

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

CONTROLADOR DE TEMPERATURA E DE VAZÃO DE
UM CHUVEIRO ELÉTRICO

Eduardo Albieri Coelho da Silva

2013

CONTROLADOR DE TEMPERATURA E DE VAZÃO DE UM CHUVEIRO ELÉTRICO

Eduardo Albieri Coelho da Silva

Projeto de Graduação apresentado ao Curso de Engenharia Mecânica da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Fernando Augusto de Noronha Castro Pinto

Rio de Janeiro

DEZEMBRO de 2013

CONTROLADOR DE TEMPERATURA E DE VAZÃO DE UM CHUVEIRO ELÉTRICO

Eduardo Albieri Coelho da Silva

PROJETO DE GRADUAÇÃO SUBMETIDA AO CORPO DOCENTE DO CURSO DE ENGENHARIA MECÂNICA DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO MECÂNICO.

Examinada por:

Prof. Fernando Augusto de Noronha Castro Pinto, Dr.-Ing.

Prof. Ricardo Eduardo Musafir; DSc

Prof. Thiago Gamboa Ritto; DSc

RIO DE JANEIRO, RJ - BRASIL

DEZEMBRO de 2013

Silva, Eduardo Albieri Coelho da
Controle de temperatura e de vazão de um chuveiro
elétrico/ Eduardo Albieri Coelho da Silva. – Rio de
Janeiro: UFRJ/ Escola Politécnica, 2013.

VI, 47 p.: il.; 29,7 cm.

Orientador: Fernando Augusto de Noronha Castro
Pinto

Projeto de Graduação – UFRJ/ Escola Politécnica/
Curso de Engenharia Mecânica, 2013.

Referencias Bibliográficas: p. 43.

1. Controladores. I. Pinto, Fernando Augusto de Noronha Castro. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Mecânica. III. Controlador de temperatura e vazão de um chuveiro elétrico.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como
parte

dos requisitos necessários para a obtenção do grau de Engenheiro Mecânico.

CONTROLADOR DE TEMPERATURA E DE VAZÃO DE UM CHUVEIRO ELÉTRICO

Eduardo Albieri Coelho da Silva

Dezembro/2013

Orientador: Fernando Augusto de Noronha Castro Pinto

Curso: Engenharia Mecânica

Nesse trabalho iremos estudar a aplicação de um controlador PID (Proporcional Integral Derivativo) de temperatura e vazão em um chuveiro para uso doméstico. Demonstraremos como foi feito e o funcionamento do chuveiro, as dificuldades de implementação e as facilidades geradas ao usuário. O foco do trabalho foi prático na intenção de observar dados reais e analisá-los. Para isso foi montado um chuveiro elétrico com um controlador de temperatura e vazão. Esse controle consiste em sensores que medem as variáveis do sistema, atuadores que agem sobre o sistema e uma placa micro controladora que contém a programação e processará as medições e controlará os atuadores.

Palavras-chave: Controlador, PID, Chuveiro, Temperatura e Vazão.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment
of
the requirements for the degree of Engineer.

TEMPERATURE AND FLOW CONTROLLER OF AN ELETRIC SHOWER

Eduardo Albieri Coelho da Silva

December/2013

Advisor: Fernando Augusto de Noronha Castro Pinto

Course: Mechanical Engineering

In this work, we will study the application of a temperature and flow PID controller in a domestic electric shower. The paper shows how it was made and how it works, the implementation difficulties and the facilities generated for the user. The paper focus was practical in order to observe real data and analyze. To this was assembled an electric shower with a temperature and flow controller. The control consists in sensors that make the measure of systems variables, the actuators that acts on the system and micro controller board that contain the logarithm and it will process the measures and control the actuators.

Keywords: Controller, PID, temperature, flow.

Sumário

1. Introdução	1
2. Descrição do chuveiro	3
2.1 Lista de componentes	4
2.2 Descrição e função dos componentes	5
2.3 Funções do Arduino	11
2.3.1. PWM	11
2.3.2. Rotina ExternalInterrupts	12
3. Modelagem matemática	14
3.1 Temperatura	14
3.2 Vazão	16
3.3 Temperatura x Vazão	17
4. Controle	19
4.1 Linearidade	20
4.2 Distúrbios ou perturbações	21
4.3 Controlador Liga-Desliga	21
4.4 Controlador PID.....	23
4.5 PID versus Sistemas não-lineares	25
4.6 Programação	26
5. Aquisição e Análise dos dados	28
5.1 Definição das constantes do controlador de temperatura	28
5.2 O controlador de temperatura e vazão	34
6. Considerações finais	39
6.1 Conclusão	39
6.2 Trabalhos futuros	40
Referências bibliográficas	42
Anexo I - Desenvolvimento da programação do controlador	43

Capítulo 1

Introdução

Cada vez mais é comum ouvir os termos concorrência, inovação e sustentabilidade. Na economia de mercado em que vivemos, sobrevive a empresa que está sempre à frente de seus concorrentes e para isso é necessário inovar. Por meio de inovações, as empresas lançam produtos e serviços novos no mercado, desenvolvem processos internos e de produção mais eficientes, novas estratégias de marketing, diminuem custos, aumentam receitas etc. A empresa que não inova está fadada ao fracasso ou será engolida pelo mercado. Já o termo sustentabilidade está relacionado com o Desenvolvimento Sustentável, que por sua vez é definido como o desenvolvimento que atende às necessidades do presente sem comprometer a possibilidade de as gerações futuras atenderem às suas próprias necessidades.

É nesse cenário que os controladores ganham cada vez mais espaço. Inovações no campo de engenharia de controle e automação viabilizam paralelamente o crescimento e o uso racional dos recursos, por meio de técnicas avançadas que aumentam a eficiência de processos.

Outra razão para expansão dos estudos e da aplicação de sistemas de controle é o conceito de domótica, termo que é o resultado da junção das palavras Domus (casa) com Robótica (controle automatizado), é aplicação dos sistemas de controles no uso doméstico com objetivo de gerir os recursos habitacionais. A ideia base de um sistema de controle no lar é a mesma que a industrial, o que muda é o contexto para qual o sistema de controle é projetado.

Enquanto na indústria o foco é o aumento da eficiência, em casa prioriza-se o aumento da qualidade de vida. Na indústria, o sistema de controle atuará sobre o aumento da produtividade e a redução de custos enquanto no ambiente doméstico atuará sobre tarefas do dia a dia que visam economizar tempo e o desgaste de rotinas para reduzir o estresse e as preocupações do cotidiano. Outra vantagem é a economia

de recursos, logo a de dinheiro também. Dessa forma, o usuário pode dedicar mais de seu tempo e dinheiro para diversão e/ou trabalhos que exigem mais atenção e responsabilidade.

Os benefícios da domótica ou da automação residencial, são grandes se comparados com os sistemas isolados. Além de trazer mais conforto e segurança para os usuários, reduz o consumo de recursos como a água, energia elétrica e gás.

Alguns exemplos de área de aplicação da domótica são automação, iluminação, climatização, segurança e comunicação.

Este trabalho tem como objetivo principal conceber um chuveiro com controle de temperatura e vazão por meio de uma placa com um micro controlador, sensores e atuadores. Outro objetivo foi coletar dados, analisá-los e compará-los com os estudos da engenharia de controle.

As principais atividades do projeto foram o estudo da viabilidade do projeto, a especificação dos componentes, a montagem do chuveiro, a programação do algorítmico de controle, o estudo dinâmico do controlador e coleta e análise dos dados.

Capítulo 2

Descrição do chuveiro

A ideia foi construir um chuveiro elétrico com um controlador de vazão e temperatura. O trabalho teve uma diretriz prática, visou primeiramente a concepção do chuveiro para assim obter dados para fazer a análise do sistema.

Antes de tudo, foi necessário fazer um pequeno modelo/croqui de como funcionaria o projeto.

Nesse projeto, o controle da temperatura se faz por meio de um PWM (Pulse Width Modulation) e a vazão por meio de uma válvula acoplada a um motor. Um termopar foi instalado dentro de um chuveiro e um medidor de vazão em suas conexões para a medição da temperatura e a vazão, respectivamente. O controle é feito por uma placa controladora e a interface com o usuário se dá por meio de uma tela LCD com botões de seleção de temperatura e vazão. Algo bem simples e prático para o usuário.

Por questões práticas, decidiu-se comprar um chuveiro elétrico e fazer todas as modificações e adaptações necessárias para concepção do projeto. O chuveiro é alimentado por uma tensão de 110 Volts AC.

A placa controladora escolhida foi o Arduino. O Arduino é uma placa que contém um microcontrolador com entradas e saídas analógicas e digitais. Sua escolha se deu por sua simplicidade, praticidade, boa para iniciantes, o fato de ser geralmente usada em projetos acadêmicos e de ser relativamente barata.

A placa funciona com um software próprio. Nesse software é feita programação do controle dinâmico do sistema e por meio de um cabo é feito o upload para a placa.

O sensor de temperatura escolhido foi o NTC. Ele foi incluído na saída do chuveirinho do chuveiro com um suporte adaptado. Dessa forma o sensor fica em

contato direto com a água, aumentando a eficiência na medição. Embebedamos com silicone todo o contato do sensor com a água, para que fosse impermeabilizado e não houvesse interferência no sinal de medição.

Para atuação na temperatura da água no chuveiro, utilizamos um PWM, que será explicado mais à frente na seção 2.3.1. O Arduino já vinha com a essa função (outra vantagem da placa).

Para fazer essa transmissão de sinal PWM entre a alimentação e a placa foi inserido um relé de estado sólido.

O medidor de vazão foi comprado para fazer a medição da vazão do sistema. O sensor é uma peça pequena e leve e é instalado nas conexões de entrada de água do chuveiro.

Para o sistema de atuação na vazão foi projetado um motor acoplado a uma válvula. Duas peças foram concebidas para integrar o sistema. Uma que transmite o torque do motor a válvula e outra que seria o suporte do motor na própria válvula.

A transmissão entre o motor e a válvulas é engastada. Um suporte deve ser projetado para que haja restrição do movimento de suas conexões. Essa conexão não pode ser engastada. Duas conexões engastadas entre as mesmas peças devem ser muito precisas, pois se não, pode formar um sistema hiperestático podendo causar tensões internas no sistema e danificá-lo.

O motor escolhido foi um de corrente contínua e deve atuar nos sentidos horário e anti-horário, que fariam a abertura e o fechamento da válvula para o controle da vazão. Dois relés e uma alimentação DC externa viabilizam isso.

Uma vez pesquisada todas as soluções existentes, foi definido o funcionamento do chuveiro e listaram-se os componentes necessários à concepção do projeto.

2.1 Lista de componentes

Para montar o controlador e chuveiro, foi necessário adquirir os seguintes componentes:

- 1 Chuveiro elétrico;
- 1 placa Arduino Uno;
- 1 sensor de temperatura;
- 1 medidor de vazão;
- 1 válvula gaveta;
- 1 motor elétrico;
- 2 potenciômetros;
- 1 tela de LCD;
- 2 relés de 2 contatos de 5 volts
- 1 relé de estado sólido;
- Alimentação de 9 Volts DC;
- Protoboard e cabos;

Também foi necessário desenhar e fabricar duas peças:

- Transmissão entre a saída do motor e a válvula;
- Suporte do motor.

2.2 Descrição e função dos componentes

O chuveiro escolhido foi um convencional de 4.800 Watts, 110 Volts AC e 40 Amperes de corrente. O chuveiro tem 4 estágio de temperatura. Foi utilizado o segundo estágio que fornecia 2.000 Watts de potencia.

O Arduino Uno (Figura 2.1) é uma placa de microcontrolador programável em C/C++. Com entradas e saídas, o Arduino funciona com uma programação em looping. Na programação, é feita leitura dos sinais de entrada, bem como seus

tratamentos. Nela também contém a modelagem dinâmica do sistema, o modelo matemático que efetivamente fará o controle e os sinais de saída que vão para os atuadores. Entre suas características, destacam-se:

- Saída de 5 Volts
- Terra
- Pinos de entradas analógicas: são entradas que medem de terra a 5 Volts porém escalonado em 1024 valores diferentes onde o terra é igual a 0 e 5 Volts é igual a 1023.
- Pinos de entradas e saídas digitais: são entradas ou saídas booleanas que podem ser terra ou 5 Volts. Os valores são 0 ou 1 ou LOW ou HIGH. Essas entradas digitais podem ter outras funções; usaremos 2 delas que serão explicadas mais a frente.



Figura 2.1: Arduino

O sensor NTC (Figura 2.2) funciona como uma resistência variável com a temperatura. Dessa forma, a entrada analógica do controlador foi usada para conectar o sensor de temperatura.



