, quanto maior a temperatura da água, menor sua densidade. Como a densidade da água fria é maior, ela tende a estar abaixo da água quente. Esse fenômeno é o principal responsável pela circulação de água nos coletores solares.

A água fria é coletada da parte de baixo do boiler e, pela gravidade, enviada para a parte inferior das placas solares. Lá, ela recebe a radiação solar e tem sua temperatura elevada, fazendo sua densidade diminuir e, conseqüentemente, a água se descolar para cima.

Na parte superior das placas, existe uma tubulação que redireciona a água para o boiler. Como esta porção da água está aquecida, sua tendência é subir, voltando para

o boiler. Isso cria uma diferença de pressões, que faz com que mais água fria vá para as placas e seja aquecida. Dado esse funcionamento, não é necessária a presença de um atuador para circular a água, desde que as placas estejam instaladas abaixo do boiler.

Figura 2 – Densidade da água pura em função da temperatura.



Fonte: (WEAST, 1972)

2.1.3 Atuador de aquecimento

Para o caso de não haver luz solar o suficiente para aquecer a água, o sistema conta com uma resistência elétrica que atua no sistema com horários programados. Dada uma resistência R , tem-se que a potência média dissipada para o aquecimento é dada por

$$P = \frac{V_{rms}^2}{R} \quad (1)$$

onde V_{rms} representa a tensão RMS (*root mean square*) da rede elétrica. A resistência presente no sistema trabalhado tem aproximadamente 24Ω . Então, considerando $V_{rms} = 220$ V, (1) resulta em

$$P = \frac{220^2}{24} = \frac{48400}{24} = 2016,6 \text{ W} \approx 2 \text{ kW}. \quad (2)$$

2.2 TRANSDUTOR DE TEMPERATURA

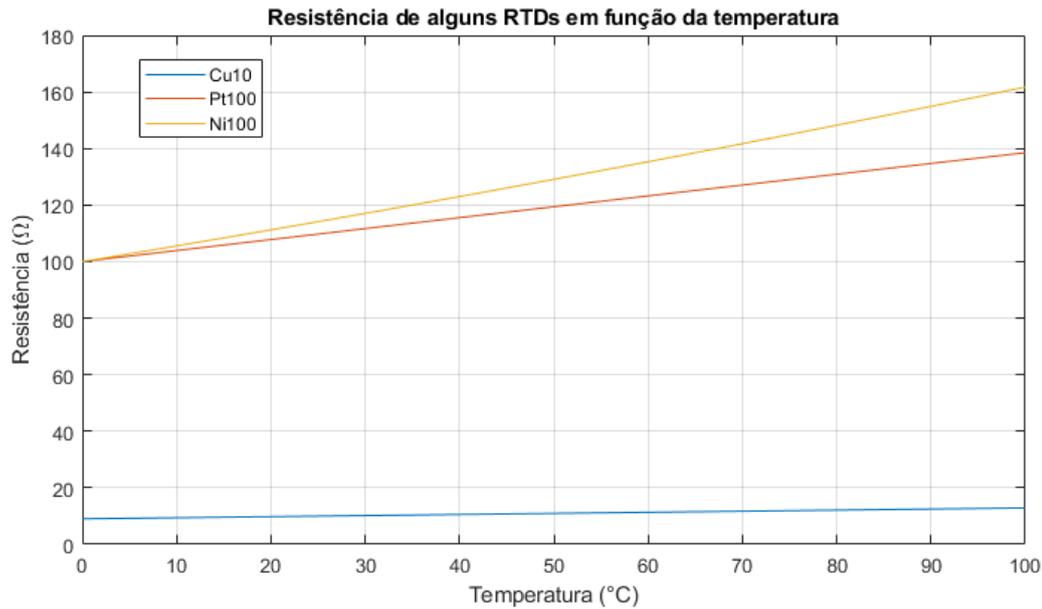
2.2.1 Princípio de Funcionamento

Um dos tipos de sensores mais utilizados para medir a temperatura são as termorresistências, também chamados de RTDs (*Resistance Temperature Detectors*). Esses sensores consistem em uma estrutura metálica cuja resistência elétrica varia com a temperatura.

De modo geral, a resistência elétrica do metal aumenta de acordo com a temperatura, de maneira não linear.

A curva de resistência em função da temperatura varia de acordo com o material do qual o sensor é feito. Os materiais mais comuns dos RTDs são platina, níquel e cobre. A Figura 3 mostra um gráfico da resistência elétrica em função da temperatura para alguns metais.

Figura 3 – Resistência dos principais RTDs em função da temperatura.



Fonte: (TECHNOLOGY, 2021)

É possível descrever matematicamente um comportamento aproximado desse tipo de sensor, utilizando a equação de Callendar-Van Dusen apresentada a seguir:

$$R(T) = \begin{cases} R_0[1 + \alpha_1 T + \alpha_2 T^2 + \alpha_3 (T - 100)T^3] & \text{para } T < 0^\circ\text{C} \\ R_0[1 + \alpha_1 T + \alpha_2 T^2] & \text{para } T \geq 0^\circ\text{C} \end{cases} \quad (3)$$

onde $R(T)$ representa a resistência elétrica do sensor na temperatura T , enquanto R_0 é a resistência medida a 0°C . α_1 , α_2 e α_3 são coeficientes que dependem do material.

2.2.2 Pt100

Um dos RTDs mais utilizados em sistemas de controle e monitoramento é o Pt100. Seu nome deriva do material do qual ele é feito (Platina - Pt) e de sua resistência a 0°C (100Ω). Os seus coeficientes α_1 , α_2 e α_3 em (3) são dados por:

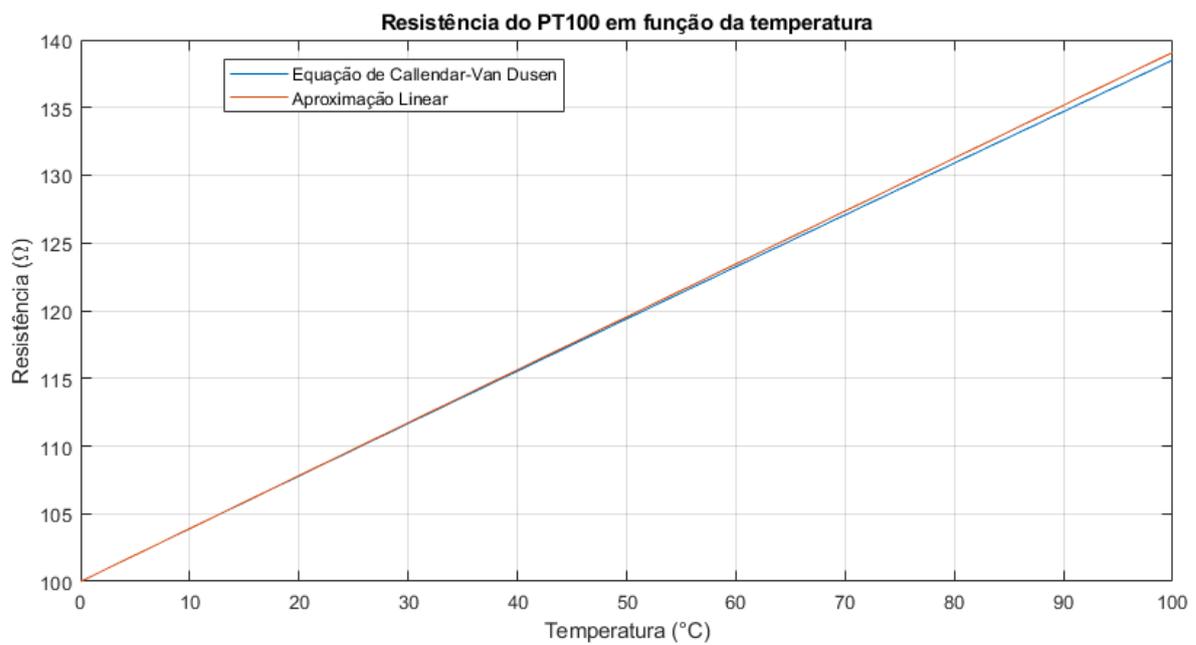
$$\begin{aligned} \alpha_1 &= 3,9083 \cdot 10^{-3} \\ \alpha_2 &= -5,775 \cdot 10^{-7} \\ \alpha_3 &= -4,183 \cdot 10^{-12}. \end{aligned} \quad (4)$$

Uma das principais vantagens do Pt100 é o seu comportamento quase linear. A Figura 4 mostra a resistência elétrica em função da temperatura para o Pt100, na faixa entre 0°C e 100°C. São mostradas duas curvas. A primeira, em azul, utiliza a equação de Callendar-Van Dusen completa, considerando o termo quadrático. A segunda, em laranja, é uma aproximação linear, considerando apenas o termo de primeira ordem. É notável a semelhança entre as duas curvas nessa faixa de temperatura. Portanto, é possível dizer que, para a medição da temperatura de água, a aproximação linear, dada por

$$R(T) = R_0 [1 + 3,9083 \cdot 10^{-3} \cdot T] = 100 + 0,39083 \cdot T \quad (5)$$

é adequada (JOVANOVIĆ; DENIĆ, 2021).

Figura 4 – Comparação do modelo de Callendar-Van Dusen e da aproximação de primeiro grau.



Fonte: Autor

2.3 FUNDAMENTOS DE CONTROLE

2.3.1 Controle PID

O controle PID é uma das técnicas de controle mais utilizadas na indústria. Esse método utiliza três ganhos, chamados de ganho proporcional (K_p), ganho integral (K_i) e ganho derivativo (K_d).

A Figura 5 mostra o diagrama de blocos do controlador PID. Tendo uma vez estabelecido uma referência $r(t)$, mede-se a saída do sistema $y(t)$ e calcula-se o erro $e(t)$

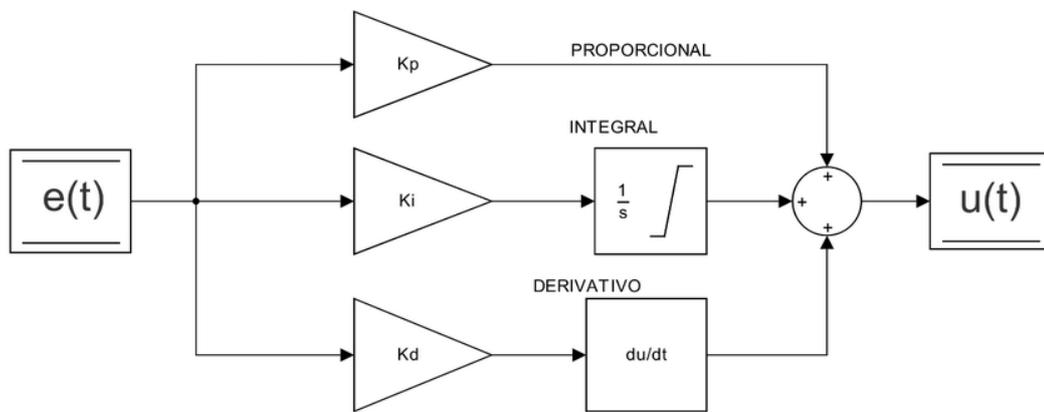
em relação a essa referência, ou seja, $e(t) = r(t) - y(t)$. Com base nos termos proporcional, integral e derivativo do erro, o sinal de controle $u(t)$ é calculado como

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{d}{dt} e(t). \quad (6)$$

No domínio da frequência complexa s , a equivalência de (6) é dada por

$$U(s) = K_p E(s) + K_i \frac{E(s)}{s} + K_d E(s) s. \quad (7)$$

Figura 5 – Diagrama de blocos do controlador PID.



Fonte: (PUHLMANN, 2014)

Para viabilizar a implementação do sistema de controle em um microcontrolador, é necessário obter $u(t)$ no domínio discreto. A abordagem utilizada neste trabalho consiste na aproximação de Euler (REIS SOARES, 1996) dada por

$$s = \frac{z - 1}{T_s} \quad (8)$$

onde T_s denota o período de amostragem e z representa a frequência complexa do sinal discreto. Então, substituindo (8) em (7), obtém-se

$$U(z) = K_p E(z) + K_i \frac{E(z) T_s}{z - 1} + K_d \frac{(z - 1)E(z)}{T_s}. \quad (9)$$

a qual pode ser reescrita como

$$\frac{U(z)}{E(z)} = \frac{\frac{K_d}{T_s} z^2 + (K_p + \frac{2K_d}{T_s}) z + (K_p + \frac{K_d}{T_s} + K_i T_s)}{z - 1}. \quad (10)$$

A função de transferência representada por (10) é imprópria, caracterizando um sistema não causal que não pode ser implementado no microcontrolador. Isso se deve do fato da derivada ser calculada utilizando o valor atual e futuro da saída, o que é impossível

realizar em um sistema que opere em tempo real. Para corrigir isso, pode-se dividir toda função por z , atrasando-a em uma amostra. Isso torna possível sua implementação em tempo real e não altera o princípio de funcionamento do controlador. Portanto, o controlador é descrito pela função de transferência a seguir:

$$\frac{U(z)}{E(z)} = \frac{\frac{K_d}{T_s} z^2 + (K_p + \frac{2K_d}{T_s}) z + (K_p + \frac{K_d}{T_s} + K_i T_s)}{z^2 - z}. \quad (11)$$

2.3.2 Mínimos Quadrados Recursivos

Para o correto cálculo dos ganhos PID, é necessário ter um modelo que descreva o comportamento dinâmico da planta. Um dos métodos mais eficazes para obter esse modelo é chamado de Mínimos Quadrados Recursivos (MQR). Esse método recebe amostras do sinal de entrada $x[n]$ e amostras do sinal de saída $y[n]$ da planta. A cada iteração, os parâmetros do modelo são atualizados.

