

**REDE DOCTUM DE ENSINO
INSTITUTO TECNOLÓGICO DE CARATINGA
CURSO SUPERIOR DE ENGENHARIA ELÉTRICA**

**CONTROLE E MONITORAMENTO DE NÍVEL EM TANQUES RESERVATÓRIOS DE
LÍQUIDOS**

ANDRÉ DE SOUZA PEREIRA

Trabalho de Conclusão de Curso

Caratinga/MG

2016

ANDRÉ DE SOUZA PEREIRA

**CONTROLE E MONITORAMENTO DE NÍVEL EM TANQUES RESERVATÓRIOS DE
LÍQUIDOS**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Engenharia Elétrica do Instituto Tecnológico de Caratinga da DOCTUM Caratinga como requisito parcial para obtenção do Grau de Bacharel em Engenharia Elétrica.

Professor Orientador: Daniel Mageste Butters.

Caratinga/MG

2016

TERMO DE APROVAÇÃO

TÍTULO DO TRABALHO

CONTROLE E MONITORAMENTO DE NÍVEL EM TANQUES RESERVATÓRIOS DE LÍQUIDOS

Nome completo do aluno: ANDRÉ DE SOUZA PEREIRA


Este Trabalho de Conclusão de Curso foi apresentado perante a Banca de Avaliação composta pelos professores Daniel Mageste Butters, Joildo Fernandes Costa Junior e Vinicius Murilo Lima Rodrigues, às 19:35 horas do dia 12 de dezembro de 2016, como requisito parcial para a obtenção do título de bacharel em Engenharia Elétrica. Após a avaliação de cada professor e discussão, a Banca Avaliadora considerou o trabalho: APROVADO (aprovado ou não aprovado), com a qualificação: Ótima (Excelente, Ótima, Bom, Satisfatório ou Insatisfatório).

Trabalho indicado para publicação: SIM () NÃO

Caratinga, 12 de dezembro de 2016



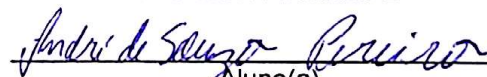
Professor Orientador e Presidente da Banca



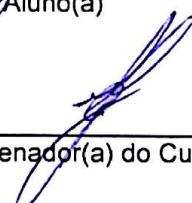
Professor Avaliador 1



Professor Avaliador 2



Aluno(a)



Coordenador(a) do Curso

À Deus por todas as bênçãos desta vida.....

AGRADECIMENTOS

A Deus por ter me dado saúde e força para superar as dificuldades.

A esta universidade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior.

Ao meu orientador Daniel Mageste Butters, pelo suporte, orientação e ajuda em seu tempo, pelas suas correções e incentivos.

Ao meu pai José de Souza Pereira, a minha mãe Ana Maria de Jesus Souza, a minha esposa Tatiane Lacerda Lopes e a minha irmã Josiane de Souza Pereira, pelo amor, incentivo e apoio incondicional.

A Rose Helena de Almeida Miranda Lacerda que se dispôs a me ajudar com orientação e correções.

Aos meus colegas de classe que estiveram presentes nos momentos alegres e difíceis ao longo do percurso deste curso.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

“Insanidade é continuar fazendo sempre a mesma coisa e esperar resultados diferentes”

(ALBERT EINSTEIN)

PEREIRA, André de Souza. **Controle e Monitoramento de Nível em Tanques Reservatórios de Líquidos**. Caratinga, 2016. Trabalho de Conclusão de Curso Curso Superior de Engenharia Elétrica. Instituto Tecnológico de Caratinga, Rede DOCTUM, Caratinga, 2016.

RESUMO

Na indústria, na agricultura, no abastecimento de água urbano e nas diversas outras áreas, desejamos controlar com precisão e rapidez o nível de líquido em reservatórios. Na maioria das vezes estes possuem uma demanda de vazão variável, que pode ocorrer na chegada do líquido que abastece o reservatório ou na saída do reservatório, durante a utilização de seu conteúdo, tornando-se então imprescindível o estudo, dimensionamento e implementação de um controlador.

Este estudo pretende realizar análise e implantação de um controlador PID, gerenciado por um Arduino em uma planta exemplo, utilizando recursos com alta disponibilidade, confiabilidade e baixo custo, para obter resultados satisfatórios, prezando sempre pela qualidade e confiança no sistema geral.