Figura 2.2: Sensor NTC

Para calibração do sensor, é necessário medir duas temperaturas conhecidas e assim por interpolação, tem-se uma função do sensor com a temperatura ambiente. As

duas temperaturas escolhidas foram 26 e 38 graus, pois é um intervalo próximo da temperatura da água em um banho.

O medidor de vazão (Figura 2.3) tem como saída um sinal de onda quadrada cuja frequência varia com o fluxo. A função de transferência entre o fluxo e a frequência é definida pelo fabricante e consta na folha de especificação do medidor.

$$Q = 60 \cdot \frac{f}{7,5} \left[\frac{L}{hora} \right] = \frac{f}{7,5} \left[\frac{L}{min} \right]$$

Mede-se a frequência do sinal pelas funções `attachInterrupt()` e `rpm()`. Por meio da função acima se obtêm o valor do fluxo. Seu funcionamento é explicado na seção 2.1.2.



Figura 2.3: Medidor de vazão

Uma válvula gaveta foi utilizada para controlar a vazão.

O motor (Figura 2.4) adquirido já vem com uma redução de 1000:1, trabalha de 3 a 9 V e tem torque de 0,88 N.m a 6 V e corrente de Stall de 1,6 Ampère. O motor deve trabalhar com uma fonte de 9 V devido ao Arduino somente fornecer 5 V e 50 mA. Para isso foram utilizados os relés.



Figura 2.4: Motor com a redução

O potenciômetro é uma resistência ajustável. Nesse caso, a resistência varia linearmente com o giro de um pequeno eixo. Conectaram-se os dois potenciômetros de 10 k Ω nas entradas analógicas para a definição dos parâmetros de referência por meio de interpolação.

A tela LCD (Figura 2.5) de 16x2 (duas linhas com dezesseis caracteres cada) foi utilizada para visualização pelo usuário das temperaturas e vazões de referências e do sistema.



Figura 2.5: Tela de LCD

O relé de estado sólido (Figura 2.6) foi usado para amplificar o sinal do PWM, uma vez que o Arduino tem saída de 5 Volts DC e a alimentação do chuveiro é de 110 Volts AC. O relé suportava até 25 Amperes de corrente.



Figura 2.6: Relé estado sólido



Figura 2.7: Relé de contato

Outros dois relés de contato (Figura 2.7), que além de amplificadores, foram usados para alterar a polaridade do motor elétrico viabilizando o fornecimento de torque nos dois sentidos, para a abertura e o fechamento da válvula. A Figura 2.8 mostra o funcionamento dos relés de contatos. Se ambos os contatos passam 0 V ou 9 V, não diferença de potencial, logo o motor na gira. Se um contato passa 0 V e outro passa 9 V o motor gira em um sentido e se invertemos os contatos o motor gira em outro sentido.

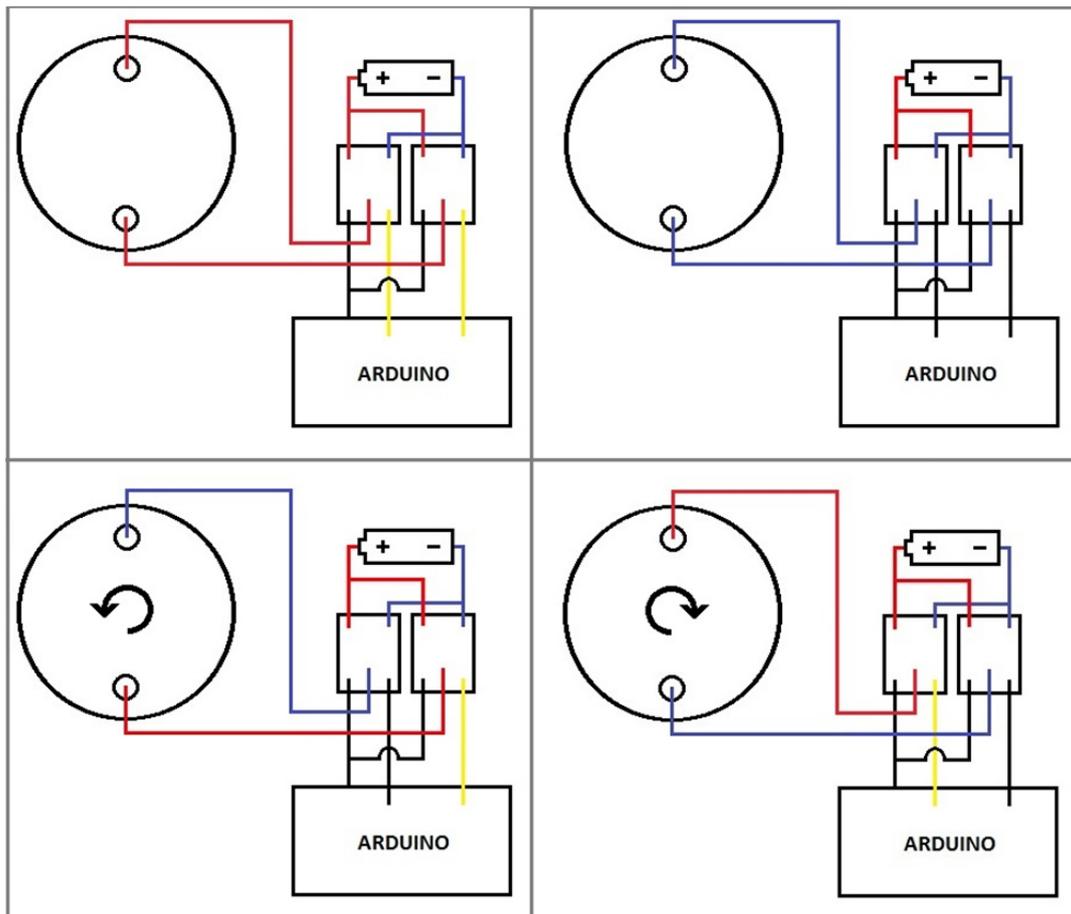


Figura 2.8: Inversão de polaridade do motor com os relés.

Foi necessário desenhar e fabricar mais duas peças para o bom funcionamento do acionamento da válvula gaveta, uma transmissão entre a saída do motor e a válvula e um suporte do motor.

A transmissão do motor para a válvula é fixada por dois parafusos enquanto o suporte é fixo no motor e “apoiado” na válvula. Isso foi projetado para que o sistema não ficasse hiperestático. Dessa forma, o sistema pode ter uma folga/grau de liberdade que evita alguns tipos de tensões internas que possam vir a danificá-lo. As figuras 2.9, 2.10 e 2.11 mostram fotos do protótipo concebido.

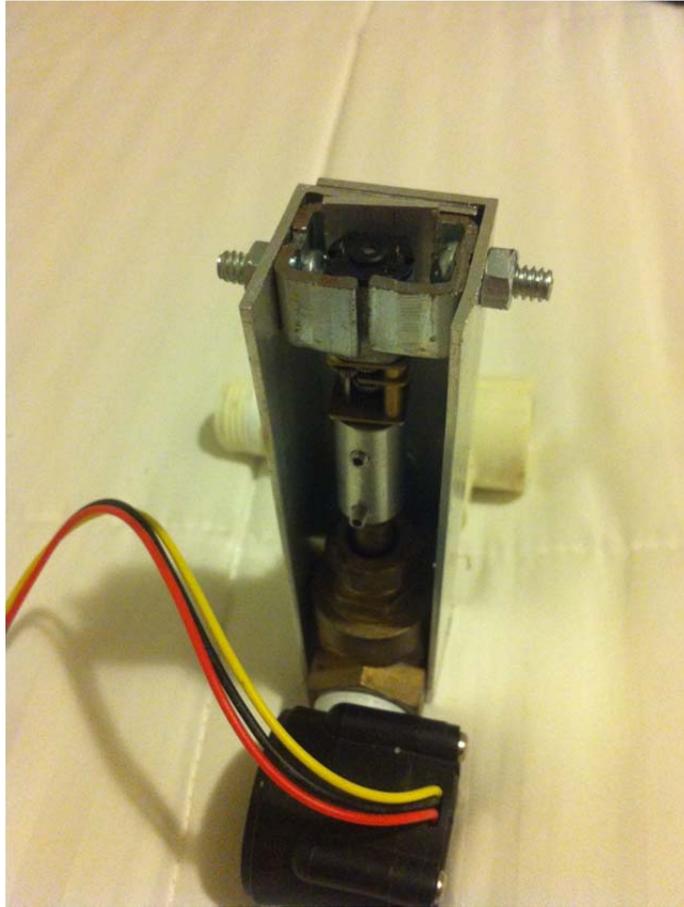


Figura 2.9: Suporte e transmissão do motor.



Figura 2.10: Montagem da transmissão do motor e a válvula gaveta.



Figura 2.11: Adaptação do sensor de temperatura na saída do chuveirinho

2.3 Funções especiais do Arduino

As entradas digitais podem ser usadas com algumas funções especiais, dentre elas destacam-se duas que foram usadas:

2.3.1 PWM

Pulse Width Modulation, ou em português, modulação por largura de pulso é uma função que tem como saída uma onda quadrada onde se ajusta uma razão de tempo em que ela será ligada (1) ou desligada (0). Essa onda tem uma frequência constante e alta, de forma que é imperceptível a transição entre o ligar e desligar, parecendo que o sinal é constante fornecendo uma razão da potência total.

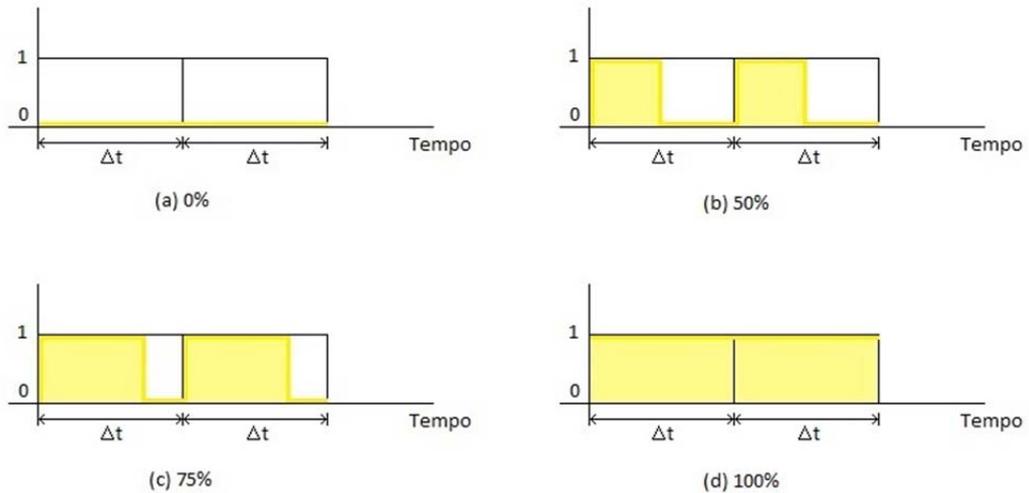


Figura 2.12: Funcionamento de um PWM

Nesse controlador, o PWM será usado para controlar a tensão fornecida à resistência de forma a controlar a temperatura da água. A figura 2.12 mostra quatro gráficos que demonstram o funcionamento do PWM. O eixo das ordenadas é a tensão e a área do gráfico é a potência fornecida ao sistema.

A função PWM no Arduino associa o número 0 a 0% e 255 a 100%.

2.3.2 Rotina Externalinterrupts

Especifica uma função a ser chamada quando ocorre uma interrupção. A interrupção pode ser disparada de quatro formas diferentes pré-definidas.

- LOW - provoca a interrupção sempre que o pino está em 0,
- RISING - provoca a interrupção quando o valor do pino muda de 0 para 1,
- FALLING – provoca a interrupção quando o valor do pino muda de 1 para 0,

- CHANGE - provoca a interrupção sempre que o pino muda de valor.

Em RISING, quando a função `attachInterrupt()` está ativada, toda vez que há a transição de 0 para 1 no pino digital indicado, essa função chama uma outra função pré definida (`rpm()`). Essa função `rpm()` veio descrita no DataSheet do medidor de vazão e faz uma contagem de quantas vezes ela foi chamada. Nesse projeto, essa função tem o papel de ler a frequência do medidor de vazão para posteriormente ser usada no cálculo da vazão do sistema.