Inicialmente, define-se um modelo discreto para representar o sistema. Por exemplo, se a escolha for identificar um sistema de segunda ordem, o modelo é dado por:

$$H(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{Y(z)}{X(z)}. \quad (12)$$

No domínio da sequência, (12) pode ser escrito como:

$$y[n] = b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2].$$

Reescrevendo no formato vetorial, obtém-se

$$y[n] = \mathbf{u}^T[n] \boldsymbol{\theta}, \quad (13)$$

onde

$$\mathbf{u}[n] = \begin{bmatrix} x[n-1] \\ x[n-2] \\ -y[n-1] \\ -y[n-2] \end{bmatrix} \quad (14)$$

e

$$\boldsymbol{\theta} = \begin{bmatrix} b_1 \\ b_2 \\ a_1 \\ a_2 \end{bmatrix}. \quad (15)$$

De modo geral, o objetivo do estimador é, a cada instante, processar cada par $x[n], y[n]$ para obter uma nova estimativa do vetor de parâmetros (ou seja, obter $\hat{\boldsymbol{\theta}}$). Sendo o vetor de parâmetros estimados representado por

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{a}_1 \\ \hat{a}_2 \end{bmatrix}. \quad (16)$$

Para tal, a cada iteração n , primeiramente, atualiza-se o vetor $\mathbf{u}[n]$ com as amostras de entrada e saída. Na sequência, computam-se as seguintes equações:

$$\begin{aligned}\mathbf{h}[n] &= \frac{\mathbf{P}[n-1]\mathbf{u}[n]}{1 + \mathbf{u}^T[n]\mathbf{P}[n-1]\mathbf{u}[n]} \\ \hat{\boldsymbol{\theta}}[n] &= \hat{\boldsymbol{\theta}}[n-1] + \mathbf{h}[n](y[n] - \mathbf{u}^T[n]\hat{\boldsymbol{\theta}}[n-1]) \\ \mathbf{P}[n] &= (\mathbf{I} - \mathbf{h}[n]\mathbf{u}^T[n])\mathbf{P}[n-1]\end{aligned}$$

onde, no caso de um sistema de segunda ordem, $\mathbf{P}[n]$ é uma matriz quadrada com dimensão 4×4 e \mathbf{I} representa a matriz identidade, também 4×4 . Vale destacar que para a primeira iteração, deve-se escolher valores iniciais para os vetores $\hat{\boldsymbol{\theta}}[n]$ e $\mathbf{P}[n]$. Usualmente, escolhe-se $\hat{\boldsymbol{\theta}}[0] = \mathbf{0}$ e $\mathbf{P}[0] = \rho\mathbf{I}$, sendo ρ um escalar positivo escolhido pelo projetista.

2.4 WEBSERVER

O ESP32 pode ser utilizado como um ponto de acesso. Nessa configuração, ele gera a própria rede WiFi, cujo nome e senha podem ser configurados via *software*. Ao ser criado o ponto de acesso, um endereço IP é gerado, sendo esse por padrão 192.168.4.1.

Quando um dispositivo se conecta na rede gerada pelo ESP32 e acessa o IP mencionado anteriormente, ele é direcionado para uma página HTML. Isso estabelece uma comunicação entre o ESP32 e o dispositivo conectado (cliente), fazendo com que o ESP32 atue como um *WebServer*.

2.4.1 WebSocket

O WebSocket (FETTE I. E MELNIKOV, 2011) é um protocolo que utiliza uma conexão persistente entre cliente e servidor, permitindo comunicação bidirecional através do protocolo TCP. Isso possibilita que tanto o cliente quanto o servidor enviem dados um para o outro a qualquer momento.

O cliente estabelece uma conexão *WebSocket* com o servidor por meio de um processo conhecido como *handshake WebSocket*. O *handshake* começa com uma solicitação/resposta HTTP, permitindo que os servidores lidem tanto com conexões HTTP quanto com conexões *WebSocket* na mesma porta. Uma vez estabelecida a conexão, o cliente e o servidor podem enviar dados *WebSocket* em modo *full duplex* (transmissão de dados em duas direções simultaneamente).

2.4.2 SPIFFS

O SPIFFS (*Serial Peripheral Interface Flash File System*) é um sistema de arquivos leve que permite armazenar e acessar arquivos na memória flash do ESP32. Com isso, é possível armazenar arquivos HTML, CSS e JavaScript dentro da memória do ESP, e enviá-los ao cliente na hora que conectarem ao *WebServer*.

3 DESENVOLVIMENTO

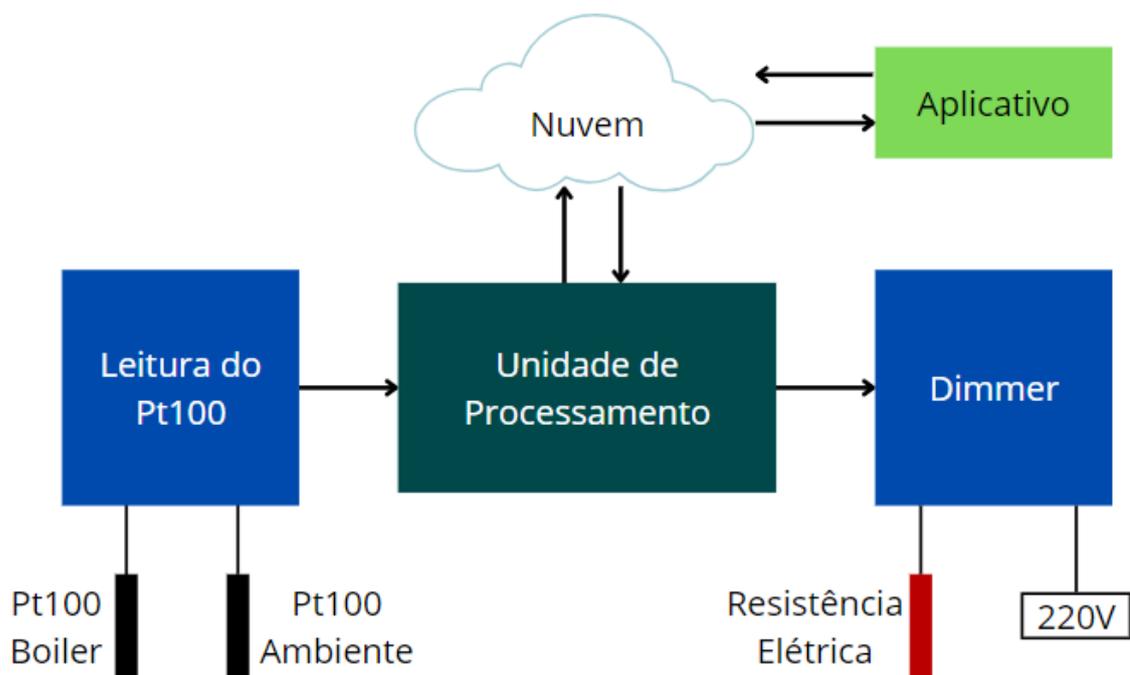
3.1 HARDWARE

O sistema eletrônico é formado por 3 circuitos principais: a unidade de processamento, o circuito de leitura do Pt100 e o *dimmer* de potência. A Figura 6 mostra um diagrama de blocos simplificado do sistema desenvolvido. As setas indicam o caminho dos dados.

A unidade de processamento, composta por um microcontrolador ESP32 (ESPRESSIF . . . , 2024) é a responsável analisar os dados adquiridos pelos sensores, realizar os cálculos de controle, estimar a planta através do algoritmo MQR, controlar o sinal do *dimmer*, processar as requisições do *WebServer* e enviar os dados coletados para a nuvem.

O circuito de leitura do Pt100 é conectado a dois sensores, um no boiler, monitorando a temperatura da água, e o outro medindo a temperatura ambiente. Uma vez tendo o sinal de controle calculado, comuta-se na resistência elétrica através do *dimmer*, que modifica a tensão 220V da rede.

Figura 6 – Esquema do Hardware desenvolvido.



Fonte: Autor

Há uma troca de informações periódica entre a unidade de processamento e a nuvem. O ESP32 envia os dados de temperatura e potência instantânea para a nuvem, que podem ser mostrados ao usuário final utilizando a plataforma IoT. Nessa plataforma,

o cliente também pode programar horários para a água ser aquecida. Esse dado é enviado para ESP32 e armazenado, para garantir o correto funcionamento mesmo *offline*.

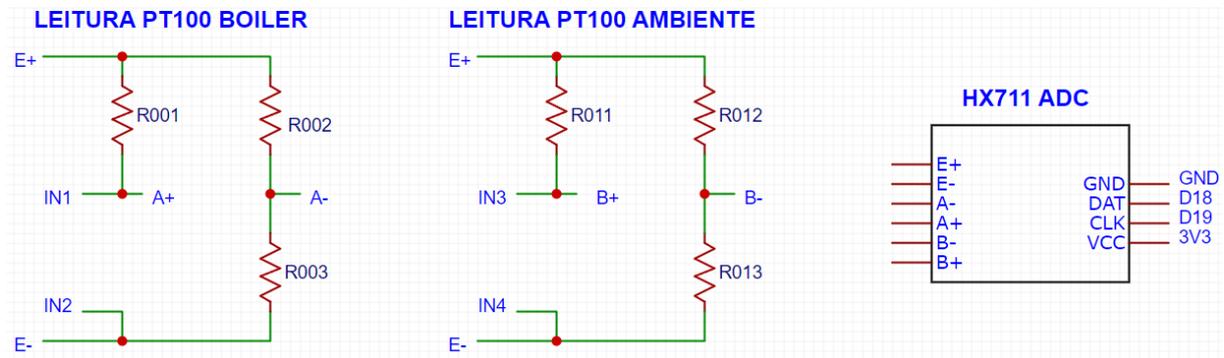
3.1.1 Leitura do Pt100

A leitura dos sensores Pt100 é feita utilizando pontes de Wheatstone (SANTOS; SOUSA CHAVES, 2022). A Figura 7 mostra um esquemático do circuito desenvolvido para a leitura. A fim de obter uma melhor exatidão, foi utilizado um conversor analógico/digital HX711 (AVIA... , 2021), com resolução de 24 bits.

O HX711 proporciona uma tensão de 3,3V no terminal E+ e 0V no terminal E- da ponte de Wheatstone. Isso proporciona uma tensão nos terminais A+, A-, B+ e B- da ponte. Conforme a resistência do Pt100 do boiler varia, a tensão em A+ também varia. O mesmo ocorre com o Pt100 referente à temperatura ambiente conectado no terminal B+. Já as tensões em A- e B- são constantes.

Os resistores R001, R002, R011 e R012 tem resistência de 10 k Ω , enquanto que R003 e R013 são de 100 Ω . Isso faz com que a ponte esteja equilibrada quando o Pt100 atingir 100 Ω , o que ocorre a 0°C.

Figura 7 – Esquemático do circuito para leitura dos Pt100.



Fonte: Autor

Considerando o sensor Pt100 posicionado no boiler, é possível calcular sua temperatura com base na diferença entre as tensões em A+ e A-. Tem-se que a tensão em A- é constante e dada por:

$$V_{A-} = \frac{3,3 \cdot 100}{10000 + 100} = 32,67\text{mV}, \quad (17)$$

e a tensão em A+, que varia em função da resistência do sensor (R_{pt}), é dada por:

$$V_{A+}(R_{pt}) = \frac{3,3 R_{pt}}{10000 + R_{pt}}. \quad (18)$$

A resistência do Pt100 R_{pt} por ser aproximada linearmente por (5). Substituindo essa aproximação em (18), obtém-se a tensão em A+ em função da temperatura, resultando

em

$$V_{A+}(T) = \frac{3,3 (100 + 0,39083 T)}{10100 + 0,39083 T}. \quad (19)$$

O valor retornado pelo HX711 representa a diferença entre o V_{A+} e V_{A-} , denotado aqui por ΔV . Assim, tem-se que

$$\Delta V = V_{A+} - V_{A-} = \frac{3,3 (100 + 0,39083 T)}{10100 + 0,39083 T} - \frac{330}{10100}. \quad (20)$$

Manipulando (20), pode-se obter o valor da temperatura em função de ΔV , isto é,

$$T = \frac{10100 \Delta V}{0,39083 (3,267 - \Delta V)}. \quad (21)$$

Como o HX711 tem uma resolução de 24 bits, o ESP32 recebe um valor inteiro entre -2^{23} e $2^{23} - 1$ na leitura. Esse inteiro representa o valor de tensão já convertido para digital. Porém, o HX711 tem um ganho na leitura variável. Esse ganho pode ser de 32, 64 ou 128 vezes. Para o caso de 32 vezes, apenas o canal B se aplica. Já para 64 e 128 vezes, apenas o canal A. Utilizando o ganho em 32, notou-se experimentalmente que, a cada 1V aplicado ao canal B do HX711, o valor digital recebido pelo ESP32 subia em 1852100. Portanto, pode-se calcular o ΔV utilizando

$$\Delta V = \frac{D}{1852100}. \quad (22)$$

3.1.1.1 Calibração

O método descrito anteriormente se baseia nos valores nominais de resistência e tensão. Porém, na prática, esses valores podem variar e causar erros significativos na leitura. Um teste foi feito em condições de aproximadamente 20°C. A leitura do Pt100 foi de 13°C. Esse erro se deve, principalmente, aos valores reais dos resistores, que variam em até 5% dos nominais.