Em sistemas de controle, devemos sempre estar em constante monitoramento. Para isso existem os softwares supervisórios para supervisionar, monitorar e controlar as variáveis de um sistema, por meio de uma tela configurada em computadores e/ou em dispositivos com recursos semelhantes. Com a finalidade de acompanhar processos e intervir em ações de processos, tanto localmente quanto remotamente. No sistema estudado, é configurado uma tela desenvolvida no Eclipse Scada, esta tela além de monitorar os processos ela é capaz de ajustar o valor de setpoint (nível desejado), e dar permissão para que o mesmo ajuste possa ocorrer localmente através de um botão regulável.

Palavras-chave: Controle. Nível. PID. Arduino. Monitoramento. Supervisórios.

PEREIRA, André de Souza. **Controle e Monitoramento de Nível em Tanques Reservatórios de Líquidos**. Caratinga, 2016. Trabalho de Conclusão de Curso Curso Superior de Engenharia Elétrica. Instituto Tecnológico de Caratinga, Rede DOCTUM, Caratinga, 2016.

ABSTRACT

In industry, agricultural, urban water supply and in several other areas, it's desired to precisely control the level of liquid in reservoirs accurately and quirky. Most of the times these reservoirs have a variable flow demand, this can occur in the arrival of liquid that supplies the reservoir or in the exit, during the use of its contents, making it then essential to study, design and implement a controller.

This study intends to perform analysis and implementation of a PID controller managed by an Arduino in a sketch map, using resources with high availability, reliability and low cost, obtaining satisfactory results and always valuing the quality and reliability of the general system.

In control systems, it is necessary to be in continuous monitoring, for that reason there are provisional software to supervise, monitor and control the variables of a system, through a screen configured by computers and/or devices with similar resources, for the purpose of monitor processes and intervene in actions both locally and remotely. In this system, a screen developed in the Scada Eclipse is configured, which in addition to monitoring the process, it is able to adjust the setpoint value (desired level) and allow a in order to the adjustment can take place locally through an adjustable button.

Key-words: Control. Level. PID. Arduino. Monitoring. Supervisory.

LISTA DE ILUSTRAÇÕES

Figura 1	– Simbologia transistores.	17
Figura 2	– Transistores.	17
Figura 3	– Resistor e código de cores.	18
Figura 4	– Esquema de um Diodo.	19
Figura 5	– Microcontrolador Atmega 2560.	19
Figura 6	– CLP Weg. PLC 300.	20
Figura 7	– Interface programação label TPW03 Weg.	20
Figura 8	– Arduino Mega2560.	21
Figura 9	– Interface Arduino programação e envio da mesma.	22
Figura 10	– Monitor serial Arduino.	22
Figura 11	– Diagrama de bloco Função de Transferência.	26
Figura 12	– Diagrama de blocos sistema malha aberta.	27
Figura 13	– Diagrama de blocos sistema Malha Fechada.	27
Figura 14	– Diagrama de blocos sistema realimentado com controlador.	28
Figura 15	– Interface Software MATLAB.	33
Figura 16	– Interface simulação MATLAB Simulink.	33
Figura 17	– Kit controle de nível Bit9.	35
Figura 18	– Gráfico tensão x vazão.	37
Figura 19	– Tabela erro regime permanente.	40
Figura 20	– Diagrama de blocos do sistema.	40
Figura 21	– Ajustando Controlador PI.	41
Figura 22	– Valores do Controlador PI.	42
Figura 23	– Resposta do sistema.	43
Figura 24	– Diagrama de blocos modificado do sistema.	43
Figura 25	– Resposta do sistema 2.	44
Figura 26	– Sensor ultrassônico HC-SR04.	45
Figura 27	– Definindo valor Proporcional e Integral.	47
Figura 28	– Definindo valor controlador PI.	47
Figura 29	– Circuito alimentação do Atuador.	48
Figura 30	– Tela de supervisão e monitoramento.	49
Figura 31	– Display Ajuste Local.	50
Figura 32	– Display Ajuste Remoto.	50
Figura 33	– Início de teste.	51
Figura 34	– Final de teste.	52
Figura 35	– Teste em regime permanente.	52
Figura 36	– Tabela de Laplace.	65