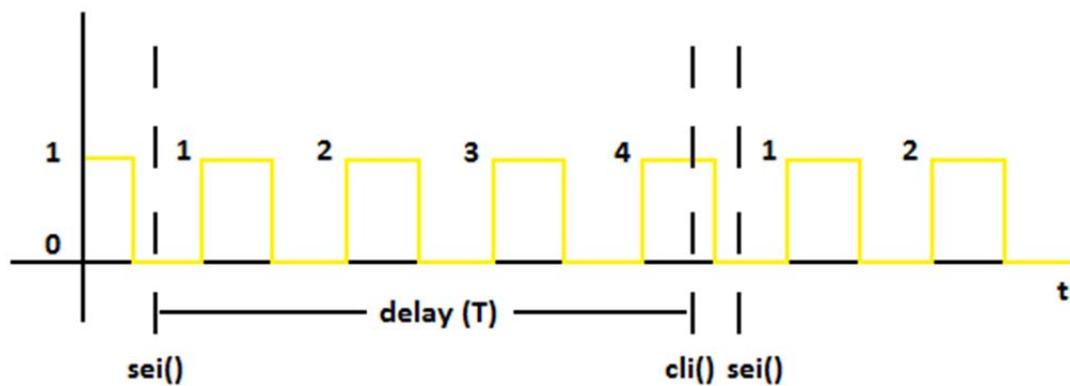


Figura 2.13: Função `attachInterrupt()`

A Figura 2.13 ilustra o funcionamento da função. Ela é ativada por meio do comando `sei()`, um intervalo de tempo é pré-definido e após a função é desativada pelo comando `cli()`. Quando ativada, a transição chama a função `rpm()` que é responsável pela contagem. Tendo assim, uma frequência como resultado.

Capítulo 3

Modelagem matemática

3.1 Temperatura

A modelagem da dinâmica da temperatura é descrita pela fórmula:

$$\eta \cdot \dot{Q} = \dot{m} \cdot c \cdot \Delta T \quad (3.1)$$

ou

$$T = \eta \frac{\dot{Q}}{\dot{m} \cdot c} + T_0 \quad (3.2)$$

onde:

$T \Rightarrow$ Temperatura de saída, a variável que se deseja controlar.

$\eta \Rightarrow$ Razão de potência fornecida pelo PWM, varia de 0 a 1.

$\dot{Q} \Rightarrow$ Potência máxima do chuveiro, 2.000W.

$\dot{m} \Rightarrow$ Vazão mássica do sistema.

$c \Rightarrow$ Calor específico do fluido, no caso, a água, $c = 1 \text{ cal/g} \cdot ^\circ\text{C} = 4.180 \text{ J/kg} \cdot ^\circ\text{C}$

$T_0 \Rightarrow$ Temperatura do fluido na entrada.

$\dot{v} \Rightarrow$ Vazão volumétrica do sistema.

$\rho \Rightarrow$ Massa específica da água.

$$\Delta T = T - T_0$$

A vazão do sistema, que também será controlada, influencia a dinâmica da temperatura de saída. Assim como a temperatura de entrada também. Porém, para a simplificação do modelo, assumiremos que as flutuações dessas variáveis são pequenas de forma que serão tratadas como distúrbios.

Por hipótese, assumiremos:

$$T_0 = 24^\circ\text{C} \quad (3.3)$$

$$\dot{v} = 2 \text{ L/min} = \frac{1}{30} \text{ L/s} \quad (3.4)$$

Como:

$$\rho = 1 \text{ kg/L} \quad (3.5)$$

$$\dot{m} = \frac{1}{30} \text{ kg/s} \quad (3.6)$$

Logo, para a temperatura temos:

$$T = \eta \cdot 14,35 + T_0 \quad (3.7)$$

$$T = \eta \cdot 14,35 + 24 \quad (3.8)$$

Para controlar a temperatura usaremos um controlador PID (Proporcional Integral e Derivativo). Este modelo calcula um erro que é a diferença entre o valor medido e o valor desejado, e por meio de fatores proporcionais, derivados e integrais minimiza-se esse erro, tendendo a 0. Esse modelo será explicado melhor no capítulo 4.

Pela fórmula (3.7) e como $\eta_{\max} = 1$ (máxima razão de potência que o chuveiro pode fornecer), inferimos que pelo modelo e as hipóteses assumidas, o aumento máximo de temperatura que o chuveiro pode realizar é de 14,35 °C.

3.2 Vazão

Para o modelo da vazão:

$$\dot{v} = \alpha \cdot \dot{v}_0 \quad (3.9)$$

Onde:

$\dot{v} \Rightarrow$ Vazão volumétrica de saída. A variável que se deseja controlar.

$\alpha \Rightarrow$ Ângulo da válvula gaveta. Variável que será controlada. Varia de 0° a 1440°.

$\dot{v}_0 \Rightarrow$ Vazão volumétrica de entrada do sistema.

A modelagem do sistema de vazão depende da vazão onde o chuveiro vai ser instalado (\dot{v}_0). O tratamento que daremos à vazão de entrada é a mesma que a da temperatura de entrada supracitada.

Por hipótese, foi definida uma vazão média e suas flutuações serão tratadas como distúrbios.

$$\dot{v}_0 = 2 \text{ L/min} = \frac{1}{30} \text{ L/s} \quad (3.10)$$

Para controlar a vazão, usaremos um controlador liga e desliga. Isso porque não temos como saber a posição angular da válvula, ou seja, não temos um sensor que meça essa posição. Dessa forma, também não temos como informar ao motor quanto que ele deve abrir ou fechar a válvula. Inviabilizando, assim, a ação de um controlador PID na vazão do sistema.

O sistema de controle funcionará da seguinte forma: quando a vazão for inferior ao ponto de ajuste, o motor abrirá um pouco a válvula e se for superior, o motor irá fechar um pouco a válvula.

A diferença está em que em vez de ligar e esperar atingir o ponto de ajuste para desligar, como funcionaria um controlador liga e desliga normal, iremos ligar e desligar o motor por um pequeno período de tempo, para que o ajuste se dê de forma mais precisa.

Para uma eficiência maior do controle da vazão, foi proposto duas faixas de tempo constantes para atuação do motor na válvula. Quando o módulo do erro for acima de 0,8 a abertura ou fechamento da válvula será maior que quando o módulo do erro estiver entre 0,3 e 0,8.

Esta solução foi proposta devido ao atraso entre a atuação e leitura do sistema, ou seja, ao abrir ou fechar a válvula, existe um atraso na estabilização da vazão. Outra razão foi o tempo necessário de leitura do medidor. Como já descrito, o medidor interrompe o programa durante um tempo para fazer a medição, afastando mais ainda a leitura e atuação da vazão.

Outra razão para o controle ser do tipo liga e desliga foi o fato do medidor de vazão ter uma precisão pequena e o chuveiro elétrico trabalhar com faixa pequena de vazão. Seria muito difícil fazer o tratamento do erro.

Essa solução se mostrou boa mesmo com um tempo alto de estabilização em grandes erros. Para pequenos erros ou distúrbios no sistema, o controlador corrigiu bem as flutuações na saída do chuveiro.

Uma desvantagem desse controlador é o alto tempo de estabilização uma vez que a correção do erro é feita gradualmente.

Outra desvantagem dessa solução é a necessidade de se estabelecer uma faixa de ajuste. Em vez de um ponto de ajuste, teremos uma faixa de ajuste. No caso, um ponto de ajuste mais ou menos um desvio. A razão dessa faixa de ajuste é que como estamos controlando a vazão por meio de pequenas aberturas ou fechamentos constantes, a tendência é que a saída deixe de ser uma função contínua e se transforme em uma função discreta. Dessa forma, a probabilidade do erro ser zero é muito baixa, fazendo que sempre haja uma atuação, mesmo se o erro for infinitesimal. Quando o módulo do erro for menor que 0,3, não há atuação no sistema.

3.3 Temperatura x Vazão

Se substituirmos (3.9) em (3.1) e isolarmos a diferença de temperatura temos:

$$\Delta T_{\max} = \eta \frac{\dot{Q}}{\dot{m} \cdot c} \cong \frac{\eta}{\rho \alpha} \cdot \frac{\dot{Q}}{\dot{v}_0 \cdot c} \quad (3.11)$$

Aplicando as hipóteses, temos:

$$\Delta T_{\max} \cong \frac{0,48}{\dot{v}_0} \quad (3.12)$$

Da fórmula (3.12) podemos ver que o aumento da temperatura é inversamente proporcional à vazão do sistema.

A figura 3.1 mostra o diagrama de bloco do sistema.

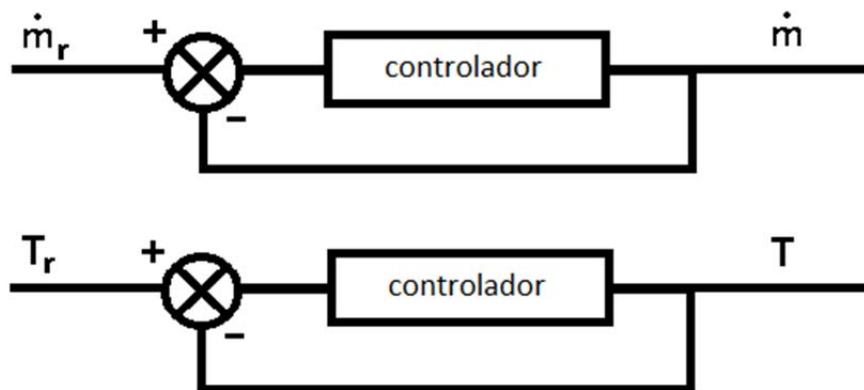


Figura 3.1: Diagrama de blocos após as hipóteses

Capítulo 4

Controle

O projeto de controladores baseia-se em um modelo matemático que aproxima o comportamento dinâmico do processo que se deseja controlar.

Os sistemas de controle são dispositivos que têm por objetivo controlar uma ou mais variáveis em um sistema. Por meio de sensores e atuadores, esses dispositivos executam tarefas sequenciais de acordo com um modelo matemático pré-programado. Os sensores são usados para a medição de uma variável que será usada como entrada no modelo dinâmico enquanto os atuadores são os responsáveis pela administração da variável a ser controlada. Por exemplo: máquina de lavar roupa, controlador de velocidade de um carro, piloto automático de um avião, ar-condicionado digital, injeção eletrônica de um motor.

Basicamente, podemos subdividir os sistemas de controle em dois grupos:

Controlador de malha aberta: são controladores não realimentados, ou seja, não utilizam o sinal de saída como entrada no controlador. Não existe a influência do sistema a ser controlado no sistema de controle. Esses controladores não possuem sensores, suas atuações são dependentes somente de rotinas relacionadas ao tempo.

Ex.: O micro ondas que para aquecer um prato de comida, por exemplo, tem como entrada uma variável de tempo, definida pelo usuário. Não há mensuração de um sinal de saída, não há verificação se a temperatura da comida está ideal, quente demais ou ainda continua fria.

Controlador de malha fechada: são controladores realimentados que, por meio de sensores, utilizam o sinal de saída e/ou uma função do sinal de saída como entrada no controlador. Existe a influência do sistema a ser controlado no sistema de controle. A diferença entre os sistemas realimentados e os sistemas de malha fechado é que

neste último trabalha-se com a diferença do sinal de saída e de entrada, chamado de erro.

Ex.: O controlador de velocidade de um carro. Uma vez definida a referência ou ponto de ajuste, o controlador compara essa referência com a velocidade real do veículo e atua de forma a ajustar a velocidade, acelerando ou freando o carro.