Para corrigir esse problema, ao invés de utilizar o cálculo descrito anteriormente, pode-se usar um método de calibração com dois resistores auxiliares, que representam a resistência do Pt100 em duas temperaturas conhecidas. Por exemplo, pode-se usar resistores de 100Ω e 124Ω, que representam o Pt100 a, respectivamente, 0°C e 62°C. Foram utilizados resistores de precisão, com erro de $\pm 1\%$. Para a calibração, deve-se conectar os resistores no lugar do Pt100 e registrar o valor lido pelo HX711. São salvos quatro valores: IN_1 , IN_2 , OUT_1 e OUT_2 . Os dois últimos são as temperaturas correspondentes aos resistores. No caso citado anteriormente, teria-se $OUT_1 = 0$ e $OUT_2 = 62$. Já IN_1 e IN_2 são os valores lidos pelo HX711, sendo inteiros que representam os bits dos valores digitais. Se, ao conectar um resistor de 100Ω no lugar do Pt100, o valor lido pelo HX711 for 1506, tem-se que $IN_1 = 1506$. Uma vez tendo esses valores, pode-se calcular a temperatura em função da leitura do HX711, resultando em

$$T(D) = \frac{(D - IN_1) (OUT_2 - OUT_1)}{IN_2 - IN_1} + OUT_1 \quad (23)$$

Esse método garante um erro menor na leitura do Pt100, pois já considera os valores reais dos resistores presentes na ponte de Wheatstone.

3.1.2 *Dimmer* de Potência

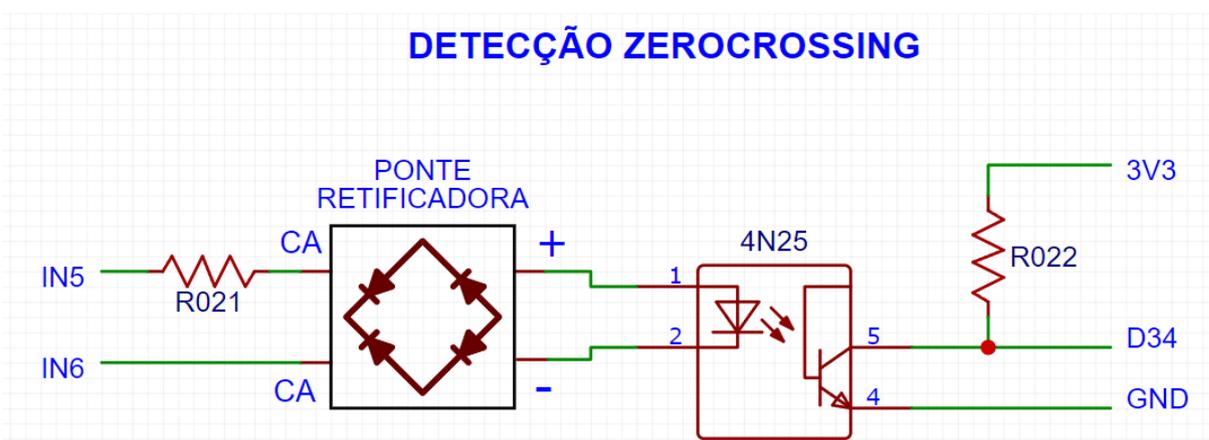
Antes do controlador ser aplicado na prática, é necessário conseguir alterar a tensão RMS aplicada na resistência e, assim, alterar a potência enviada ao sistema. Para isso, um circuito *dimmer* de potência foi construído.

O sistema consiste de duas partes. A detecção de cruzamento por zero (*zerocrossing*) e o *dimmer* em si. A primeira tem a função de informar ao ESP32 quando a senoide de tensão passa por zero, ou seja, quando uma nova meia onda se inicia. A Figura 8 mostra o esquemático do circuito de detecção *zerocrossing*.

A tensão da rede é aplicada em IN5 e IN6, e passa por uma ponte retificadora. Os terminais positivos e negativos do sinal retificado servem de entrada para um optoacoplador 4N25 (VISHAY..., 2010). Enquanto houver uma diferença de tensão entre os dois terminais, o LED interno do 4N25 irá fechar o transistor no outro lado do circuito. Nesse lado, há uma resistência de 10k Ω que conecta o 3V3 e o pino D34 do ESP32. Ao acender o LED do 4N25, o fechamento do transistor gera um curto circuito entre D34 e o GND, fazendo o pino ir a nível lógico baixo.

Por outro lado, no momento que a tensão da rede passar por 0V, o LED do 4N25 irá apagar e abrir o transistor, fazendo com que a corrente no resistor R022 seja nula e o pino D34 vá para nível lógico alto.

Figura 8 – Esquemático do circuito de *Zerocrossing*.

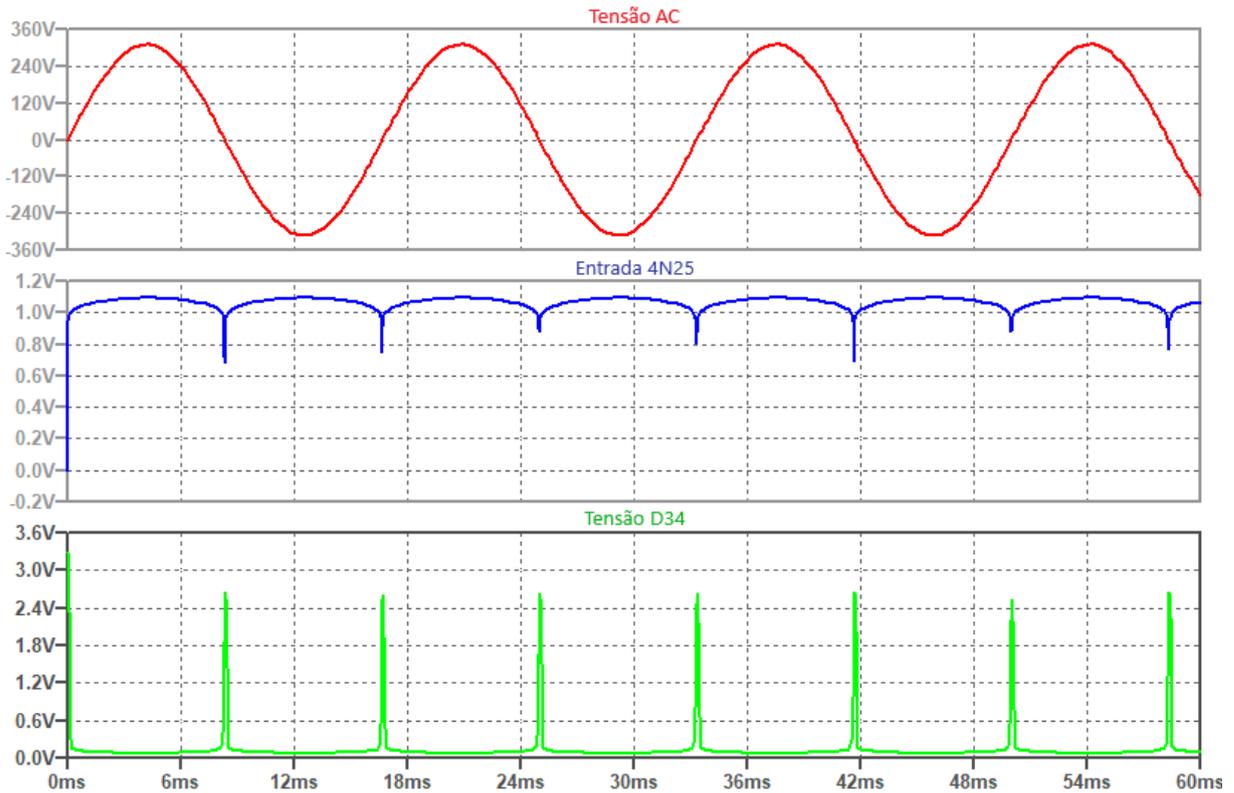


Fonte: Autor

A Figura 9 mostra alguns sinais do circuito de *zerocrossing*. Primeiro, em vermelho, tem-se a tensão da rede, configurada com valor RMS de 220V e frequência de 60Hz. Abaixo, em azul, tem-se o sinal de entrada do 4N25. Nota-se que a tensão permanece entre 1V e

1,1V, exceto quando a tensão da rede passa por 0V. É nesse momento que o LED interno desliga. Por último, em verde, é apresentado o sinal de tensão do pino D34. Nota-se que, como explicado, ele sobe para nível lógico alto quando a tensão AC passa pelo 0V.

Figura 9 – Sinais do circuito de *Zerocrossing*.



Fonte: Autor

O ESP32 está configurado com uma interrupção nesse pino. Portanto, sempre que seu nível lógico subir, uma rotina de tratamento de interrupção é chamada. Essa rotina manipula a segunda parte do sistema, que é o *dimmer* em si.

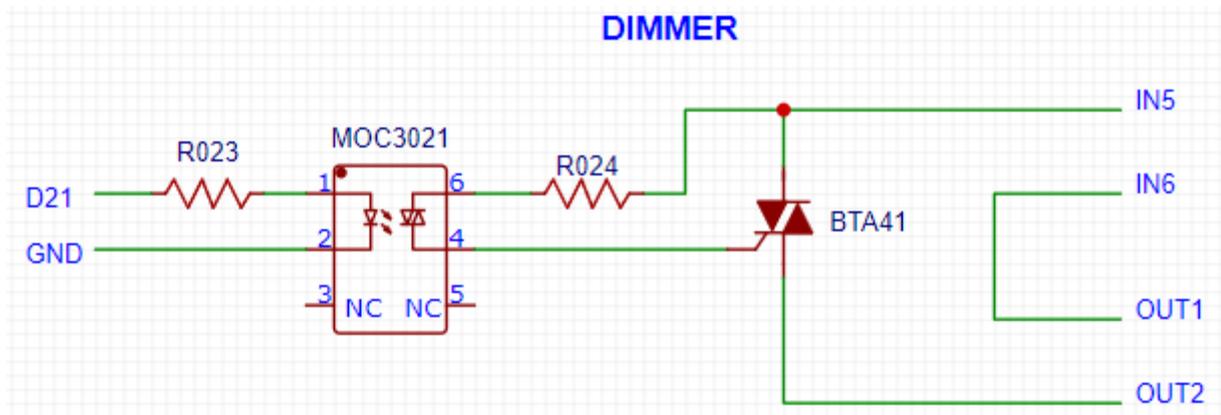
O *dimmer* controla a tensão aplicada na resistência. A Figura 10 mostra o seu esquemático. A tensão da rede entra por IN5 e IN6 e é direcionada à resistência elétrica por OUT1 e OUT2. A passagem de corrente para a resistência é controlada por um TRIAC BTA41 (ST..., 2023).

O TRIAC libera ou bloqueia a passagem de corrente dependendo da tensão no *gate* (V_G) e da corrente entre os terminais (I_T). Caso houver diferença de potencial entre os terminais, a corrente passa a ser conduzida quando V_G for maior que certo limiar. Uma vez conduzindo, V_G pode voltar a 0V sem que I_T seja bloqueada. O componente parará de conduzir apenas quando essa corrente chegar a 0A. Uma vez nesse estado, mesmo com diferença de potencial entre os terminais, o componente não irá mais conduzir corrente.

Para aplicar a tensão no *gate* do TRIAC, a tensão de IN5 é levada até um optoaco-

plador MOC3021 (FAIRCHILD... , 2000), que por sua vez libera a passagem de corrente apenas quando o pino D21 do ESP32 estiver em nível lógico alto.

Figura 10 – Esquemático do circuito *dimmer*.