LISTA DE TABELAS

Tabela 1 – Medidas do Reservatório	36
Tabela 2 – Tensão X Vazão	37
Tabela 3 – Especificações Sensor ultrassônico HC-SR04.	45

LISTA DE ABREVIATURAS E SIGLAS

V	Volts
kp	Ganho Proporcional
ki	Ganho Integral
kd	Ganho Derivativo
PI	proporcional-integral
PID	proporcional-integral-derivativo
MW	Megawatt

LISTA DE SÍMBOLOS

*	Multiplicação em operações matemáticas
%	Por cento
μ	micro

SUMÁRIO

1 INTRODUÇÃO	14
2 REVISÃO DA LITERATURA	16
2.1 Instrumentação	16
2.1.1 Circuito Elétrico	16
2.1.1.1 Transistor	16
2.1.1.2 Resistor	18
2.1.1.3 Diodo	18
2.1.2 Microcontrolador	19
2.1.3 Controlador Lógico Programável	19
2.1.4 Arduino	21
2.1.5 CLP Versus Arduino	24
2.1.6 Sensor Ultrassônico	24
2.2 Teoria de Controle	25
2.2.1 Sistemas Lineares	25
2.2.2 Sistemas de Primeira Ordem	25
2.2.3 Sistemas de Segunda Ordem	25
2.2.4 Sinais de Entrada e Saída	25
2.2.5 Análise e modelagem matemática	25
2.2.6 Função de Transferência	26
2.2.7 Malha Aberta	26
2.2.8 Malha Fechada	27
2.2.9 Transformada de Laplace	28
2.2.10 Controle	29
2.2.10.1 Controle on-off	29
2.2.10.2 Controle Proporcional (P)	30
2.2.10.3 Controle Integral (I)	30
2.2.10.4 Controle Derivativo (D)	31
2.2.10.5 Controle Proporcional Integral Derivativo (PID)	32
2.3 Softwares	32
2.3.1 MATLAB	32
2.3.2 Programas Scada	33
2.3.2.1 Elipse Scada	34
2.3.3 Modbus	34
3 PROCEDIMENTOS METODOLÓGICOS	35
3.1 Planta de Controle	35

3.1.1 Kit para controle de nível da bit9	35
3.2 Análise da planta	36
3.3 Modelagem Matemática	37
3.4 Simulação em software	40
3.4.1 Ajuste Fino da Simulação	43
3.5 Implantação do Controle e monitoramento de nível	44
3.6 Programação Arduino	45
3.7 Circuito eletrônico	47
3.8 Monitoramento	48
4 RESULTADOS E APLICAÇÕES	51
4.1 Resultados	51
4.2 Aplicações	53
4.2.1 Saneamento urbano	53
4.2.2 Pequena Central Hidrelétrica	53
5 CONCLUSÃO	55
6 REFERÊNCIAS BIBLIOGRÁFICAS	56
APÊNDICE A Apêndice	58
ANEXO A Tabela de Laplace	65

1 INTRODUÇÃO

Sistemas de controle surgem com a finalidade de automatizar e agilizar processos sem a intervenção humana, em um sistema de nível por exemplo sem automação seria necessário que uma pessoa ficasse responsável por aumentar e diminuir a vazão do reservatório de acordo com a necessidade. Já no controle automático basta definir um valor de Setpoint que o sistema irá controlar automaticamente de acordo com o desejado.

O CLP é parte essencial na indústria para controle de processos por ser um equipamento robusto e com grande capacidade de operações. Tem um custo elevado, tornando a implantação de controle mais cara. O Arduino é uma plataforma para Microcontrolador capaz de realizar controle de alguns processos sem deixar a desejar quanto aos CLPs, além de ter um custo bem mais acessível, se torna mais atraente e propício para o investimento em diversos tipos de projetos. Vale ressaltar que o Arduino é uma plataforma que facilita a interação com o Microcontrolador, não sendo projetado para utilização contínua, seu propósito se dar para facilitar o desenvolvimento de projetos, quando prontos e forem colocados em práticas deve-se desenvolver uma placa eletrônica que contenha o Microcontrolador e suas conexões necessárias para o funcionamento do projeto.

Para se pôr à prova a capacidade e facilidade de implantação de controle gerenciado pelo Arduino, será usado uma planta exemplo onde haverá simulação de um cenário complexo, que venha a exigir um controle rápido e preciso. Em cima dessa planta de estudo, serão realizadas simulações práticas e via software, para que sejam comparados os resultados obtidos em ambos os testes.