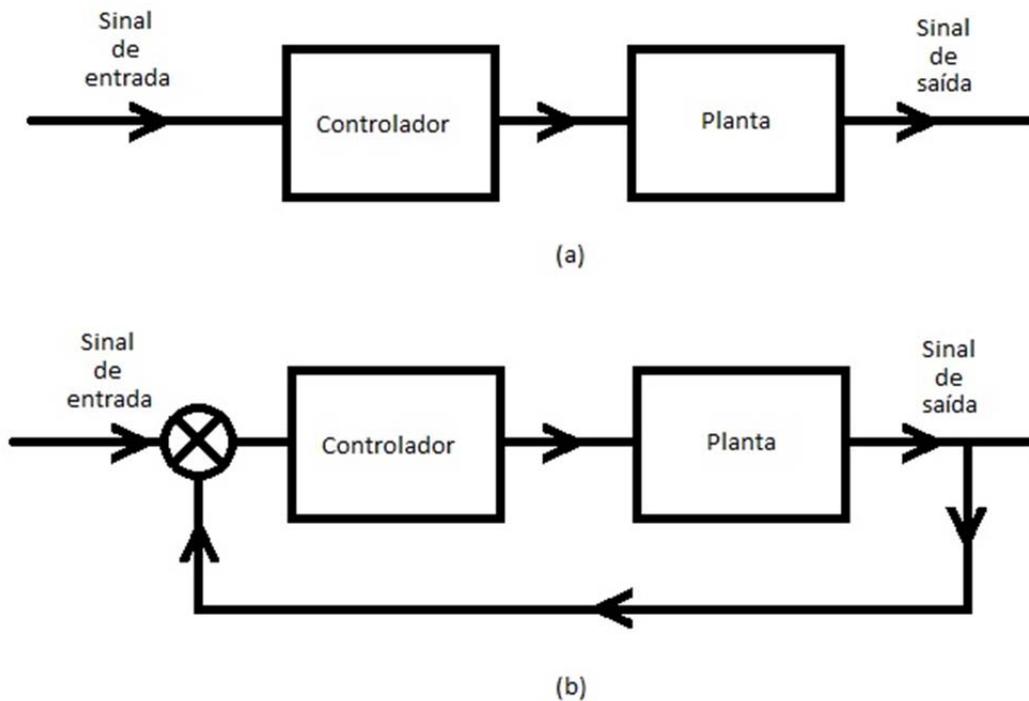


Figura 4.1: Diagrama de blocos: (a) malha aberta e (b) malha fechada

4.1 Linearidade

A propriedade da linearidade define que:

$$F(ax + by) = aF(x) + bF(y)$$

Embora muitos fenômenos sejam representados por equações lineares, na prática as relações entre eles não são efetivamente lineares. Estudos mostram que esses fenômenos podem ser lineares somente em um intervalo de operação.

Por isso, ao modelar um sistema não-linear, faz-se o que chamamos de linearização. Modelar nada mais é que definir o comportamento das variáveis de um sistema. Ao linearizar um sistema, estamos representando matematicamente seu comportamento dentro de um intervalo pré-definido.

Uma boa linearização do sistema define uma boa modelagem da dinâmica do sistema, que por sua vez proporciona uma maior acurácia do sistema de controle.

4.2 Distúrbios ou Perturbações

São todas as variáveis/estímulos que afetam/impactam o sistema que não são exercidas pelos atuadores e/ou não estejam inseridas no modelo dinâmico, ou seja, alguma interferência sobre a variável a ser controlada que não tenha sido causada pela ação do controlador ou por outra ação previamente calculada. Os distúrbios não são mensurados nem entram na modelagem, são inerentes ao meio.

4.3 Controlador Liga e Desliga

O controlador liga e desliga é o controlador mais simples que existe. A saída do controlador muda de ligada para desligada, ou vice-versa, à medida que o sinal do erro passa pelo ponto de ajuste.

Exemplo: O termostato de geladeira, se a temperatura está acima da temperatura de referência, o controlador liga o compressor para refrigerar a geladeira e, se estiver abaixo, desliga.

Esse tipo de controlador é bem rústico e é pouco preciso. Por atuar somente como ligado ou desligado, o controlador gera oscilação na variável controlada. Trabalha dentro de uma média, oscilando entre um valor máximo e mínimo. O controle dinâmico do sistema é limitado.

Geralmente, o controlador liga e desliga é utilizado em ocasiões onde o sistema não é impactado pelas flutuações da variável controlada. O processo permite essas oscilações.

As vantagens do controlador liga e desliga são a não necessidade de um estudo profundo da dinâmica do sistema, o projeto geralmente é simples e de fácil implementação, tem baixo custo e retorno em pouco tempo.

O controle de nível de água de uma caixa d'água é um bom exemplo de sistema onde o um controlador liga e desliga funciona perfeitamente. Quando o nível estiver abaixo de um valor mínimo, a bomba é acionada e, quando este nível chega a um valor máximo, a bomba é desligada. A Figura 4.2 mostra um exemplo de gráfico de um controlador liga e desliga.

Nesse caso, não tem como controlar o nível exato da água na caixa d'água, mas pode-se afirmar que o nível estará sempre entre uma faixa, nunca abaixo do nível mínimo, de forma a evitar o risco de faltar água e nunca acima do nível máximo, evitando o vazamento e o desperdício de água.

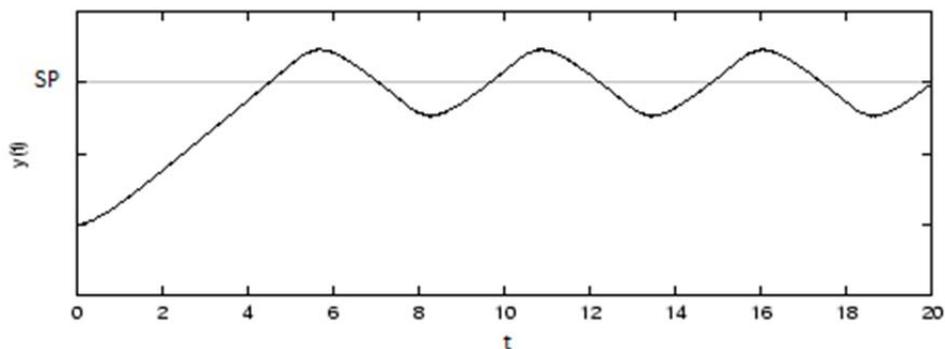


Figura 4.2: Exemplo de gráfico de um controlador liga e desliga

4.4 Controlador PID (Proporcional Integral e Derivativo)

É um controlador realimentado de malha fechada que minimiza o erro entre o sinal de saída e o de entrada contribuído por três fatores: proporcional, integral e derivado do próprio erro.

- Proporcional: indica o valor presente do erro. Quanto maior o erro, maior a atuação do controlador.

- Integral: indica a acumulação dos erros passados. Esse fator corrige o erro estacionário, ou seja, gerado por uma perturbação constante. Imagine um elevador que tem um controlador que controla a força para definir a velocidade e o nível dos andares. Para o mesmo nível (ponto de ajuste), dependendo do número de pessoas, ou seja, do peso que ele carrega, a força exercida será diferente. Ao se aproximar do ponto de ajuste, K_p e K_d tenderiam a 0, o controle iria atuar sempre de forma constante. Esse fator corrige tal erro.

- Derivado: indica a taxa de variação no tempo dos erros, ou seja, uma indicação/previsão do futuro do erro. Se essa taxa de variação estiver aumentando, significa que o sistema se afasta do ponto de ajuste, logo o fator derivativo será maior e o controlador irá atuar de forma mais expressiva para corrigir o erro. Já se essa variação estiver diminuindo, o sistema se aproxima do ponto de ajuste, o fator derivativo será negativo e diminuirá a atuação com objetivo de não extrapolar o ponto de ajuste. É um fator que amortiza o sistema, evita grandes oscilações e propicia um menor tempo de estabilização.

$$\text{Fator Proporcional} = K_p e(t)$$

$$\text{Fator Integral} = K_i \int_0^t e(t) dt$$

$$\text{Fator Derivativo} = K_d e'(t)$$

A soma desses três fatores gera o controlador PID. A compreensão de cada fator e de suas contribuições são fundamentais para a definição das constantes K_p , K_d e

K_i . Essas constantes são a alma do controlador PID, são elas, junto com uma boa modelagem dinâmica do sistema, que definem a estabilidade e o comportamento do sistema.

$$PID = K_p e(t) + K_i \int_0^t e(t) dt + K_d e'(t)$$

Também conhecida como:

$$PID = K_p \left(e(t) + \frac{\int_0^t e(t) dt}{T_i} + T_d \cdot e'(t) \right)$$

onde:

$$K_i = \frac{K_p}{T_i} \quad \text{e} \quad K_d = K_p T_d$$

Parametros	Tempo de subida	Overshoot	Tempo de estabilização	Erro estacionário	Estabilidade
↑ K_p	Reduz	Aumenta	Pouca alteração	Reduz	Reduz
↑ K_i	Reduz	Aumenta	Aumenta	Elimina	Reduz
↑ K_d	Pouca alteração	Reduz	Reduz	Sem efeito	Aumenta se K_d é pequeno

Tabela 4.1: Análise das constantes do PID

A tabela 4.1 resume o efeito de cada constante no sistema dinâmico de controle. O inverso também é válido.

Por exemplo: ao diminuir o K_p , o tempo de subida aumenta e assim por diante.

4.5 PID versus Sistemas não-lineares

Os controladores PID são lineares e simétricos, o que faz seu desempenho em sistemas não-lineares ser variável. Por exemplo, num controle de temperatura em um ambiente por meio de um ar condicionado, um caso de uso comum, o arrefecimento é ativo, mas o aquecimento passivo (arrefecimento desligado, sem aquecimento), o overshoot apenas pode ser corrigido lentamente porque não pode ser forçado. Neste caso, o PID deve ser ajustado para ser superamortecido, para prevenir ou reduzir o overshoot. Isso reduz o desempenho e aumenta o tempo de estabilização.

4.6 Programação

A programação do controlador foi feita em um software da própria placa Arduino em linguagem C/C++. O programa se encontra no Anexo I.

O Algoritmo de programação é bem simples e pode ser dividido em 3 etapas:

- Leitura das variáveis:

Leitura dos sensores e dos potenciômetros.

Cálculo das sensibilidades para a definição dos valores do sistema e da definição dos pontos de ajustes.

- Controle:

Aplicação do modelo matemático da dinâmica do controlador.

- Atuação no sistema:

Programação dos atuadores, isto é, as saídas do sistema.

Para que ocorra efetivamente o controle das variáveis e mantenha-se a estabilidade, essas três etapas devem ser executadas em sequência e de forma iterativa. Leem-se as variáveis, aplica-se o controle e atua-se no sistema. Releem-se as variáveis e assim por diante. Por isso o programa principal é um looping.

A estrutura da programação do Software consiste em:

- Declaração das variáveis:

Nessa primeira parte, declaram-se todas as variáveis e seus respectivos tipos que serão usadas em todo o programa.

Aqui também incluímos bibliotecas (`#include`). Uma biblioteca é um conjunto de comandos previamente feitos por alguém. O Arduino já possui uma rica biblioteca, cada uma desenvolvida para uma aplicação particular.

- Declaração de função (`void #nomedafunção()`):

Caso haja necessidade, pode se criar uma função para ser usada no programa. Essa função é descrita fora do programa principal.

Na figura abaixo, pode-se ver todas as variáveis declaradas e a inclusão da biblioteca <LiquidCrystal.h>, que é necessária para o funcionamento do LCD.

Vê-se também a definição da função rpm().

- Definição das configurações (void setup()):

Nesse espaço são definidas as configurações do programa.

É onde se define se o pino digital por tipo (entrada ou saída), o tamanho do display e onde se configura a função attachInterrupts().

- O programa (void loop()):

É o programa principal, onde são desenvolvidas as três etapas.

A tabela 4.2 destaca alguns comandos:

NoTopsFan++	NoTopsFan = NoTopsFan + 1
pinMode(pino, INPUT/OUTPUT);	define o pino como entrada ou saída
attachInterrupt(pino, função, TIPO);	define o pino, a função a ser chamada e o TIPO (LOW/CHANGE/RISING/FALLING)
analogRead(pino);	faz a leitura analógica (0 à 1023)
analogWrite(pino, pwm);	comanda a saída pwm (0 à 255)
digitalRead(pino);	faz a leitura digital (0 ou 1)
digitalWrite(pino, LOW/HIGH);	comanda a saída digital (0 ou 1)
sei();	habilita a interrupção
cli();	desabilita a interrupção
delay(tempo);	para o programa por um tempo (em ms)
map(x, min, max, novomin, novomax);	interpola o número x de uma escala para outra
lcd.setCursor(x,y);	define o cursor do LCD
lcd.print(var);	imprime no LCD
Serial.print();	imprime no Serial

Tabela 4.2: Comandos principais do Arduino

Capítulo 5

Aquisição e análise dos dados

Para aquisição dos dados, foi usado o comando `Serial.print()` e depois exportados para o Excel para fazer análise e geração de gráficos.

5.1 Definição das constantes do controlador de temperatura

Primeiramente, para definir as constantes do controlador de temperatura, foram feitos alguns ensaios somente utilizando o controlador de temperatura.

Nesses experimentos, usou-se um intervalo de 1 segundo entre as medições e atuações. Esse intervalo de tempo é o que é usado no projeto junto com a vazão.