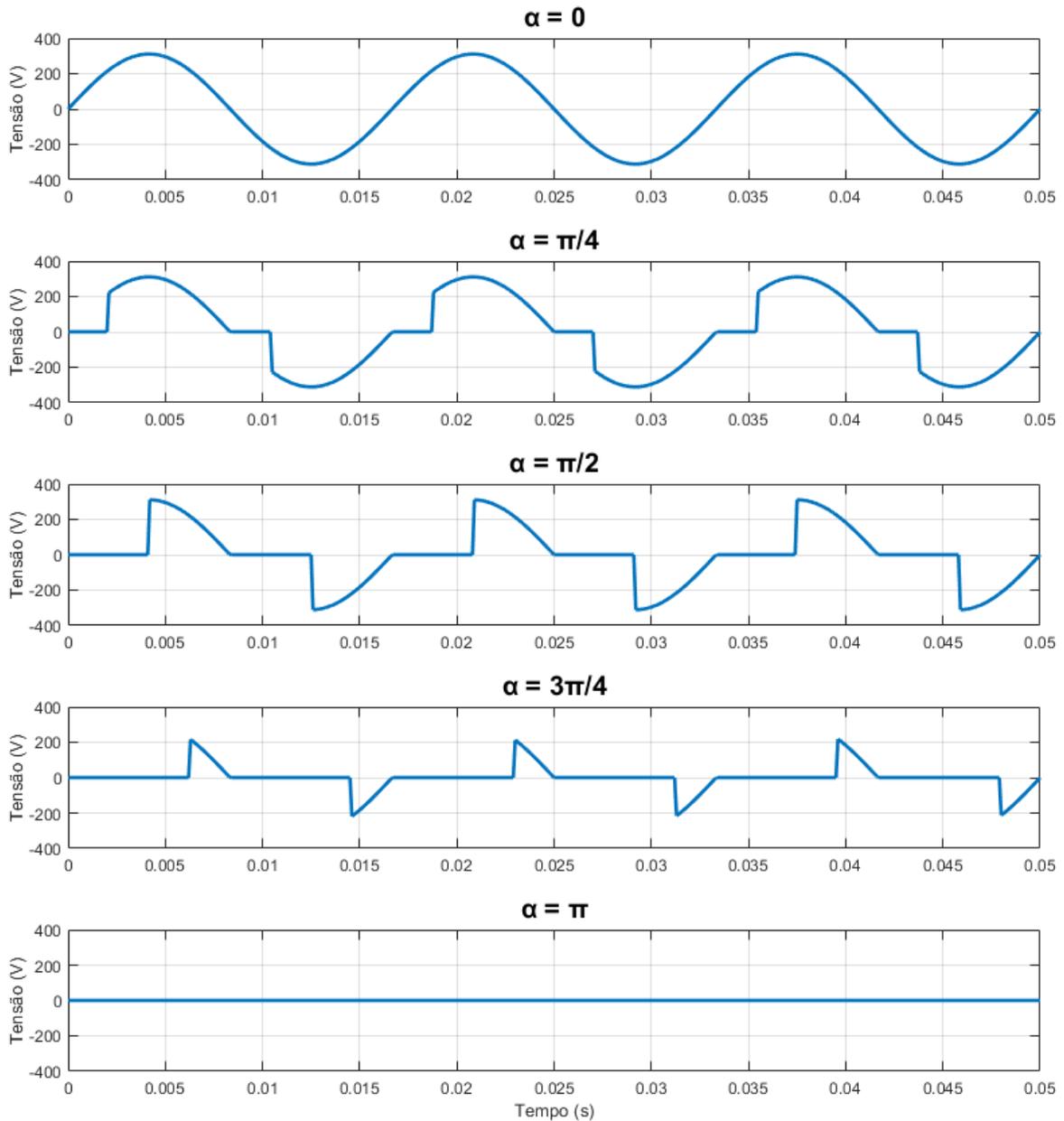


Fonte: Autor

Quando o circuito de *zerocrossing* identificar o início de uma nova onda, a rotina de interrupção é chamada. Essa rotina espera um determinado tempo e envia um pulso no pino D21, que libera a corrente na resistência utilizada no aquecimento. O tempo esperado depende da potência que se deseja mandar ao sistema. Esse intervalo representa o tempo que a resistência permanecerá desligada.

A Figura 11 mostra alguns exemplos de sinais enviados pelo *dimmer* para a resistência. A tensão sobe de 0V e começa a seguir a senoide no momento que o pulso é enviado pelo pino D21. Define-se o ângulo de condução $0 \leq \alpha \leq \pi$ como sendo o ângulo da senoide em que é iniciada a condução de corrente. Na Figura 11, tem-se que as ondas tem um ângulo de condução de 0 , $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$ e π , respectivamente.

Figura 11 – Sinais aplicados na resistência com diferentes potências desejadas.

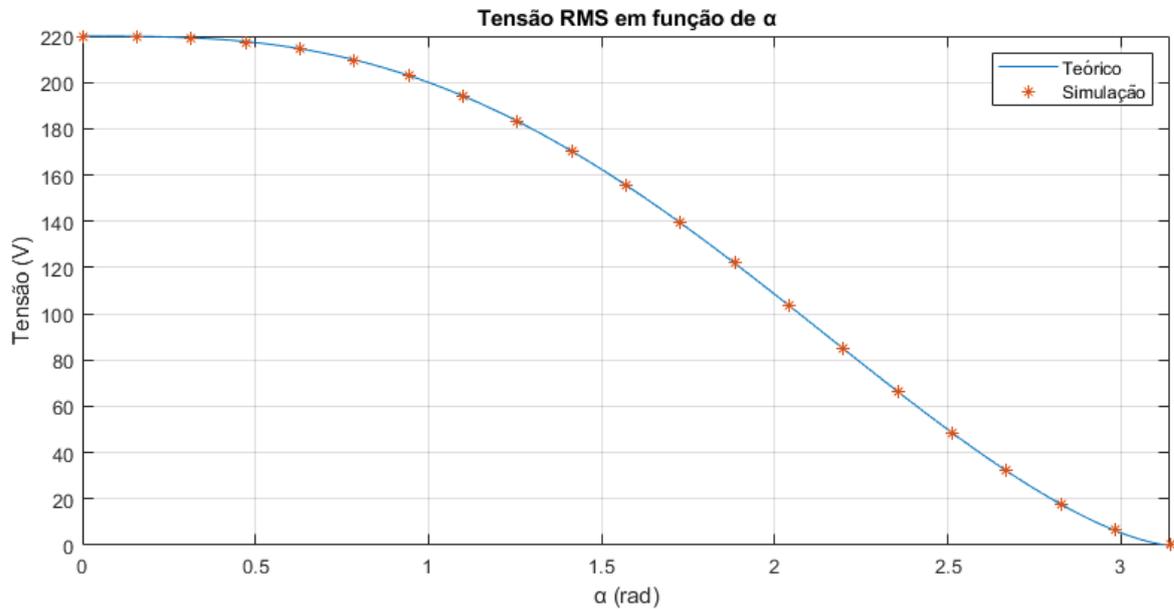


Fonte: Autor

A tensão RMS produzida pelo *dimmer* pode ser calculada utilizando a equação

$$V_{rms}(\alpha) = V_{in_{rms}} \sqrt{1 - \frac{\alpha}{\pi} + \frac{\text{sen}(2\alpha)}{2 + \pi}} \quad (24)$$

sendo $V_{in_{rms}}$ a tensão RMS da entrada, nesse caso 220V, e α o ângulo de condução da senoide, em radianos. A Figura 12 mostra um gráfico da tensão RMS em função de α . A linha contínua representa os valores teóricos e os asteriscos os valores simulados no *software* LTspice.

Figura 12 – Tensão RMS gerada pelo *dimmer* em função do ângulo de atraso.

Fonte: Autor

Pode-se verificar em (24) que α é diretamente proporcional ao tempo esperado para enviar o pulso. Portanto, encontrando uma expressão que determine α em função da tensão RMS desejada, é possível calcular o tempo de espera para alcançar essa dada tensão, e assim aplicar corretamente o sinal do controlador. Entretanto, inverter (24) e encontrar α em função de V_{rms} não é um processo trivial devido ao termo $\text{sen}(2\alpha)$ presente na equação. Porém, esse problema pode ser superado através da aproximação

$$V_{rms}(\alpha) \approx V_{in_{rms}} \cos\left(\frac{\alpha}{2}\right). \quad (25)$$

A Figura 13 mostra a comparação dessa aproximação (laranja) com a curva real (azul).

Com isso, pode-se calcular o valor de α com base no valor de tensão RMS esperado na saída utilizando

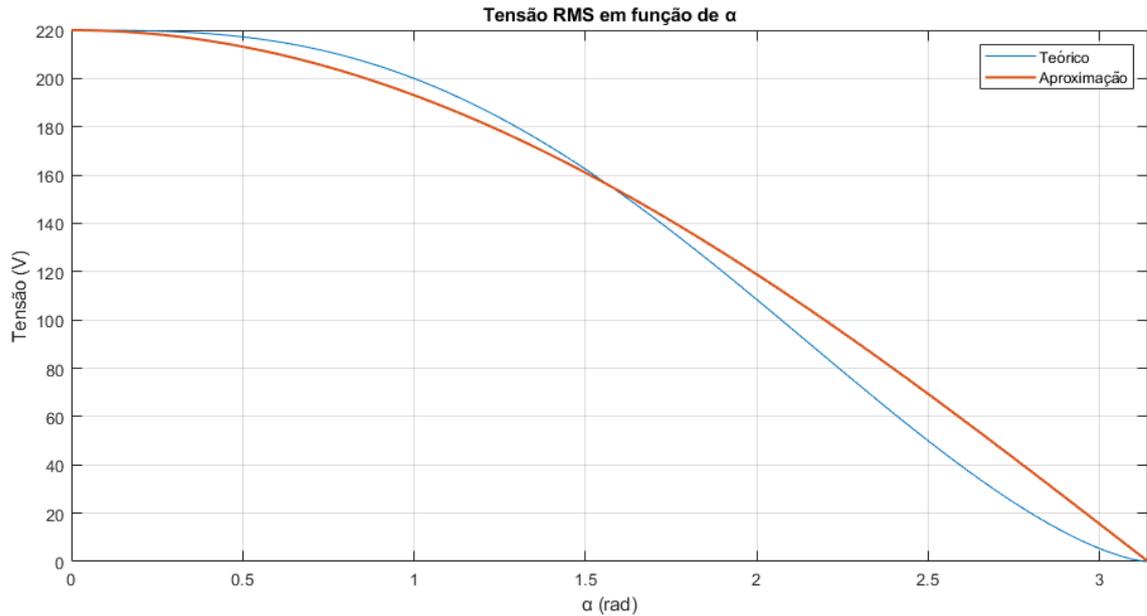
$$\alpha(V_{rms}) = 2 \arccos\left(\frac{V_{rms}}{V_{in_{rms}}}\right). \quad (26)$$

É importante se atentar que o sinal de controle não é um valor de tensão, e sim de potência elétrica. Como mostrado em (2), a resistência possui 24Ω . Portanto, tendo o valor do sinal de controle em Watts, pode-se calcular a tensão RMS para alcançar essa potência utilizando

$$V_{rms} = \sqrt{R P} = \sqrt{24 U}, \quad (27)$$

sendo R a resistência elétrica e U o sinal de controle. Nota-se que tensão e potência não possuem uma relação linear.

Figura 13 – Aproximação da tensão RMS gerada pelo *dimmer* em função do ângulo de atraso.



Fonte: Autor

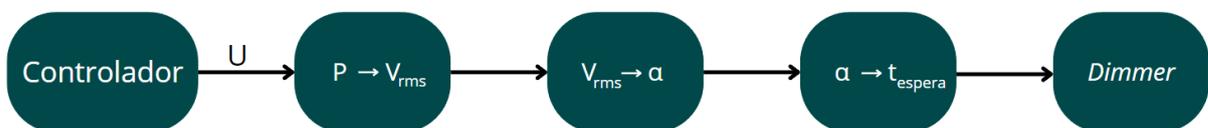
A Figura 14 mostra um esquema das etapas para garantir que o sinal de controle seja aplicado corretamente na planta. Tendo o sinal de controle, calcula-se a tensão RMS desejada. Com essa tensão, o ângulo de condução é calculado e, conseqüentemente, determina-se tempo de espera para o *dimmer* disparar o pulso depois do *zerocrossing*. Como a tensão da rede possui 60Hz, cada meia onda possui período de

$$T = \frac{1}{60 \cdot 2} = 8,333\text{ms}. \quad (28)$$

Portanto, um ângulo de atraso de π radianos significa um tempo de espera de 8,333ms. Calcula-se o tempo de espera do pulso utilizando

$$t_{espera} = \frac{8,333 \cdot \alpha}{\pi}. \quad (29)$$

Figura 14 – Processamento do sinal de controle para aplicação na planta.



Fonte: Autor

3.1.3 Protótipo

A Figura 15 mostra o protótipo desenvolvido em uma PCB perfurada.

Figura 15 – Protótipo do sistema desenvolvido.



Fonte: Autor

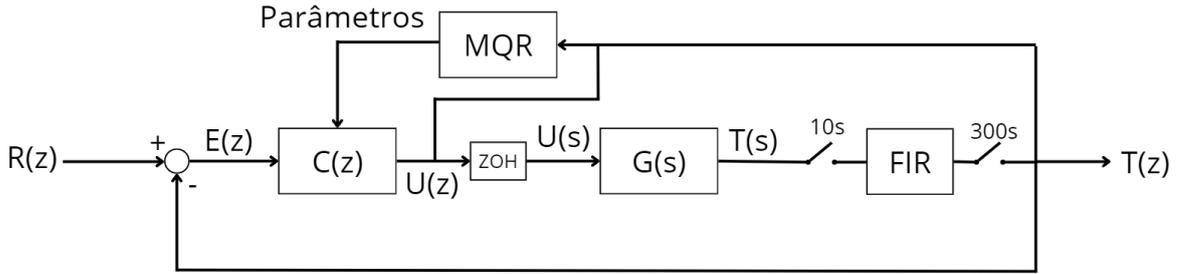
3.2 SISTEMA DE CONTROLE

O sistema de controle consiste em um controlador PID, com os ganhos ajustados automaticamente utilizando o algoritmo de identificação de sistemas MQR. A Figura 16 mostra o diagrama de blocos do sistema de controle. O erro $E(z)$ é calculado utilizando $E(z) = T(z) - R(z)$, sendo $R(z)$ a referência de temperatura e $T(z)$ a temperatura do sistema. O sinal de erro é aplicado no controlador, que calcula o sinal de controle $U(z)$ e aplica na planta. A temperatura do sistema é lida a cada 10s. Essa temperatura passa por um filtro FIR (*Finite Impulse Response*) de média móvel, para diminuir as oscilações da leitura. Esse sinal filtrado é então direcionado para o cálculo do erro (realimentação). Uma nova amostra é disponibilizada na saída do filtro a cada 300s.