O projeto terá como objetivo controlar o nível de um determinado reservatório, fazendo com que o atuador que no caso será uma bomba d'água possa drenar o líquido em seu conteúdo sempre que o mesmo esteja acima do Setpoint e durante a vazão de abastecimento o mesmo possa ser drenado em proporção sempre deixando o nível estável, não ficando abaixo nem acima do valor de Setpoint.

Uma das partes essenciais em sistemas controlados, é poder acompanhar o andamento e estado dos processos envolvidos, afim de se obter os resultados para saber em quais situações encontra-se o sistema tornando necessária a criação de uma interface que informará ao usuário em tempo real todas as ocorrências no sistema controlado. Além de fornecer informações de monitoramento do sistema, a tela permitira que o usuário possa alterar o valor de Setpoint do sistema e definir se o mesmo será ajustado localmente por auxílio de um botão ou remotamente pela tela.

Uma das principais vantagens do projeto que será abordado é poder alterar o valor de Setpoint tanto remotamente pela tela de gerencia e controle quanto localmente pelo simples

apertar de um botão.

Contudo espera-se no final do trabalho esclarecer dúvidas remanescentes sobre o emprego do Arduino em sistemas de controle, junto a técnica de controle PID e o monitoramento através de software. Facilitando o desenvolvimento de projetos e estudos que se baseiam em poder automatizar e controlar sistemas existentes ou em implantação.

2 REVISÃO DA LITERATURA

2.1 INSTRUMENTAÇÃO

Em sistemas de controle, onde se trabalha com malha fechada, sempre haverá a necessidade de se ter equipamentos de sensoriamento, onde o controlador utilizará informações destes componentes para analisar e identificar a existência de erro e/ou estabilização do sistema. Além destes equipamentos, os controladores necessitam do desenvolvimento de circuitos responsáveis por realizar acionamento direto ou indireto dos atuadores.

2.1.1 Circuito Elétrico

Circuitos Elétricos são circuitos fechados onde o fim é o início, estes circuitos são alimentados por uma fonte de energia que pode ser uma tomada, bateria ou pilhas. Através de um condutor, a corrente elétrica percorre todo o circuito a fim de alimentar e suprir a necessidade de uma carga. No percorrer do circuito pode haver a existência de diversos componentes com aplicações distintas.

"Um circuito elétrico, ou rede elétrica, é uma coleção de elementos interconectados de maneira específica."(JOHNSON et al, 1994, p. 2).

"O termo condutor é aplicado a qualquer material que sustenta um grande fluxo de carga ao se aplicar, através de seus terminais, uma fonte de tensão de amplitude limitada". (BOYLESTAD e NASHELSKY, 2004, p. 3).

2.1.1.1 Transistor

O Transistor é um componente eletrônico que possui variadas funções em um circuito, sendo a mais utilizada é a de amplificar.

O Transistor possui três terminais, onde temos base, coletor e emissor, para a corrente circular entre elas é preciso alimentar a base, conforme a tensão é aplicada nela, o transistor altera a resistência existente entre o pino emissor e coletor, conseqüentemente variando o fluxo de passagem de corrente elétrica.

"Todos os amplificadores (dispositivos que aumentam a tensão, corrente ou nível de potência) possuem no mínimo três terminais e um deles controla o fluxo entre os outros dois". (BOYLESTAD e NASHELSKY, 2004, p. 6).

A seguir veremos a estrutura e símbolo de um Transistor representado pela figura 1 e logo em seguida alguns modelos de Transistores bipolares na figura 2.

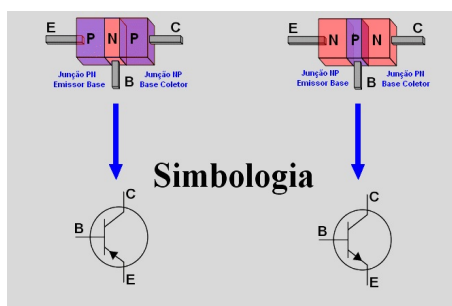


Figura 1: Simbologia transistores.

Fonte: Electronica-pt.