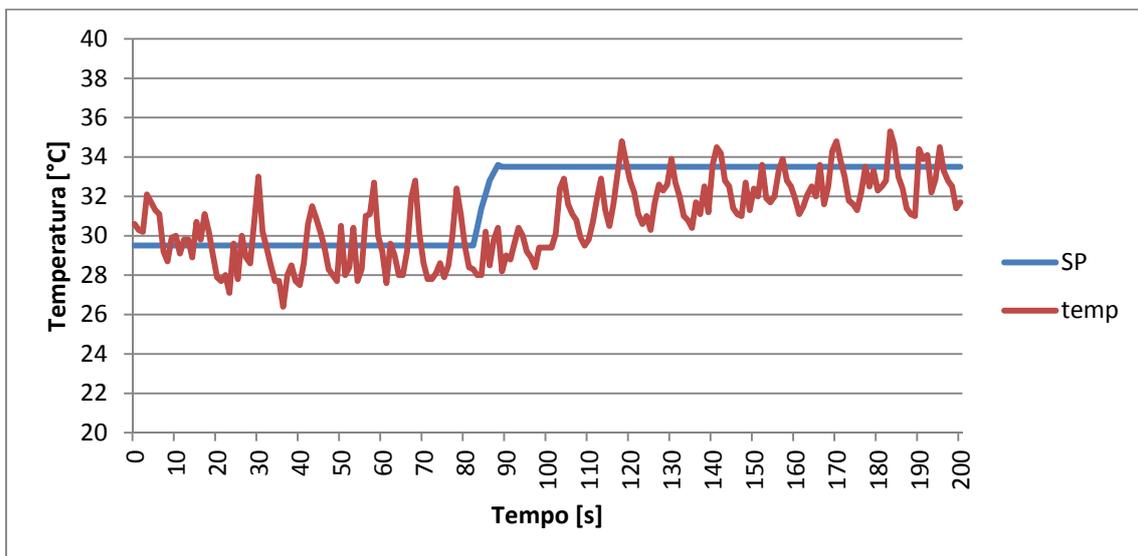


Figura 5.1 - Gráfico da temperatura no tempo do 1º experimento.

Para o primeiro experimento foram atribuídos valores arbitrários para o K_p de 10 e K_i de 0,2. Analisando a figura 5.1, o gráfico da temperatura desse experimento

mostra que o controlador responde bem ao sistema tendendo a temperatura de referência ou ponto de ajuste (SP). Porém, percebe-se que a volatilidade da medição é muito alta. Essa volatilidade alta, “atrapalha” a dinâmica do sistema. Com essa volatilidade, fica difícil também determinar o tempo de estabilização do sistema. Por isso foi incluído um filtro de média móvel de dois para atenuar essa volatilidade. A média móvel de dois usa a média entre o valor atual e o valor anterior medido. Funciona como um filtro passa baixa onde atenua as medições de alta frequência que não são interessantes para o sistema.

O primeiro abaixou-se a temperatura, esperou-se estabilizar, logo em seguida voltou à temperatura anterior e esperou estabilizar novamente.

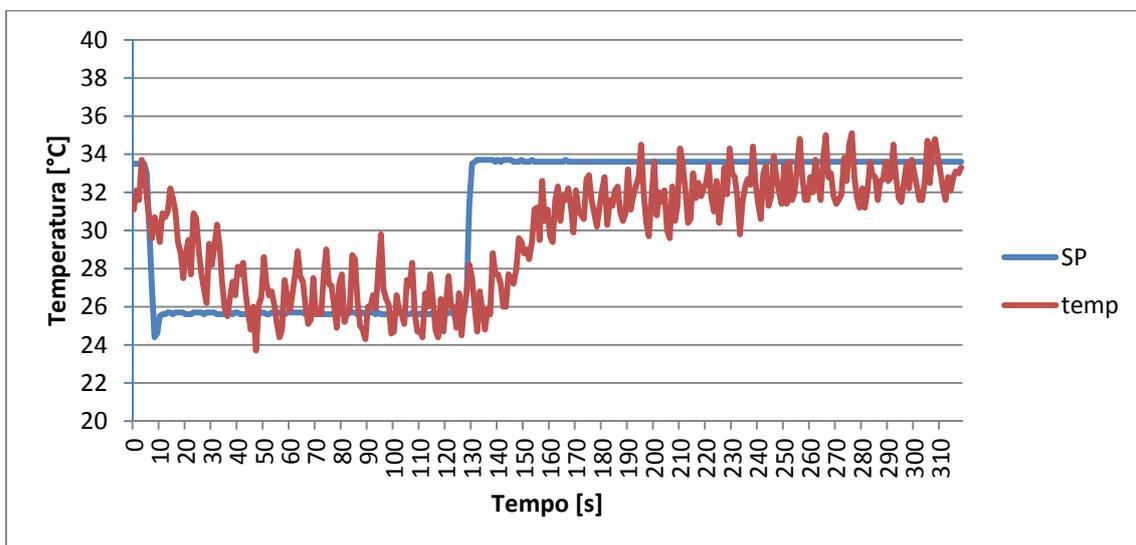


Figura 5.2 - Gráfico da temperatura no tempo do 2º experimento.

Pode-se ver pela figura 5.2 que ao inserir o filtro de média móvel, a dinâmica da leitura da temperatura fica mais estável e estabelece um padrão melhor. Vê-se também que a temperatura lida tende a seguir a temperatura estabelecida pelo ponto de ajuste, o que demonstra que o controlador está atuando de forma corretiva como queríamos.

Nesse experimento tivemos 2 modificações no ponto de ajuste. Na primeira baixamos a temperatura em torno de 8 graus, esperamos estabilizar e depois retornamos a temperatura inicial. Em relação ao tempo de estabilização, vemos que quando abaixamos a temperatura, o sistema levou em torno de 30 segundo para estabilizar enquanto quando retornamos a temperatura inicial, a estabilização se deu em torno de 65 segundos.

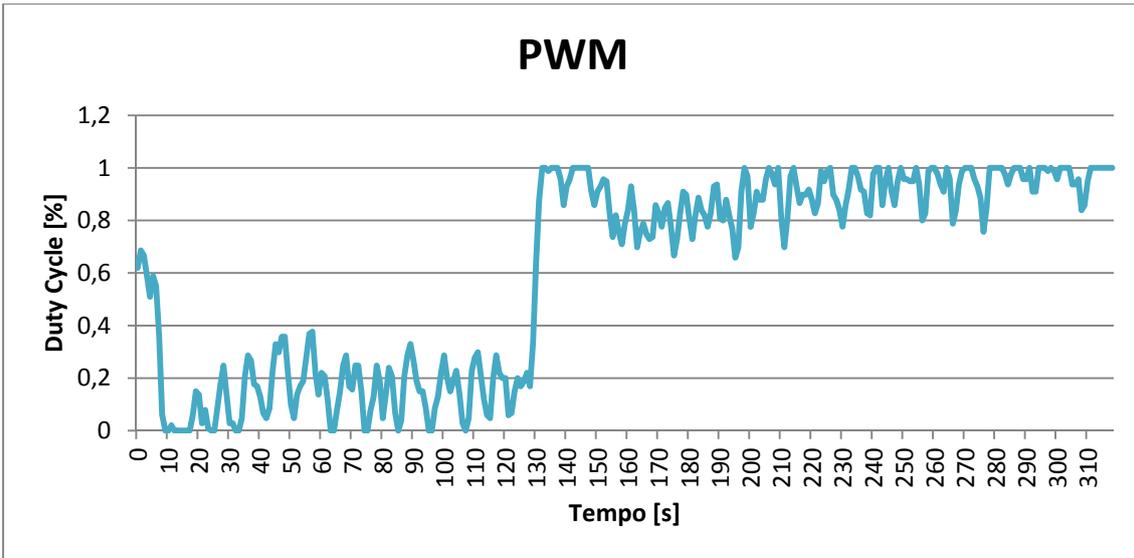


Figura 5.3 – PWM do 2º experimento.

A figura 5.3 mostra a atuação do PWM no segundo experimento.

No terceiro experimento, foi aumentado o K_p para 30 e manteve-se K_i em 0,2. A intenção foi diminuir o tempo de estabilização do sistema.

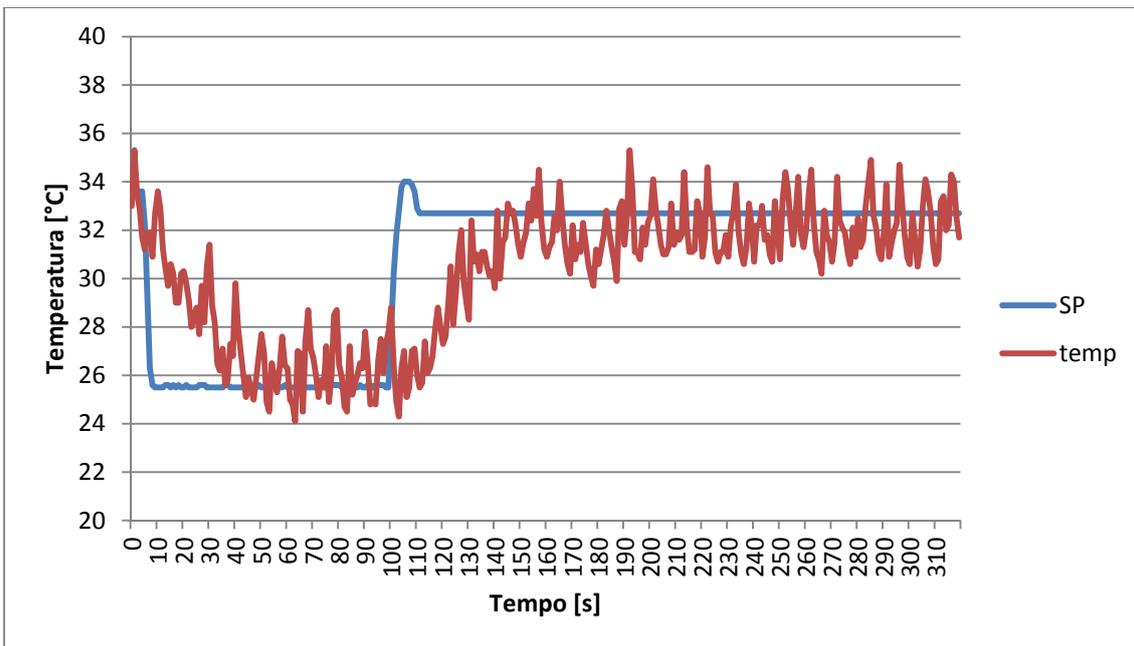


Figura 5.4 - Gráfico da temperatura no tempo do 3º experimento.

A figura 5.4 mostra que o padrão da dinâmica do sistema se manteve. O controlador atuou de forma corretiva e estabilizou bem as temperaturas.

Como esperado, pelo fato de uma constante proporcional maior, o tempo de estabilização das temperaturas foi menor. Quando se abaixou a temperatura, o sistema estabilizou-se em torno de 25 segundos enquanto aumentamos a temperatura o sistema estabilizou-se em 50 segundos. A figura 5.5 mostra a atuação do PWM no segundo experimento.

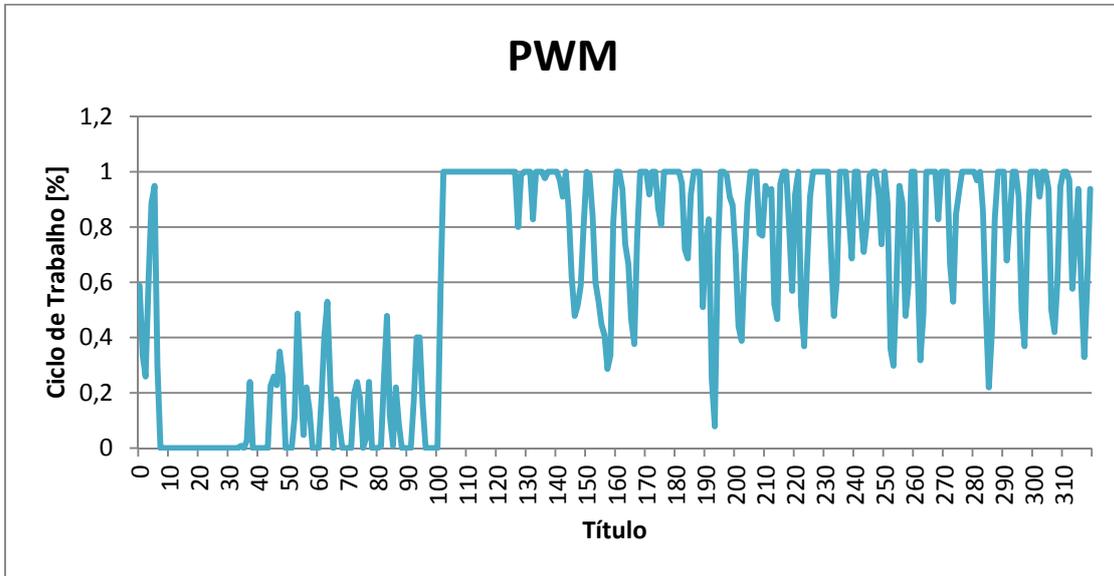


Figura 5.5 – PWM do 3º experimento.