Quando o sistema inicia, ainda não existem dados sobre o sistema. Portanto, durante o primeiro aquecimento, é aplicado um degrau com potência máxima na entrada da planta ao sistema. A cada nova iteração, que ocorre a cada 300s, o MQR calcula novos parâmetros da planta e, uma vez calculados, esses valores são utilizados para alterar os parâmetros de

$C(z)$. O MQR atua apenas durante o primeiro aquecimento. Uma vez obtido o modelo da planta e calculado o controlador, a identificação da planta não é mais feita pelo algoritmo. O sistema ativa o controlador 1 hora e meia antes dos horários programados pelo usuário, para que, nesses horários, a água já esteja quente.

Figura 16 – Diagrama de blocos do sistema de controle desenvolvido.



Fonte: Autor

O filtro de média móvel calcula a média das últimas amostras de temperatura lidas. Isso ajuda a diminuir as oscilações causadas pelos erros aleatórios de leitura. Dado um filtro de ordem N , tem-se que a temperatura filtrada T_F segue a equação:

$$T_F[k] = \frac{1}{N} \sum_{i=k-N}^k T[i]. \quad (30)$$

O filtro aplicado ao sistema tem ordem 6. Portanto, é a média dos 6 últimos valores de temperatura lidos.

Considerando que o modelo discreto de primeira ordem da planta obtido pelo método MQR é dado por

$$G(z) = \frac{g_1 z + g_2}{z^2 + d_1 z + d_2} \quad (31)$$

e que a função de transferência discreta do controlador é modelada como

$$C(z) = \frac{a_0 z^2 + a_1 z + a_2}{(z - 1)(z - \lambda)}, \quad (32)$$

o comportamento do sistema em malha fechada segue a equação

$$T(z) = \frac{C(z)G(z)}{1 + C(z)G(z)} = \frac{N_C N_G}{D_C D_G + N_C N_G}. \quad (33)$$

Nessa equação, N é o numerador e D é o denominador da função de transferência. Aplicando (31) e (32) em (33), obtém-se:

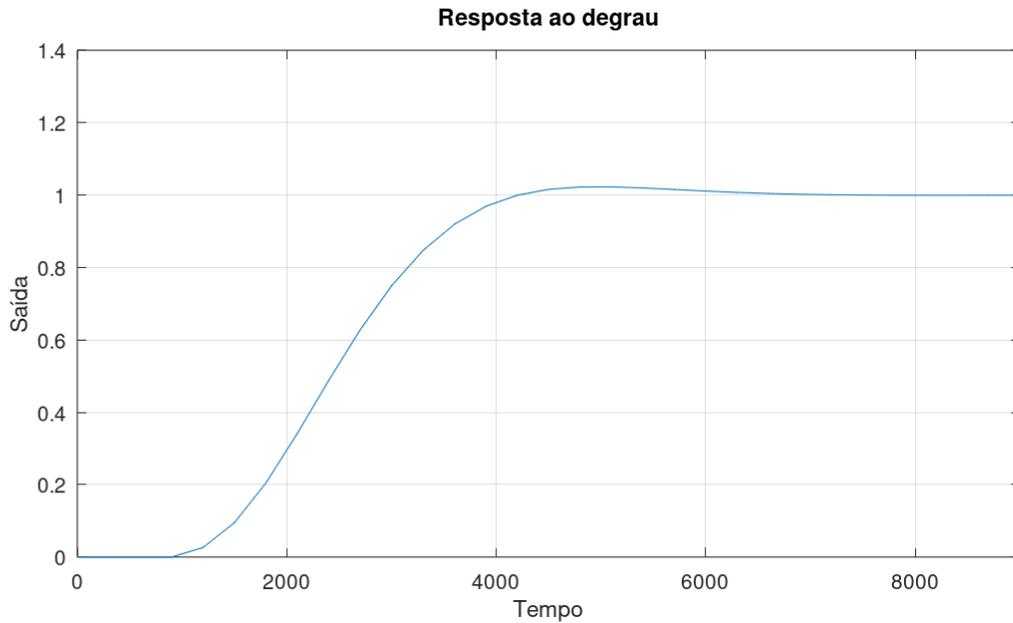
$$T(z) = \frac{(a_0 z^2 + a_1 z + a_2)(g_1 z + g_2)}{(z - 1)(z - \lambda)(z^2 + d_1 z + d_2) + (a_0 z^2 + a_1 z + a_2)(g_1 z + g_2)}. \quad (34)$$

Para calcular os valores dos coeficientes do controlador, é necessário igualar o denominador da função em (34) com um que tenha o comportamento desejado. Desse modo, os polos foram obtidos experimentalmente com objetivo de reduzir o sobressinal e obter um tempo de assentamento próximo de uma hora. Especificamente, foram escolhidos os polos $p_{1,2} = 0,65 + 0,2j$ e $p_{3,4} = 0,65 - 0,2j$, resultando no denominador

$$(z - 0,65 - 0,2j)^2 (z - 0,65 + 0,2j)^2 = z^4 - 2,6z^3 + 2,615z^2 - 1,2025z + 0,21390625. \quad (35)$$

A Figura 17 mostra a resposta ao degrau dessa função de transferência. A partir dessa figura, pode-se verificar que o sistema em malha fechada atinge a temperatura desejada em aproximadamente 4000s (1 hora e 7 minutos aproximadamente) e que o sobressinal não ultrapassou 3% do valor em regime permanente.

Figura 17 – Resposta ao degrau do sistema em malha fechada.



Fonte: Autor

A partir de (34) com (35), obtém-se as seguintes equações para os parâmetros do controlador:

$$\begin{aligned} a_0 = & (d_2 g_1^2 - d_1 g_2^2 - d_2 g_2^2 + g_2^2 h_0 + g_2^2 h_1 + g_1^2 h_3 + g_2^2 \\ & + d_1^2 g_2^2 - d_1 g_1 g_2 - g_1 g_2 h_2 - d_1 d_2 g_1^2 + d_1^2 g_1 g_2 - d_1 g_2^2 h_0 \\ & + d_2 g_1^2 h_0 - d_1 d_2 g_1 g_2 - d_1 g_1 g_2 h_0 + d_2 g_1 g_2 h_0) \cdot \\ & [(g_1 + g_2) (d_2 g_1^2 - d_1 g_1 g_2 + g_2^2)]^{-1} \quad (36) \end{aligned}$$

$$\begin{aligned}
a_1 = & (d_1 g_2^2 - g_1^2 h_3 + g_2^2 h_2 - d_1^2 g_2^2 - d_2^2 g_1^2 + g_1 g_2 h_2 \\
& - g_1 g_2 h_3 + d_1 d_2 g_1^2 + d_1 d_2 g_2^2 - d_1^2 g_1 g_2 - d_2^2 g_1 g_2 \\
& + d_1 g_2^2 h_0 + d_2 g_1^2 h_1 - d_2 g_2^2 h_0 + d_1 g_1^2 h_3 + 2 d_1 d_2 g_1 g_2 \\
& - d_1 g_1 g_2 h_1 - d_2 g_1 g_2 h_0 - d_1 g_1 g_2 h_2 + d_2 g_1 g_2 h_1). \\
& [(g_1 + g_2) (d_2 g_1^2 - d_1 g_1 g_2 + g_2^2)]^{-1} \quad (37)
\end{aligned}$$

$$\begin{aligned}
a_2 = & (d_2 g_2^2 + g_2^2 h_3 + d_2^2 g_1^2 + g_1 g_2 h_3 - d_1 d_2 g_2^2 \\
& + d_2^2 g_1 g_2 + d_2 g_2^2 h_0 - d_1 g_1^2 h_3 + d_2 g_1^2 h_2 \\
& + d_2 g_1^2 h_3 - d_1 d_2 g_1 g_2 - d_2 g_1 g_2 h_1 - d_1 g_1 g_2 h_3). \\
& [(g_1 + g_2) (d_2 g_1^2 - d_1 g_1 g_2 + g_2^2)]^{-1} \quad (38)
\end{aligned}$$

e

$$\begin{aligned}
\lambda = & -(g_2^3 h_0 - d_1 g_2^3 - g_1^3 h_3 + g_2^3 - d_1 g_1 g_2^2 \\
& + d_2 g_1 g_2^2 + d_2 g_1^2 g_2 - g_1 g_2^2 h_1 + g_1^2 g_2 h_2). \\
& [(g_1 + g_2) (d_2 g_1^2 - d_1 g_1 g_2 + g_2^2)]^{-1} \quad (39)
\end{aligned}$$

Esse controlador pode ser aplicado no ESP32 utilizando a equação a diferenças apresentada a seguir:

$$U[n] = (1 + \lambda) U[n - 1] - \lambda U[n - 2] + a_0 E[n] + a_1 E[n - 1] + a_2 E[n - 2]. \quad (40)$$

Para evitar comportamentos indesejados devido aos zeros do sistema em malha fechada, um filtro de referência (OGATA, 2014) pode ser aplicado. Pode-se notar em (33) que os zeros de malha fechada são os mesmos que os da planta e os do controlador. A planta dificilmente terá zeros problemáticos. Portanto, o filtro de referência deve cancelar os zeros do controlador. Assim, considerando (32), a função de transferência do filtro de referência é dada por

$$F_R(z) = \frac{R_F(z)}{R(z)} = \frac{(a_0 + a_1 + a_2)z^2}{a_0 z^2 + a_1 z + a_2} \quad (41)$$

Para aplicar o filtro no microcontrolador, utiliza-se a equação a diferenças a seguir:

$$R_F[n] = \frac{(a_0 + a_1 + a_2)R[n] - a_1 R_F[n - 1] + a_2 R_F[n - 2]}{a_0}. \quad (42)$$

3.3 WEBSERVER

O ESP32 atua como um *WebServer* onde é realizada a configuração inicial do sistema. Ao se conectar na rede sem fio gerada pelo ESP32 e digitar o endereço de IP padrão, 192.168.4.1, em seu navegador, o usuário tem acesso à página de configuração. Ela consiste em três campos, sendo o primeiro relacionado a configurações gerais e os outros dois a calibração dos sensores Pt100.

A Figura 18 mostra a parte de configurações gerais. São três campos a serem preenchidos: Nome da rede Wifi, Senha da rede Wifi e *Token*. O nome e a senha do Wifi referem-se à rede de internet que o ESP32 deverá se conectar para enviar os dados. O *token* consiste na chave de acesso para a plataforma IoT (Seção 3.4). Ao clicar no botão "Salvar", uma mensagem é enviada ao ESP32 com os três campos, que são salvos na sua memória EEPROM (*Electrically-Erasable Programmable Read-Only Memory*).

Figura 18 – Campo do *WebServer* relacionado a configurações gerais.



O formulário é exibido em um fundo azul claro. Ele contém três campos de entrada de texto, cada um com um rótulo em azul escuro à esquerda: "Nome do Wifi", "Senha do Wifi" e "Token". Abaixo dos campos, há um botão cinza com o texto "Salvar" em azul escuro.

Fonte: Autor

A Figura 19 mostra as telas do *WebServer* em que a calibração dos sensores é feita. Quando os dois valores de input são iguais a zero, o valor mostrado no campo em cinza claro é a leitura do HX711. Já quando os valores de *input* estão preenchidos, o campo mostra a temperatura medida pelo sistema.

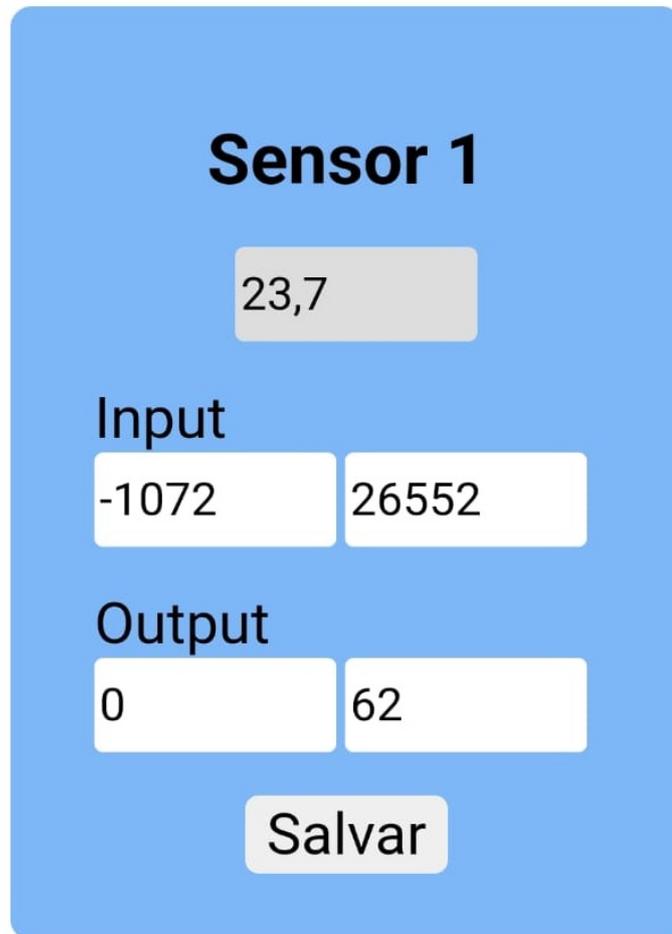
A calibração se baseia no processo descrito anteriormente (Seção 3.1.1.1). São necessárias dois resistores, que representam o Pt100 em duas temperaturas conhecidas. A Figura 20 mostra um exemplo de preenchimento desse campo. Ao colocar um resistor de 100Ω (0°C), o valor lido pelo HX711 foi de -1072 . Já com um resistor de 124Ω (62°C), o

valor lido foi 26552. Nessa figura, o campo em cinza mostra a temperatura medida naquele momento pelo sensor Pt100.