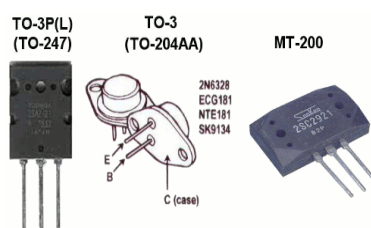


Figura 2: Transistores.

Fonte: Electronica-pt.

Os Transistores se dividem em dois grupos, eles podem ser bipolares ou de efeito de campo. No grupo dos bipolares existem dois tipos, o do tipo NPN ou PNP. Em Transistores do tipo NPN a corrente circula do coletor para o emissor e no PNP, do emissor para o coletor.

Os semicondutores do tipo N durante seu processo de dopagem recebem elementos com cinco elétrons na camada de valência, ficando então com elétrons livres. Já os do tipo P recebem dopagem com três elétrons o que ocasiona lacunas em sua estrutura molecular. Então basicamente semicondutores do tipo N possuem excesso de elétrons enquanto os do tipo P possuem lacunas em suas estruturas.

"O material do tipo n e do tipo p são formados pela adição de um número predeterminado de átomos de impureza em uma base de germânio ou silício. O material do tipo n é criado com a introdução dos elementos de impureza que têm cinco elétrons de valência (pentavalente), como o antimônio, o arsênio e o fósforo.

[...] O material do tipo p é obtido dopando-se um cristal de germânio ou silício puro com átomos de impureza que possuem três elétrons de valência. Os elementos mais comumente utilizados para esse propósito são o boro, o gálio e o índio.

[...] O espaço vazio resultante é chamado de lacuna e é representado por um pequeno círculo ou sinal positivo devido à ausência de uma carga negativa". (BOYLESTAD e NASHELSKY, 2004, p. 6).

2.1.1.2 Resistor

O Resistor é um componente muito empregado em circuitos eletrônicos com a finalidade de se limitar ou reduzir a corrente elétrica, esta limitação ou redução ocorre devido ao efeito Joule, onde o resistor dissipa energia elétrica em forma de energia térmica.

A unidade do Resistor é dada em ohms, quanto maior for seu valor maior será sua resistência e menor será o fluxo de corrente elétrica. Sua empregabilidade pode ser em série ou paralelo ao circuito, isso será de acordo com a necessidade e sua função no circuito.

Para determinar o valor de resistência de um Resistor pode ser utilizado um multímetro com a específica configuração para medição de resistividade em resistores, através de seu código de cores ou com a equação 2.1 que representa a primeira lei de Ohm, onde o valor de resistência será igual a diferença de potencial aplicada no Resistor dividido pela corrente elétrica que o atravessa. (JÚNIOR, 2016).

Na figura 3 é observado um Resistor com suas cores junto a uma tabela com indicação da representação dos valores das cores.

$$R = \frac{U}{I} \tag{2.1}$$

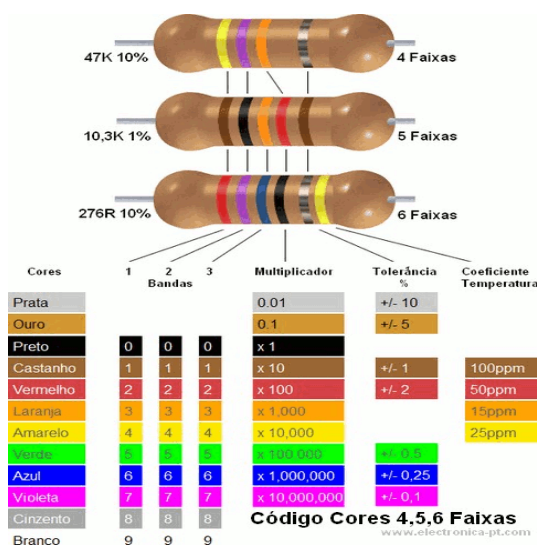


Figura 3: Resistor e código de cores.

Fonte: Electronica-pt.

2.1.1.3 Diodo

O Diodo é um componente semicondutor, em sua estrutura possui uma junção N e outra P. Existem diversos tipos de Diodos, cada um com sua característica base. Em sua natureza básica o Diodo consiste em permitir a passagem de corrente apenas por um sentido, evitando

então que correntes ou sinais elétricos circulem em sentidos indevidos. Na figura 4 temos um Diodo junto ao seu simbolo e composição estrutural, que seria a junção N e P.