No quarto experimento aumentou-se o K_p para 50. O resto do experimento se manteve.

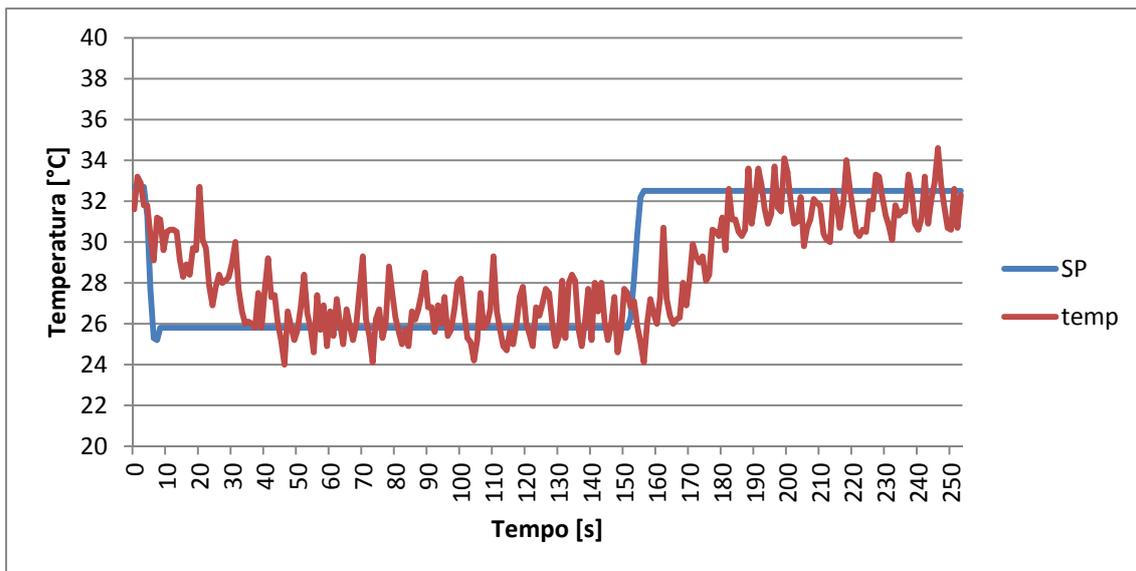


Figura 5.6 - Gráfico da temperatura no tempo do 4º experimento.

A dinâmica do experimento foi igual a das anteriores com um tempo de estabilização menor. A figura 5.6 mostra que quando se abaixou a temperatura o tempo de estabilização foi de em torno de 30 segundos enquanto aumentamos a temperatura o sistema estabilizou-se em 38 segundos.

Na verdade, o tempo de estabilização quando abaixamos a temperatura foi maior no quarto experimento que no terceiro. Isso pode ser explicado pelo fato que atuação no caso de resfriamento da água é passiva, ou seja, não há atuação. Espera-se que o sistema por si só resfrie a água decorrido certo tempo. Nesse tipo de situação, o sistema fica a mercê de situações adversas que não controlamos como, por exemplo, a temperatura de entrada da água e a perda de energia do sistema para o meio.

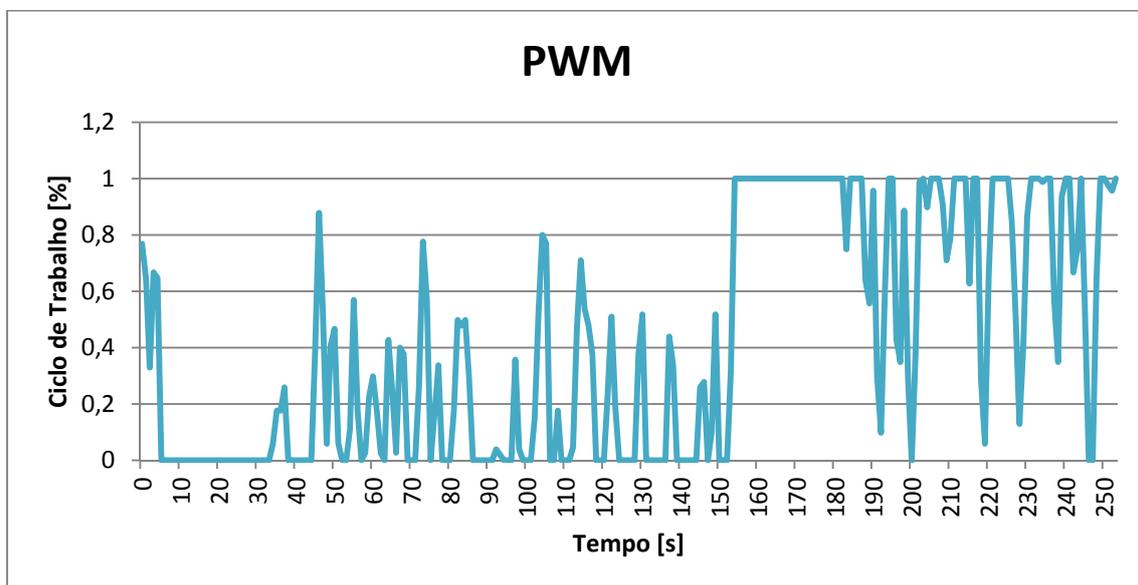


Figura 5.7 – PWM do 4º experimento.

Vê-se na figura 5.7 que entre os tempos 5 e 33 o sistema não atua, esperando passivamente a estabilização da temperatura.

Dentre o segundo, terceiro e quarto experimento. Analisaram-se os resultados e decidiu-se por escolher as constantes $K_p = 50$ e $K_i = 0,2$. Essas constantes responderam de forma eficaz.

O fato de não se ter introduzido um K_d , ou seja, um fator derivativo, se deu por conta da alta volatilidade da medição da temperatura. O fator derivativo tem como objetivo tentar antecipar erros futuros e como explicado anteriormente, ele é calculado pela variação dos erros. Para introdução de fator derivativo, seria necessário um

sensor mais preciso e/ou a introdução de métodos matemáticos de filtragem mais eficientes na dinâmica da leitura do sensor.

Foi realizado também um experimento somente com um controlador proporcional. Só que dessa vez diminuimos o intervalo de tempo entre as medições e atuações para 0,1 segundos.

Sem a constante de integração o sistema não conseguiria eliminar o erro estacionário. Porém com um constante de integração alta e um intervalo muito pequeno entre as medições e atuações, o sistema deveria oscilar com uma amplitude pequena em torno do ponto de ajuste. Essa amplitude seria pequena ao ponto de ser imperceptível na saída do sistema.

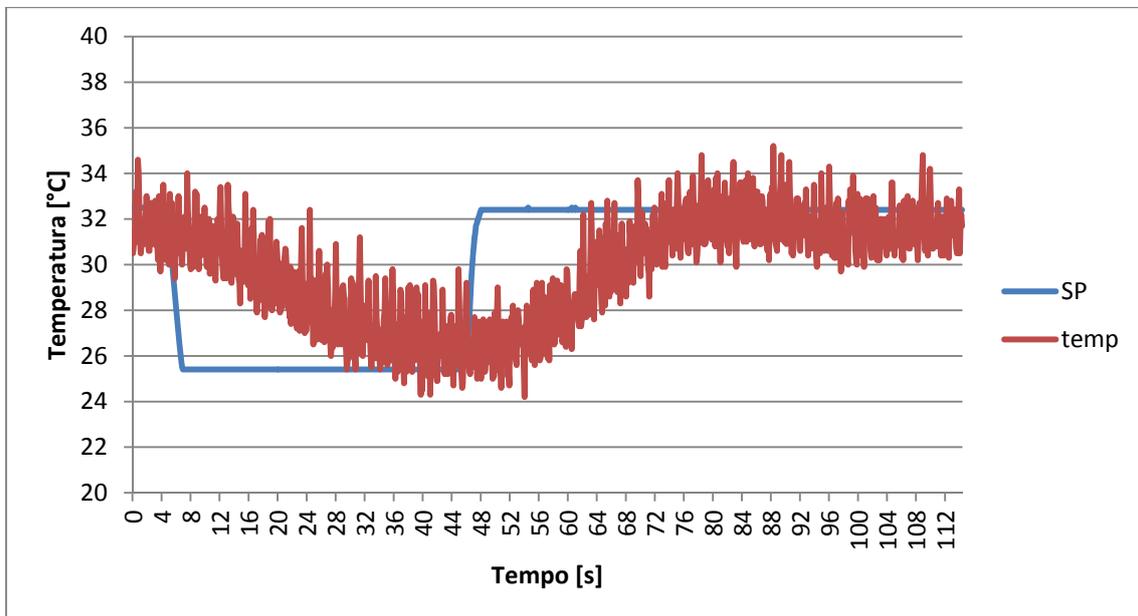


Figura 5.8 - Gráfico da temperatura no tempo do 5º experimento

A figura 5.8 mostra que o sistema respondeu como esperado e o tempo de estabilização quando aumentamos a temperatura foi de em torno de 30 segundos. A figura 5.9 mostra a atuação do PWM no quinto experimento.

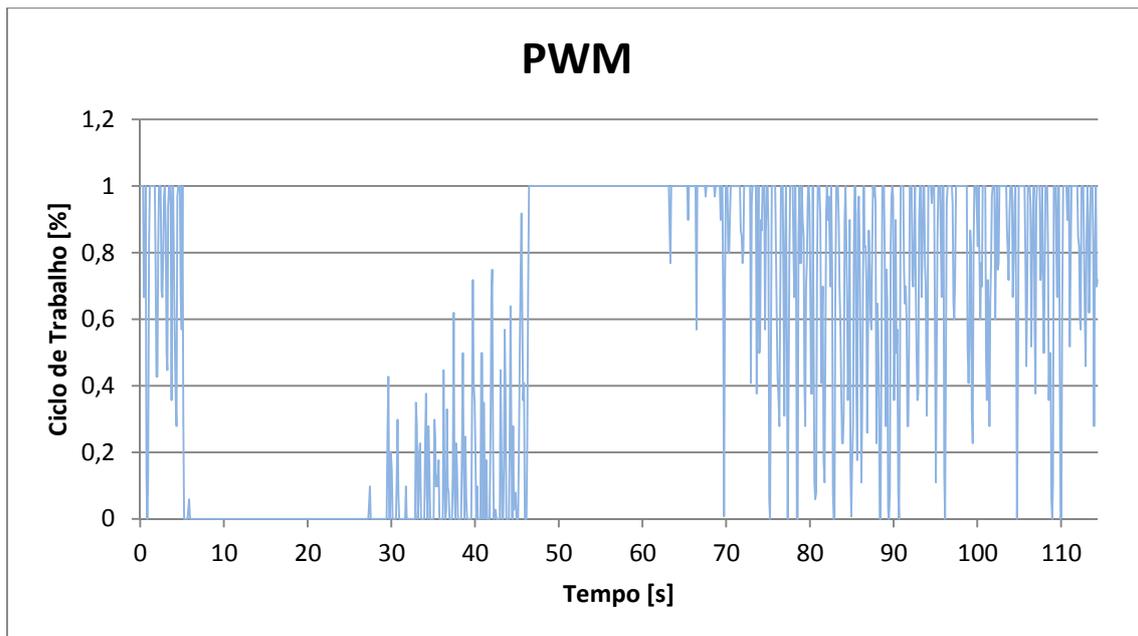


Figura 5.9 – PWM do 5° experimento.

Esse quinto experimento foi feito para entendermos melhor o funcionamento do controlador por meio da alteração de parâmetros além das constantes. Nesse caso, pode-se ver que como o controlador tem a capacidade de atuar de forma mais rápida e mesmo sem um fator integral, ele atua melhor e de forma mais eficiente na dinâmica do sistema, corrigindo o erro estacionário. No nosso projeto, por causa do medidor de vazão, o intervalo de tempo é de 1 segundo.

5.2 Controlador de temperatura e vazão

Foram realizados três experimentos com o controlador:

No primeiro, aplicou-se uma diferença ao mesmo tempo no controlador de temperatura e no controlador de vazão. A intenção foi visualizar o comportamento dinâmico do sistema quando fosse solicitado grande atuação nas duas variáveis a serem controladas.