Figura 19 – Campo do *WebServer* relacionado a calibração dos sensores.

The figure displays two identical web server calibration forms, one for 'Sensor 1' and one for 'Sensor 2'. Each form is contained within a blue rounded rectangle. At the top of each form is the title 'Sensor 1' or 'Sensor 2' in bold black text. Below the title is a horizontal gray bar representing a temperature reading. Underneath the bar, the word 'Input' is followed by two white rectangular input fields. Below the input fields, the word 'Output' is followed by two more white rectangular input fields. At the bottom center of each form is a gray button with the text 'Salvar' in black.

Fonte: Autor

Figura 20 – Campo do *WebServer* relacionado a calibração dos sensores preenchido.

The image shows a web interface for calibrating a sensor. It has a blue background and contains the following elements:

- Sensor 1**: Title of the calibration page.
- 23,7**: A grey box displaying the current sensor reading.
- Input**: A section header for the calibration parameters.
- 1072** and **26552**: Two white input boxes for the sensor's minimum and maximum values.
- Output**: A section header for the calibration parameters.
- 0** and **62**: Two white input boxes for the sensor's output minimum and maximum values.
- Salvar**: A grey button to save the calibration data.

Fonte: Autor

3.4 PLATAFORMA IOT

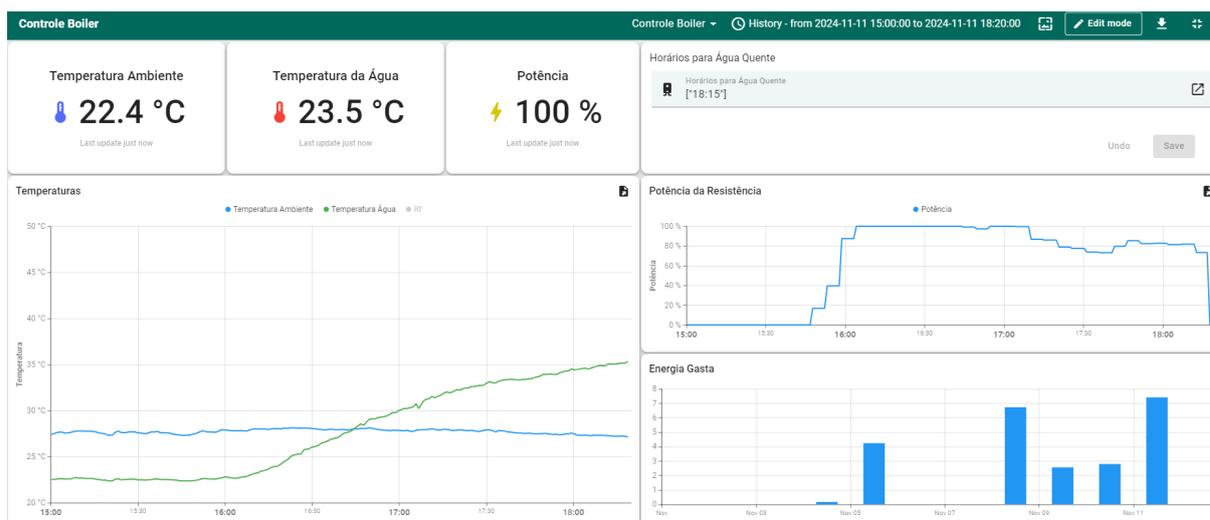
A plataforma IoT utilizada foi o *Thingsboard* (THINGSBOARD, 2021). Essa plataforma possui uma biblioteca compatível com o ESP32 e permite a criação de *dashboards* para visualizar os dados e controlar o sistema.

A comunicação entre o ESP32 e o *Thingsboard* ocorre através do protocolo MQTT (*Message Queuing Telemetry Transport*). O ESP32 envia as medições a cada 60 segundos para a nuvem, sendo elas referentes às duas temperaturas medidas, à potência enviada para a resistência e à energia gasta pelo sistema, calculada pela integral da potência. O ESP32 também coleta um atributo da nuvem, que consiste em um objeto JSON com todos os horários programados para a água ser aquecida.

A Figura 21 mostra o *dashboard* desenvolvido para o sistema. Ela contém dois gráficos, um com as temperaturas medidas pelos sensores Pt100 e outro com a potência da

resistência. Além disso, também são apresentados os *cards* contendo os últimos valores dessas variáveis. No canto superior direito, é disponibilizado um campo para o preenchimento dos horários de água quente. Ao clicar em salvar, o valor é atualizado no *Thingsboard* e é lido pelo sistema, que passa a ligar a resistência em um intervalo de 1 hora antes desses horários.

Figura 21 – *Dashboard* desenvolvido para o sistema.



Fonte: Autor

4 RESULTADOS

Para analisar os resultados, foram feitos dois tipos de testes. O primeiro consistiu em uma simulação no *MATLAB*, onde um sistema com um comportamento parecido ao do *boiler* foi simulado e seu modelo foi obtido a partir do MQR. Com esse modelo, o controlador proposto anteriormente foi calculado e simulado em malha fechada com a planta. A segunda rodada de testes ocorreu em uma bancada criada para o projeto, a qual é formada por uma caixa térmica e uma resistência de 200W para criar o sistema térmico.

A metodologia dos testes seguiu a seguinte sequência:

1. Aquecimento por duas horas e meia para estimação da planta através do método MQR;
2. Cálculo automático do controlador e filtro de referência pelo ESP32;
3. Espera para o esfriamento da água;
4. Teste de aquecimento com o controle aplicado.

Os dados do teste foram enviados pelo ESP32 para a plataforma IoT, que permite a exportação desses dados em um arquivo CSV (*Comma-Separated Values*). Esses arquivos CSV contêm tanto a temperatura da água quanto a potência enviada para a resistência. Esses valores foram utilizados para analisar os resultados do sistema.

Durante a realização desse trabalho, não foi possível realizar a instalação no sistema em um *boiler* para a realização de testes oficiais. Porém, esses testes se mantêm como planos futuros do projeto, para validar o sistema desenvolvido em sua aplicação real.

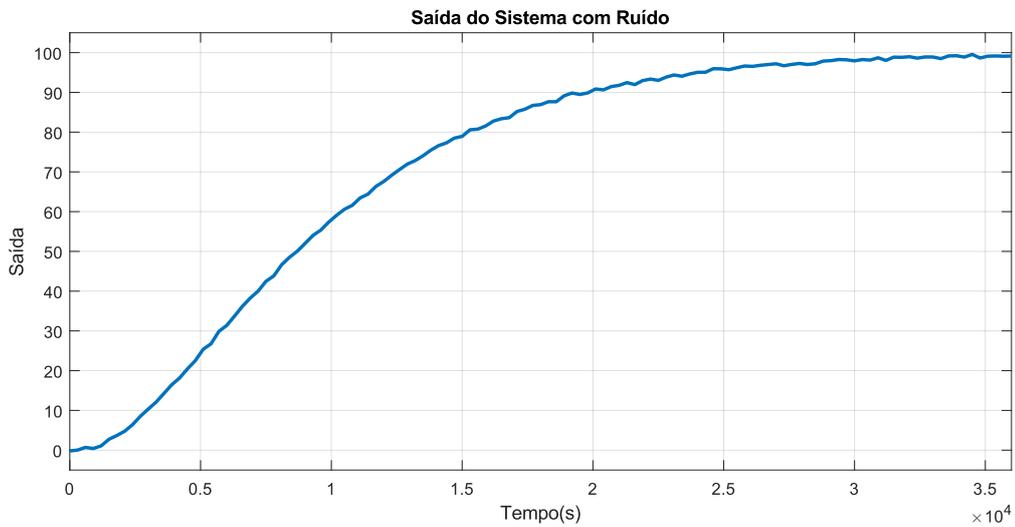
4.1 SIMULAÇÃO

Para simular o comportamento do sistema de controle proposto, criou-se uma simulação no *MATLAB*. Um sistema com função de transferência

$$G(s) = \frac{100}{(5000s + 1)^2} \quad (43)$$

foi simulado com a entrada sendo um degrau unitário. A resposta desse sistema serviu de entrada para o algoritmo MQR. Para aumentar a proximidade do teste com as condições reais, foi adicionado um ruído de baixa intensidade na saída, conforme mostrado na Figura 22.

Figura 22 – Saída do sistema com ruído.



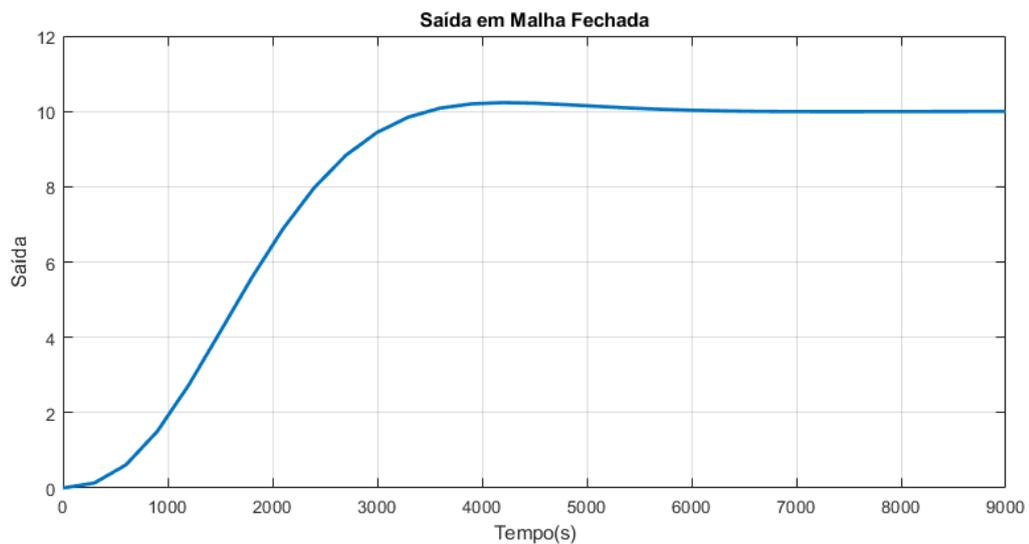
Fonte: Autor

O sistema calculado pelo algoritmo MQR serviu de base para os cálculos de controle mostrados na Seção 3.2, resultando no controlador a seguir:

$$C = \frac{1.736 z^2 - 2.852 z + 1.193}{z^2 - 1.017 z + 0.01731}. \quad (44)$$

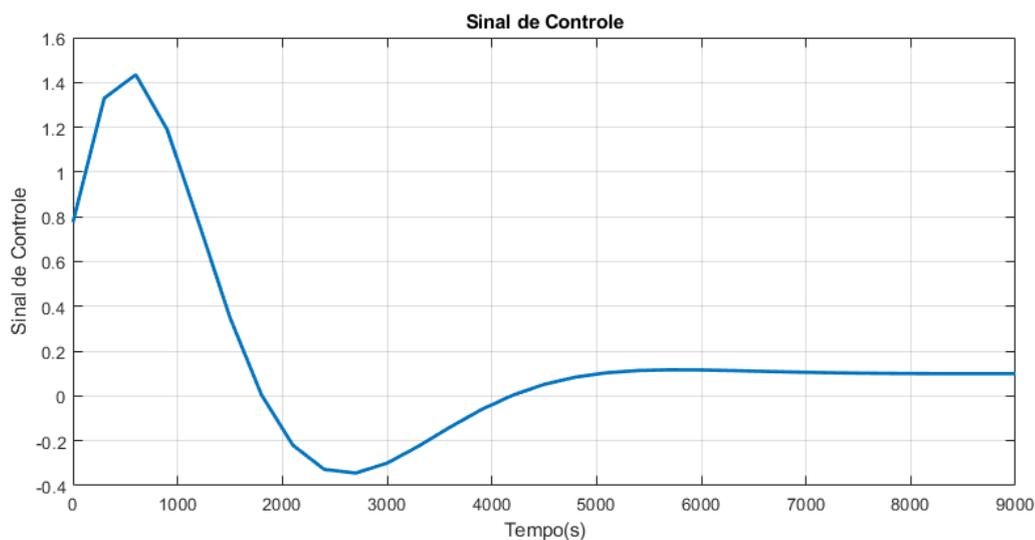
Simulando esse controle com a planta original, obtém-se a saída mostrada na Figura 23 e o sinal de controle da Figura 24.

Figura 23 – Saída do sistema em malha fechada.



Fonte: Autor

Figura 24 – Sinal de controle do sistema.



Fonte: Autor

Durante a simulação, o sinal de controle apresenta sinal negativo. Isso representaria, no sistema físico, que a resistência estaria retirando calor da água. Valores negativos de sinal de controle não são possíveis nesse caso, sendo saturados em 0. Isso pode resultar em um sobressinal na saída do sistema real.

4.2 TESTE EM BANCADA

Para testar o sistema na prática, montou-se uma estrutura de teste, utilizando uma caixa térmica e uma resistência de 200W acoplada ao sistema. A Figura 25 mostra a estrutura desenvolvida para a realização dos testes.

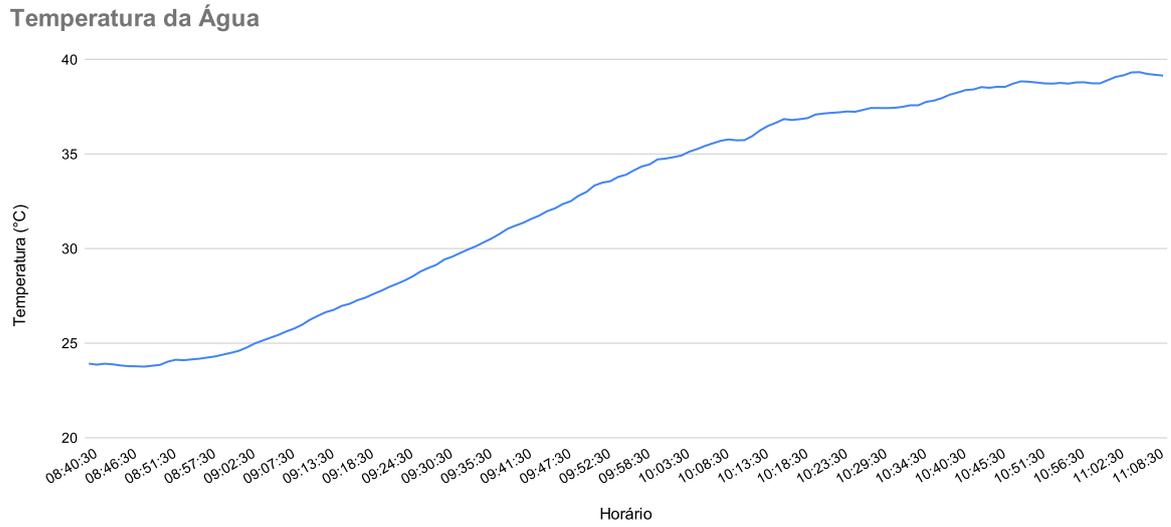
Figura 25 – Estrutura desenvolvida para testar o sistema.



Fonte: Autor

Durante a primeira etapa, a potência da resistência ficou em 100% durante aproximadamente duas horas e meia. Durante esse tempo o algoritmo de MQR calculou uma aproximação discreta de segunda ordem do sistema. Logo após desligar, esses parâmetros foram usados para calcular o controlador. A Figura 26 mostra a temperatura do sistema durante a realização do MQR.

Figura 26 – Saída do sistema durante o algoritmo de MQR.



Fonte: Autor

Essa saída resultou na função de transferência

$$G(z) = \frac{0,01664 z + 0,00586}{z^2 - 0,99547 z + 0,04466}, \quad (45)$$

no controlador

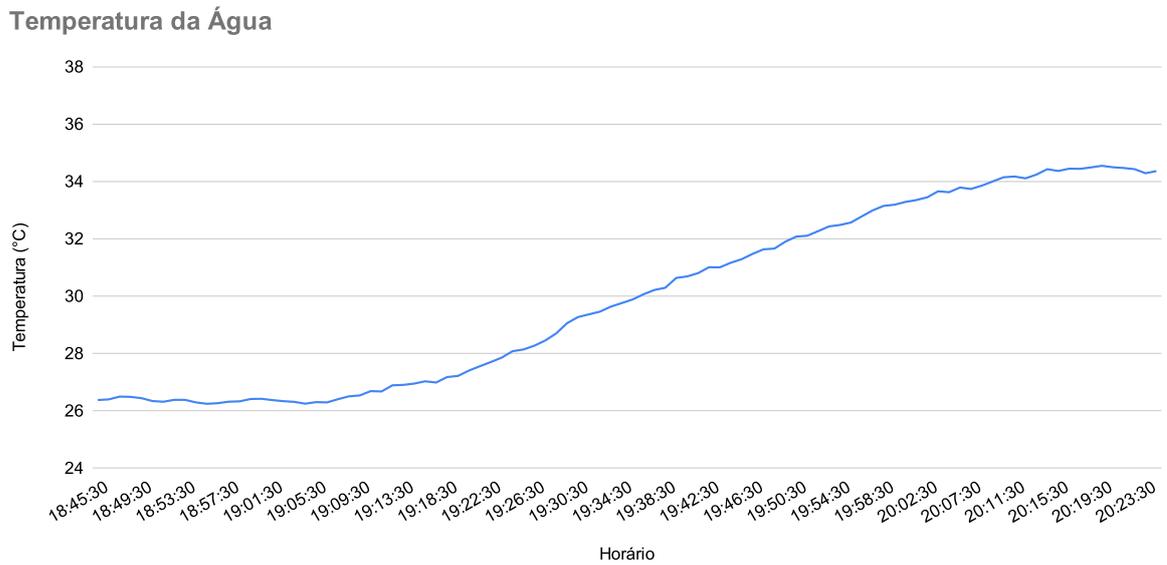
$$C(z) = \frac{35,872 z^2 - 62,059 z + 27,36}{z^2 - 2,2014 z + 1,2014} \quad (46)$$

e no filtro de referência

$$F_R(z) = \frac{1,173 z^2}{35,872 z^2 - 62,059 z + 27,36}. \quad (47)$$

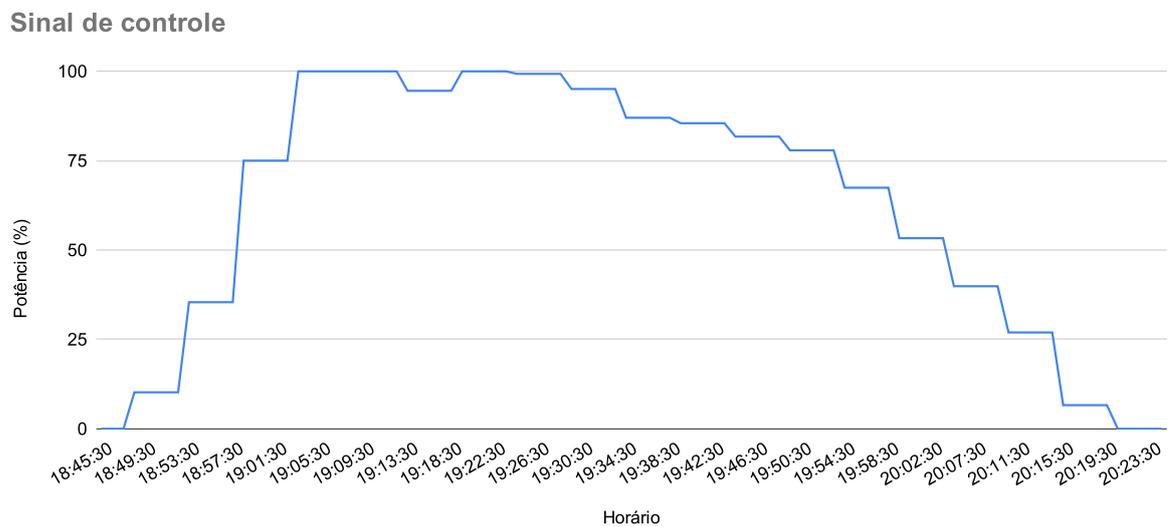
Já durante a segunda etapa de teste, o controle foi aplicado, com uma referência em 35°C. Foram realizados dois testes de aquecimento com o sistema de controle. A Figura 27 mostra a temperatura do sistema e a 28 mostra o sinal de controle durante o primeiro teste.

Figura 27 – Saída do sistema no primeiro teste.



Fonte: Autor

Figura 28 – Sinal de controle do sistema no primeiro teste.



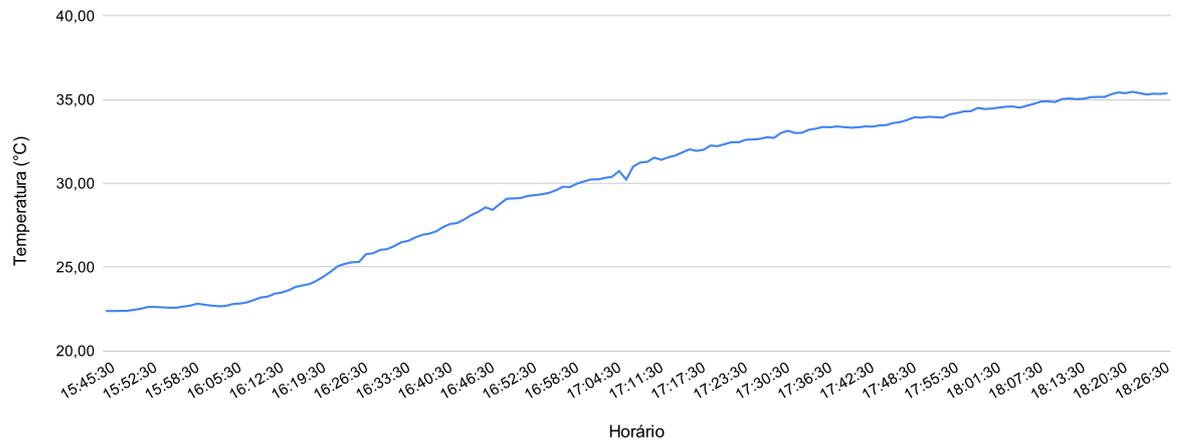
Fonte: Autor

A temperatura fica um pouco abaixo da referência, atingindo um pico de $34,5^{\circ}\text{C}$. Porém, esse erro é considerado aceitável. Analisando o sinal de controle, é perceptível que ele não satura demasiadamente, o que poderia causar um efeito de *windup*.

As Figuras 29 e 30 mostram, respectivamente, a temperatura e o sinal de controle durante o segundo teste.

Figura 29 – Saída do sistema no segundo teste.

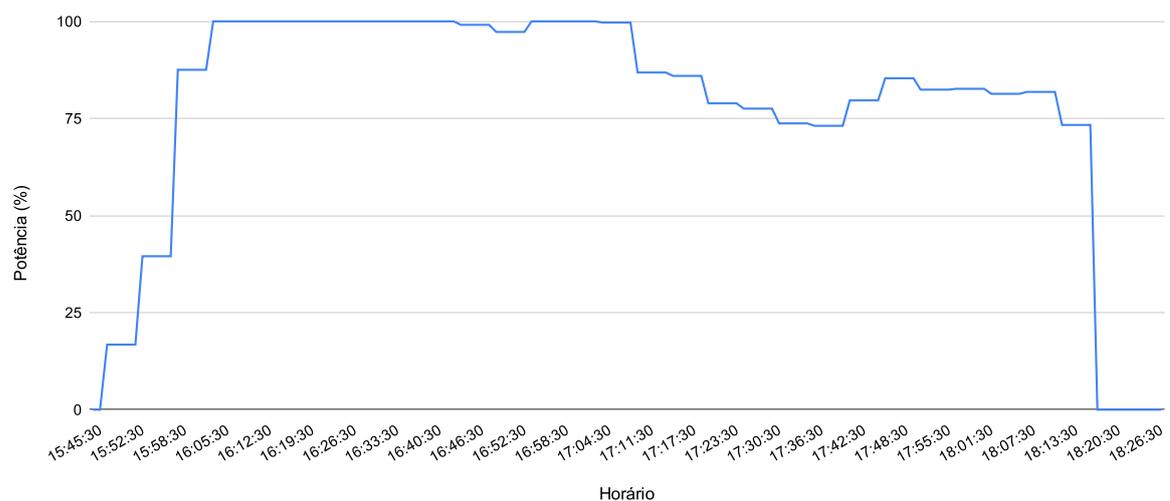
Temperatura da Água



Fonte: Autor

Figura 30 – Sinal de controle do sistema no segundo teste.

Sinal de Controle



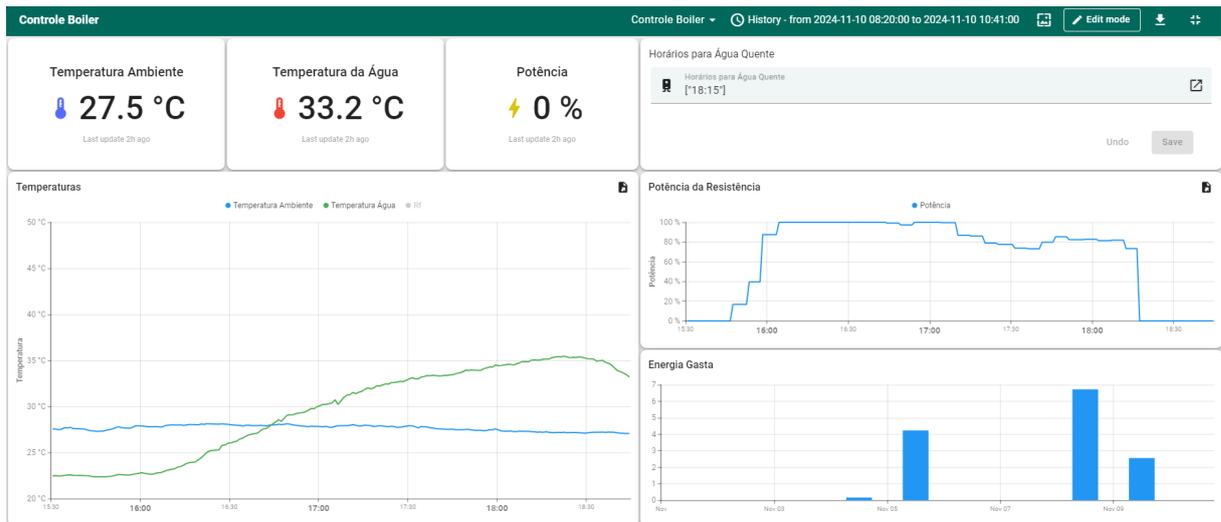
Fonte: Autor

Nesse teste, a temperatura final ficou um pouco acima da referência, chegando a 35,5°C. Como esse teste iniciou de uma temperatura mais baixa, acabou sendo necessário mais que as duas horas previstas. Portanto, o sistema ficou ligado por aproximadamente

duas horas e meia. Isso se deve ao fato da resistência ser fraca para o sistema, fato que não aconteceria no *boiler* em si, visto que duas horas seria suficiente para esquentar mesmo iniciando de temperaturas mais baixas.