No segundo e no terceiro deixou-se a temperatura constante e variou-se a vazão. No segundo diminuiu-se a vazão enquanto no terceiro aumentou-se. A intenção foi ver o quanto de alteração na vazão influenciaria a dinâmica da temperatura.

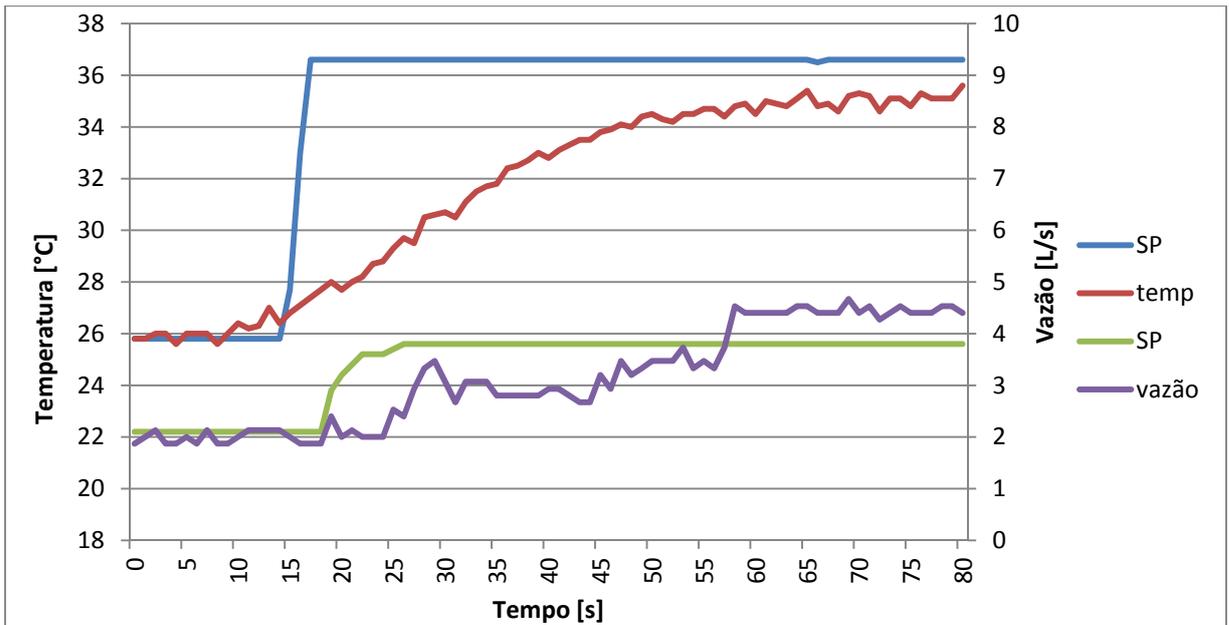


Figura 5.10 - Gráfico da dinâmica dos controladores do 1º experimento

A figura 5.10 mostra que os controladores reagiram da forma esperada. O controlador de temperatura não atingiu a temperatura do ponto de ajuste por falta de potência do projeto, pois como podemos ver pela figura 5.11, o controlador fornecia o máximo de potência que poderia. Enquanto o controlador de vazão trabalhou dentro do intervalo que foi projetado.

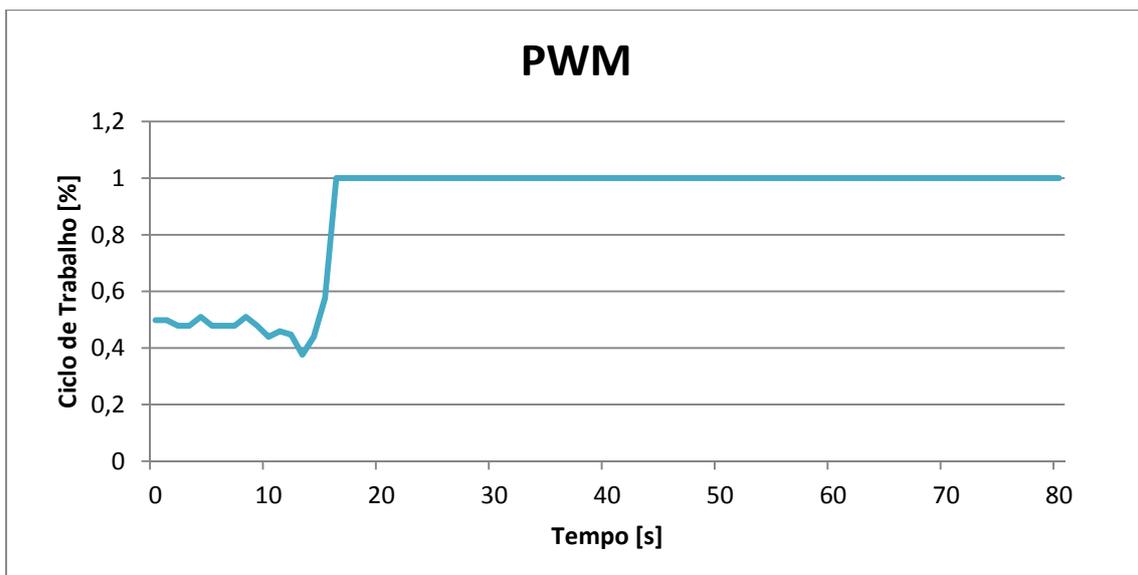


Figura 5.11 – PWM.

Nesses experimentos, pode-se ver que as temperaturas medidas se mantiveram constantes parecendo assim que a temperatura do sistema é independente da vazão. Porém, se analisarmos o gráfico da atuação do PWM sobre a temperatura, vemos que houve uma alteração na atuação da temperatura.

O que podemos concluir em relação aos experimentos dois e três é que a resposta do sistema de controle da temperatura é alta e a da vazão é lenta. Então por isso, é imperceptível o impacto na temperatura quando alteramos a vazão do sistema. O controlador corrige a diferença de temperatura causada pela alteração na vazão de forma rápida, antes mesmo que o usuário perceba.

As figuras 5.12 e 5.14 mostram a dinâmica da temperatura e vazão no segundo e terceiro experimento respectivamente e as figuras 5.13 e 5.15 mostram a atuação do PWM no segundo e terceiro experimento respectivamente.