A Figura 31 mostra o *dashboard* com os dados do teste de aquecimento realizado. Além das curvas já apresentadas anteriormente, essa figura mostra os dados de gasto de energia, onde pode-se analisar a eficiência energética do controle PID.

Figura 31 – *Dashboard* do usuário mostrando os dados de aquecimento.

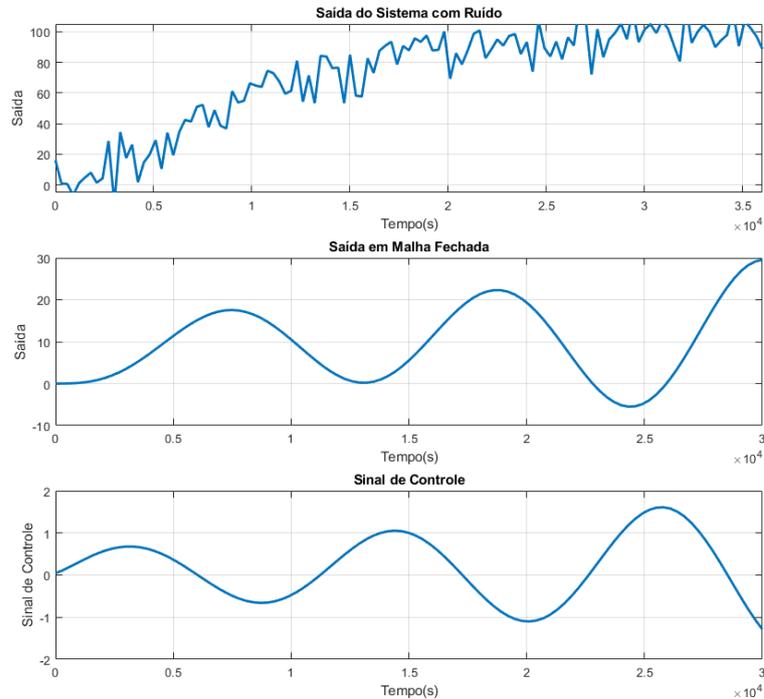


Fonte: Autor

4.3 ANÁLISE DOS RESULTADOS

Durante a etapa de testes via simulações, notou-se que o algoritmo proposto para o controle era eficaz no processo de sintonia automática do controlador PID. Porém, é importante ressaltar alguns problemas encontrados. O primeiro problema foi observado no caso de sinais muito ruidosos entrando no MQR. A Figura 32 mostra uma leitura muito ruidosa que serviu como entrada para o MQR. Esse sinal gerou uma diferença grande entre o sistema real e o estimado, resultando em um ajuste inadequado do PID. Conseqüentemente, o sistema em malha fechada apresentou um comportamento instável. Porém, o sistema de medição de temperatura do sistema não apresenta um ruído tão elevado e conta ainda com um filtro de média móvel para diminuir eventuais ruídos. Portanto, isso acaba não representando um problema no sistema real.

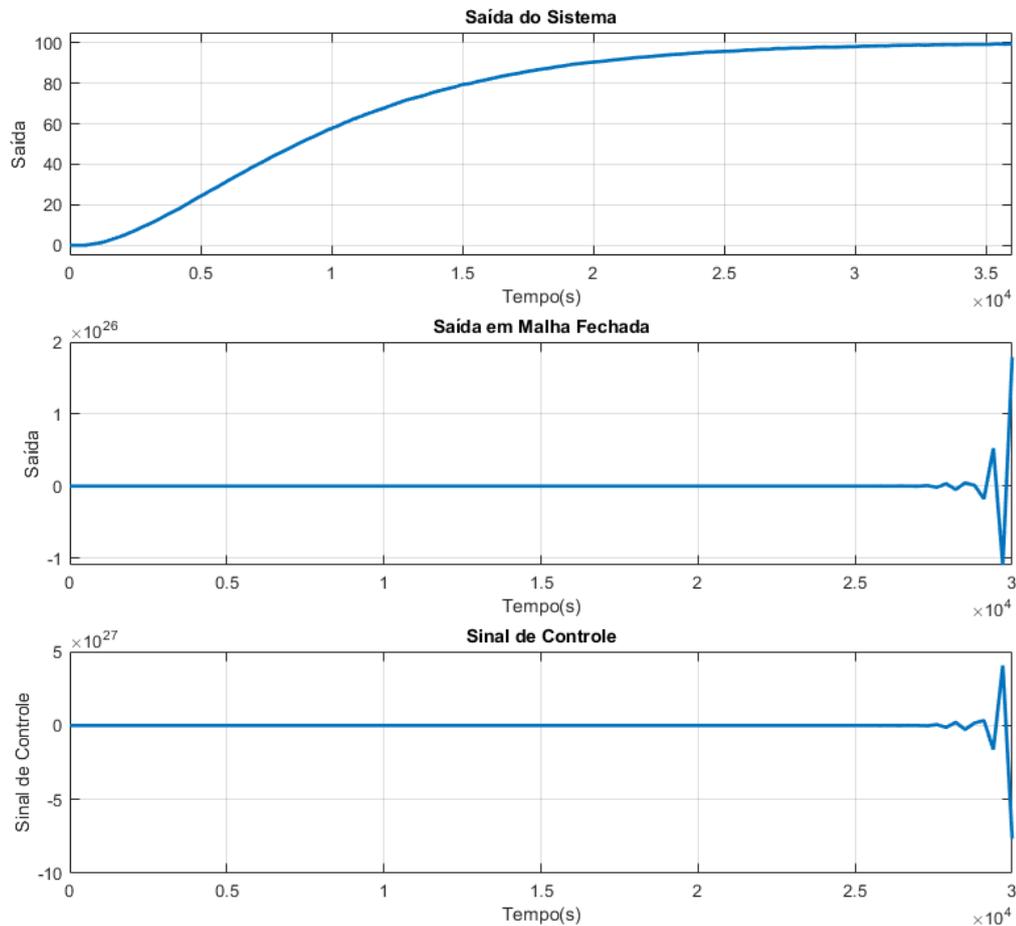
Figura 32 – Leitura, Saída em malha fechada e Sinal de controle para simulação com muito ruído na leitura ($\pm 40\%$ do valor de regime permanente).



Fonte: Autor

Outro problema notado durante as simulações foi a alocação de polos com módulo muito próximo de 1. Como o controlador é calculado com base na estimação da planta, podem haver leves diferenças entre o sistema real e o estimado. Caso o controlador tente alocar os polos com módulo próximo de 1, essa diferença pode criar polos instáveis (módulo maior que 1) no sistema em malha fechada. Por exemplo, caso o sistema tentasse alocar 4 polos em $-0,95$, o resultado seria o mostrado na Figura 33. O cálculo de controle acabou gerando polos em $-1,5844 \pm 0,8334 i$.

Figura 33 – Leitura, Saída em malha fechada e Sinal de controle para simulação instável.



Fonte: Autor

Nos testes em bancada, pode-se testar o sistema na prática e aperfeiçoar o algoritmo de controle. Notou-se que a planta atingiu a referência com o comportamento dinâmico esperado e manteve um erro pequeno ($\pm 0,5^\circ\text{C}$). Além disso, o sinal de controle se manteve em faixas aceitáveis em relação à saturação. Pode-se atestar também o funcionamento correto do MQR para estimar a função de transferência da planta. O objetivo desse teste foi analisar o sistema de controle e fazer os ajustes necessários, uma vez que o acesso à bancada para ajustes era mais fácil do que ao *boiler*, que costuma ficar em cima das residências. Foram necessárias diversos testes para alcançar um algoritmo que apresentasse o resultado esperado.

5 CONCLUSÃO

Esse trabalho abordou o desenvolvimento de um sistema IoT para monitoramento e controle do aquecimento em um *boiler* residencial. O sistema desenvolvido foi projetado para otimizar o processo de aquecimento da água, garantindo maior conforto ao usuário. A sintonia automática do controlador PID permite um ajuste dinâmico e preciso dos parâmetros de controle, ajustando-se às variações de temperatura, o que representa uma importante melhoria em relação aos sistemas tradicionais de aquecimento.

A funcionalidade de monitoramento remoto, que envia dados de temperatura para a nuvem, proporciona ao usuário um acesso fácil e em tempo real às condições do sistema, aumentando a conveniência e a sensação de controle. Além disso, a possibilidade de programar horários para o acionamento do *boiler* pelo celular não apenas oferece maior flexibilidade, mas também contribui para o uso mais eficiente dos recursos, ajustando-se às necessidades diárias dos moradores de maneira inteligente.

Com os resultados obtidos, é possível concluir que o sistema de aquecimento automatizado desenvolvido é uma solução inovadora que alia conforto, praticidade e eficiência energética. A integração de tecnologias como o controle PID e IoT não só aprimora o desempenho do sistema, mas também abre portas para novas possibilidades de automação residencial, como a integração com outros dispositivos e a criação de soluções mais inteligentes e sustentáveis para o lar.

REFERÊNCIAS

- 4N25, 4N26, 4N27, 4N28. **Vishay**. [S.l.], jan. 2010. Rev. 1.8. Disponível em: <https://www.vishay.com/docs/83725/4n25.pdf>.
- BTA40, BTA41, BTB41. **ST Microelectronics**. [S.l.], 2023. Disponível em: <https://www.st.com/resource/en/datasheet/bta40.pdf>.
- ELETROSOL. **Boiler para Aquecimento Solar**. 2022. Disponível em: <https://www.eletrosolindustria.com.br/boiler-aquecimento-solar>. Acesso em: 1 nov. 2024.
- ESP32. **Espressif Systems**. [S.l.], 2024. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- FETTE I. E MELNIKOV, A. **RFC 6455: The WebSocket Protocol**. USA: RFC Editor, 2011.
- HX711. **AVIA Semiconductor**. [S.l.], 2021. Disponível em: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf.
- JOVANOVIĆ, Jelena; DENIĆ, Dragan. Mixed-mode Method Used for Pt100 Static Transfer Function Linearization. **Measurement Science Review**, v. 21, n. 5, p. 142–149, 2021.
- MOC3021, MOC3022, MOC3023. **Fairchild Semiconductor**. [S.l.], 2000. Disponível em: <https://www.farnell.com/datasheets/97984.pdf>.
- OGATA, Katsuhiko. **Engenharia de Controle Moderno**. Edição: Roger Trimer. [S.l.]: Pearson, 2014. P. 577. https://www.academia.edu/45620848/ENGENHARIA_DE_CONTROLE_MODERNO_5a_EDIÇÃO_OGATA(visitado 2024-11-15).
- PUHLMANN, Henrique Frank Werner. **PID digital – Método de cálculo numérico**. 2014. Disponível em: <https://embarcados.com.br/pid-digital-metodo-de-calculo-numerico/>. Acesso em: 1 set. 2024.
- REIS SOARES, Pedro Manuel Oliveira dos. **Discretização de Controladores Contínuos**. 1996. Diss. (Mestrado) – Faculdade de Engenharia da Universidade do Porto.

SANTOS, Juliano Augusto Rosa dos; SOUSA CHAVES, Camilla de. **ANÁLISE DA APLICAÇÃO DA PONTE DE WHEATSTONE PARA A MEDIÇÃO DE TEMPERATURA**. [S.l.], 2022. Disponível em:

<http://200.131.116.17/index.php/enpe/article/view/283/177>.

SOLESTROL. **Aquecedor Solar Soletrol Digital**. 2022. Disponível em:

<https://www.soletrol.com.br/extras/aquecedor-solar-digital/>. Acesso em: 1 nov. 2024.

SOLESTROL. **Como Funciona o Aquecedor Solar de Água Soletrol**. 2022.

Disponível em:

<https://www.soletrol.com.br/extras/como-funciona-o-aquecedor-solar-soletrol/>. Acesso em: 16 ago. 2024.

TECHNOLOGY, Sensor. **Sensores de Elementos**. 2021. Disponível em:

<https://www.sensor-technology.com.br/elementos-sensores>. Acesso em: 1 set. 2024.

THINGSBOARD. **Thingsboard**. 2021. Disponível em: <https://thingsboard.io>. Acesso em: 11 nov. 2024.

WEAST, Robert C. (Ed.). **Handbook of Chemistry and Physics**. [S.l.]: Chemical Rubber Co., 1972. 53rd Edition.