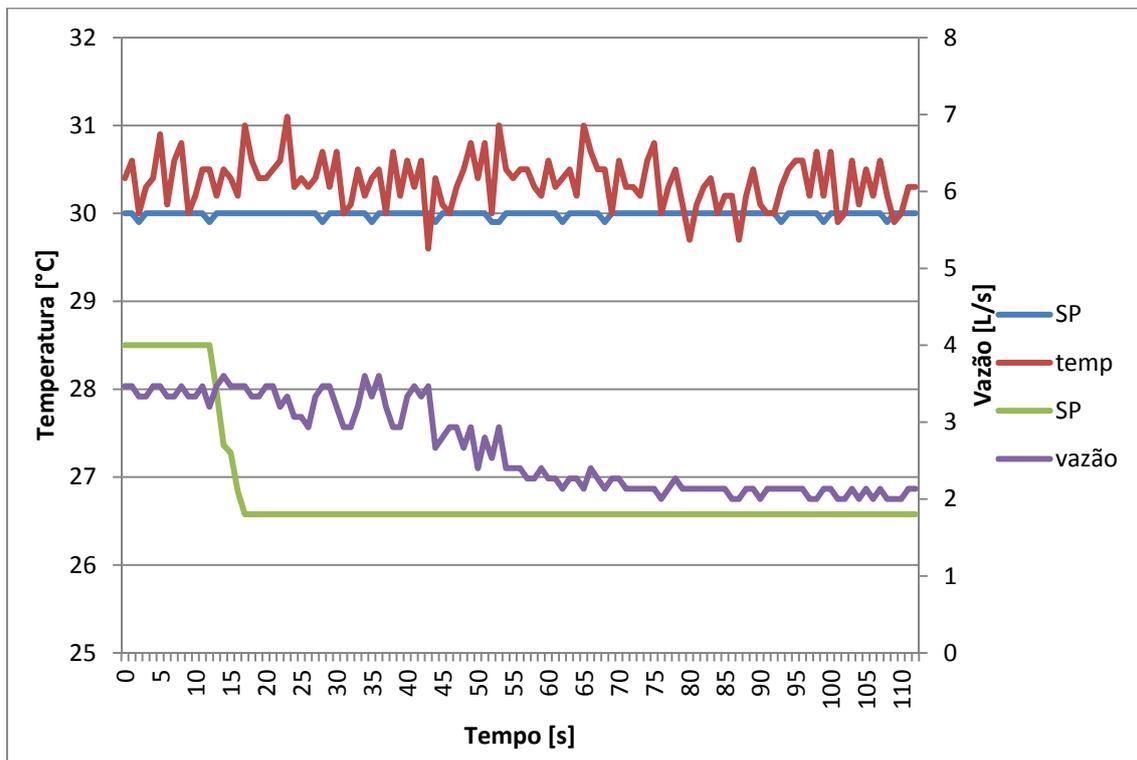


Figura 5.12 - Gráfico da dinâmica dos controladores do 2º experimento

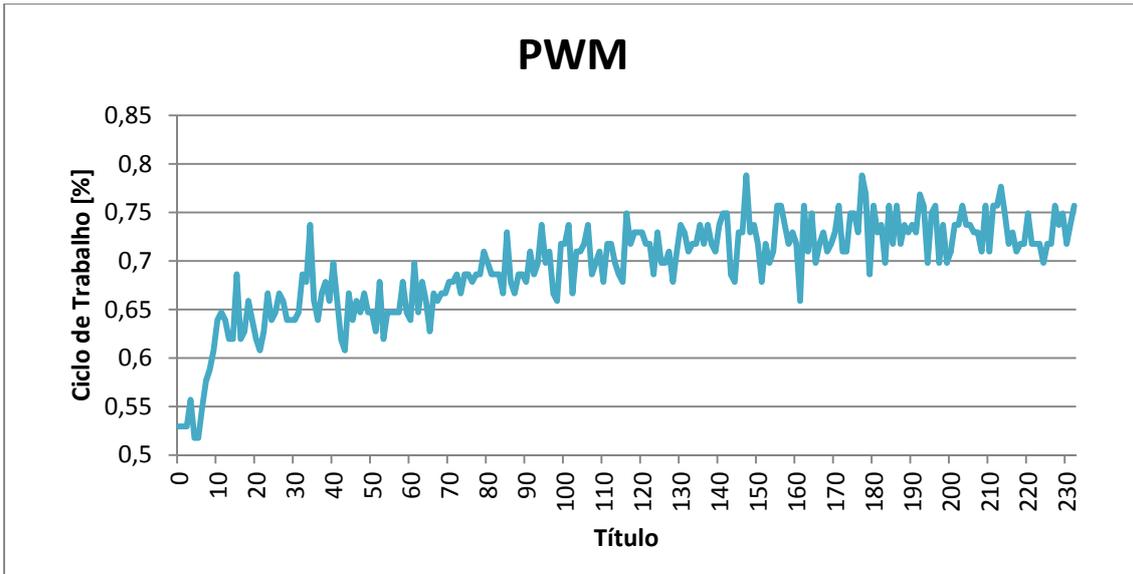


Figura 5.13 – PWM

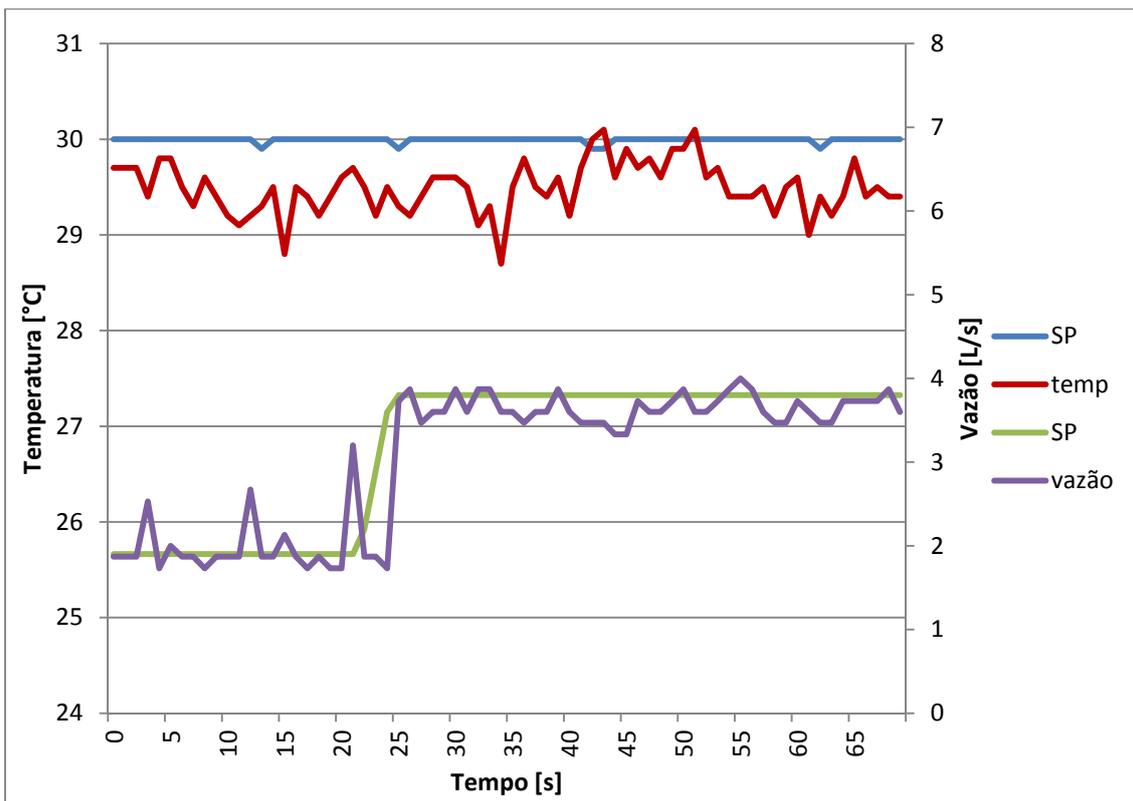


Figura 5.14 - Gráfico da dinâmica dos controladores do 3º experimento

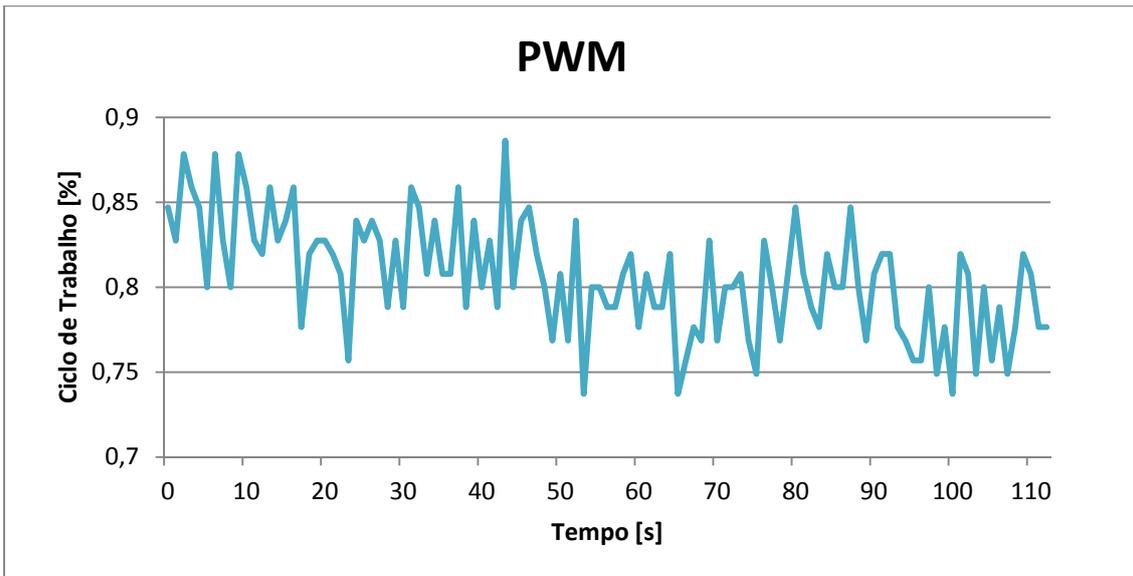


Figura 5.15 – PWM

Capítulo 6

Considerações finais

A implementação de um controlador realmente facilita e simplifica a vida do usuário. Um controlador bem projetado trás segurança, praticidade e conforto no dia a dia. Outra vantagem é a economia de tempo disponibilizando mais tempo para atividades que exigem menos trabalhos operacionais e mais analíticos.

Um controlador de temperatura em um chuveiro elétrico, por exemplo, é uma boa opção para locais onde não existe gás encanado. O gás é uma fonte de energia mais barata que a elétrica e o boiler permite regular a temperatura de forma mais precisa que os chuveiros elétricos convencionais. Com o controlador de temperatura no chuveiro elétrico, apesar de mais caro que os boilers, a definição da temperatura é mais simples e rápida. Evita que o usuário fique regulando a temperatura manualmente por meio de macetes ou que tome um banho muito frio ou muito quente, como acontece com os chuveiros elétricos convencionais. Isso acarreta em um tempo de banho menor do que o normal, diminuindo o consumo de água e de energia.

O controlador de vazão em um chuveiro é aconselhado para locais com vazões inconstantes ou com vazões muito altas. Da mesma forma, o usuário economiza tempo transferindo o controle da vazão para o próprio chuveiro. Ainda por cima é mais eficiente.

6.1 Conclusão

O controlador do chuveiro de temperatura e vazão inicialmente planejado foi concluído. As respostas e os gráficos gerados corresponderam à expectativa inicial do projeto. O objetivo de estudar e entender mais sobre controladores, a concepção e a visualização de um controlador físico foram alcançados com sucesso.

O que foi realizado:

Definição, adaptação e calibração do sensor de temperatura no chuveiro.

Definição, adaptação e calibração do medidor de vazão no chuveiro.

Sistema de atuação válvula motor com transmissão e suporte adaptados para o chuveiro.

Modelagem matemática do sistema. Implementação de média dos valores medidos para diminuir a imprecisão e volatilidade dos instrumentos de medição

Programação do controlador foi desenvolvida em C/C++.

6.2 Trabalhos futuros

O projeto ainda pode ter melhorias significativas aperfeiçoando-se individualmente cada componente do chuveiro.

Como propostas de trabalhos futuros destacam-se:

Na ênfase do controle da temperatura, ver a possibilidade de usar dois termopares para a medição da temperatura antes e depois do aquecimento da água. A medição da temperatura de entrada reduziria o peso dos distúrbios do sistema, diminuindo o fator integral do controlador. Por sua vez, um fator integral pequeno diminui o tempo de resposta e aumenta a estabilidade do sistema.

Outra melhoria poderia ser na modelagem da temperatura. Incluir a vazão nesse modelo traria mais eficiência ao chuveiro.

Na vazão, procurar uma solução para transformar o sinal em frequência para um sinal em tensão do medidor de vazão. Dessa forma, a medição se daria

instantaneamente possibilitando a instalação de um controlador mais robusto como o PID e também melhoraria o desempenho do controlador de temperatura.

Melhorar o desempenho do sistema de atuação motor-válvula. A implementação de um motor de passo daria mais eficiência ao modelo, pois se conseguiria saber a posição angular da válvula.

Referências Bibliográficas

[1]PID Controller, Wikipedia. Disponível em:

< http://en.wikipedia.org/wiki/PID_controller> Acessado em: 18 ago. 2013.

[2]Robocore. Disponível em:

< www.robocore.net> Acessado em: 18 ago. 2013.

[3]OGATA, Katsuhiko. *Engenharia de Controle Moderno*. 5. ed. São Paulo: Pearson Prentice Hall, 2010.

Anexo I

Desenvolvimento da programação do controlador

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 6, 5, 4, 3);

int caso = 0;

int pottemp = 5;          // Pino de entrada dos potenciômetros dos setpoints
int potvazao = 4;

float temp = 0;          // Setpoints (temp e vazão)
float vazao = 0;

const int NTC = 1;       // Pino do sensor de temperatura
volatile int NbTopsFan;  // measuring the rising edges of the signal
int hallsensor = 2;     // The pin location of the sensor

float ValorVazao = 0;    // Leitura da vazão do sistema
float errovazao = 0;    // Erro da vazão do sistema
float errovazao2 = 0;
float mediaerrovazao = 0;

float ValorTemp = 0;    // Leitura da temperatura do sistema

float errotemp = 0;     // Erros (temp)
float errotemp2 = 0;

float mediaerro = 0;

float errotemp_integral = 0;
float dertemp = 0;

float kpt = 50;         // Constantes do controlador
float kit = 0.2;

int pwm = 0;           // Controladores (Saídas)
int pinopwm = 10;
int saidatemp = 0;
int motor1 = 8;
int motor2 = 7;

float t = 1;           // Tempo entre as medições (à determinar)
// _____
void rpm ()           //This is the function that the interrupt calls
{
  NbTopsFan++;
}
// _____
void setup(){

  lcd.begin(16, 2);
  Serial.begin(9600);

  pinMode(hallsensor, INPUT); // Entradas e saídas
  pinMode(pinopwm, OUTPUT);
  pinMode(motor1, OUTPUT);

```

```

pinMode(motor2, OUTPUT);

TCCR1B = TCCR1B & 0b11111000 | 0x05;
attachInterrupt(0, rpm, RISING); //and the interrupt is attached
    }
// _____
void loop(){
// ----- Início da leitura da vazão -----
    NbTopsFan = 0; //Set NbTops to 0 ready for calculations
    sei(); //Enables interrupts
    delay (1000); //Wait 1 second
    cli(); //Disable interrupts
    ValorVazao = NbTopsFan; //(Pulse frequency x 60) / 7.5Q, = flow rate in L/hour
    ValorVazao = ValorVazao / 7.5;
// ----- Final da leitura da vazão -----

ValorTemp = map(analogRead(NTC), 508, 635, 260, 380); // Leitura da temperatura
ValorTemp = ValorTemp/10;

temp = map(analogRead(pottemp), 0, 1023, 200, 400); // Definição do SP da temperatura
temp = temp/10;
vazao = map(analogRead(potvazao), 0, 1023, 10, 80); // Definição do SP da vazão
vazao = vazao/10;

// Controlador: _____ COONTROLE DA TEMPERATURA _____
    errotemp = temp - ValorTemp; // Erro temp

mediaerro = (errotemp + errotemp2);

errotemp_integral = errotemp_integral + mediaerro*t; // Acumulação do erro, parte integral

saidatemp = kpt * mediaerro + kit * errotemp_integral;

pwm = map(saidatemp, -50, 50, 0, 255); // Controlador da temperatura

if (pwm > 255){
    pwm = 255;
}
if (pwm < 0){
    pwm = 0;
}

    analogWrite(pinopwm, pwm);
// _____

    errotemp2 = errotemp; // Guarda os erros anteriores para o
    // cálculo da média

// _____ CONTROLE DA VAZÃO _____

    errovazao = vazao - ValorVazao; // Erro vazao
    mediaerrovazao = (errovazao + errovazao2)/2;
    errovazao2 = errovazao;

```

```

    caso = 0;
    if ((mediaerrovazao < -0.3) && (mediaerrovazao > -0.8)){ // Controlador da vazão
p/ um sentido
        digitalWrite(motor2,HIGH);
        delay(300);
        caso = 1;
        digitalWrite(motor2,LOW);
        }

    if ((mediaerrovazao > 0.3) && (mediaerrovazao < 0.8)){ // Controlador da vazão
p/ o outro sentido
        digitalWrite(motor1,HIGH);
        delay(300);
        caso = 2;
        digitalWrite(motor1,LOW);
        }
    if (mediaerrovazao < -0.8){ // Controlador da vazão p/ um sentido
        digitalWrite(motor2,HIGH);
        delay(700);
        caso = 3;
        digitalWrite(motor2,LOW);
        }

    if (mediaerrovazao > 0.8){ // Controlador da vazão p/ o outro
sentido
        digitalWrite(motor1,HIGH);
        delay(700);
        caso = 4;
        digitalWrite(motor1,LOW);
        }

// FIM DO CONTROLADOR_____

// LCD _____

// 0123456789012345
// T: 30,5 30,5 C
// V: 1,4 1,6 L/m
// _____

lcd.setCursor(0,0);
lcd.print("T:");
lcd.setCursor(0,1);
lcd.print("V:");
lcd.setCursor(14,0);
lcd.print("C");
lcd.setCursor(13,1);
lcd.print("L/m");
lcd.setCursor(3,0);
lcd.print(temp);
lcd.setCursor(7,0);
lcd.print(" ");
lcd.setCursor(3,1);
lcd.print(vazao);
lcd.setCursor(7,1);

```

```
lcd.print(" ");  
lcd.setCursor(9,1);  
lcd.print(ValorVazao);  
lcd.setCursor(8,0);  
lcd.print(ValorTemp);  
lcd.setCursor(13,0);  
lcd.print(" ");
```

```
// FIM DO LCD_____
```

```
} //void loop
```