"As características de um diodo ideal são as de uma chave que teria a capacidade de conduzir corrente em um único sentido". (BOYLESTAD e NASHELSKY, 2004, p. 1).

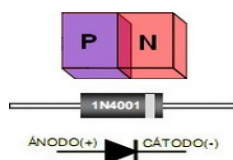


Figura 4: Esquema de um Diodo.

Fonte: Electronica-pt.

2.1.2 Microcontrolador

Microcontrolador é um minicomputador dentro de um único circuito integrado (CI), este único CI possui processador, memória e periféricos de entrada e saída. Na figura 5 pode ser visualizado o CI microcontrolador da Atmel modelo Atmega 2560.



Figura 5: Microcontrolador Atmega 2560.

Fonte: Nenotech.

Devido ao tamanho e capacidade ilimitada de operações, os microcontroladores são cada vez mais utilizados em diversas áreas principalmente industrias, onde se tem a grande necessidade de controlar e automatizar processos.

Com o grande crescimento da necessidade de facilitar o emprego destes Microcontroladores, surgem plataformas para seu acoplamento. Com estas plataformas, tornou-se mais fácil a configuração e comunicação com os Microcontroladores.(PALMIERE, 2016).

2.1.3 Controlador Lógico Programável

O Controlador Lógico Programável (CLP) surgiu para o gerenciamento e controle de máquinas e processos. Foi desenvolvida uma linguagem de programação unificada denominada

de Label pelas marcas para facilitar a lógica de programação dos CLPs. Assim, os programadores não teriam que aprender diversas linguagens de programação, apenas se acostumarem com a interface de programação de cada empresa fornecedora do CLP, ressaltando que cada interface tem suas particularidades, umas com mais e outras com menos recursos, porém a lógica e maneira de programar prevalece com a mesma base.

A função básica do CLP seria acionar e interromper funcionamento de máquinas, isso tudo é determinado na programação. Ele possui interfaces físicas para entrada de sinais elétricos e através disso, recebe comandos e informações para serem processadas em sua programação e de acordo com esta, emitir sinais elétricos em seus terminais de saída que irão acionar um contator ou relé que irá ligar algum equipamento como um motor, uma lâmpada e etc. Graças a essas entradas e saídas fica fácil de se acoplar módulos criados para a expansão de seus recursos como transmissão de informação, melhoramento de hardware dentre outras funcionalidades. (PALMIERE, 2016).

Na figura 6 temos um CLP da empresa Weg e na figura 7 a tela do programa da Weg disponibilizado por ela para a programação e transmissão da mesma para o CLP.



Figura 6: CLP Weg. PLC 300.

Fonte: Weg.

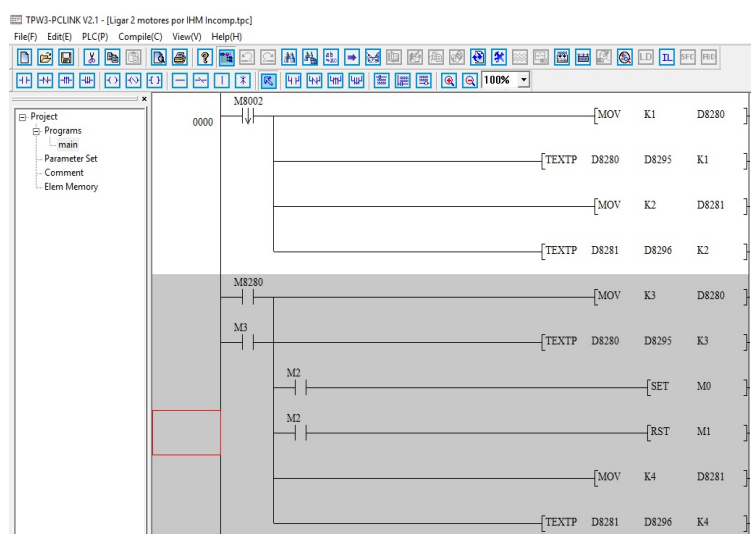


Figura 7: Interface programação label TPW03 Weg.

Fonte: Autoral.

2.1.4 Arduino

O Arduino é uma plataforma de prototipagem eletrônica de hardware livre, criada com a intenção de ser uma ferramenta prática com baixo custo e de fácil utilização por projetistas, professores, estudantes e curiosos com intuito de realizarem variados tipos de projetos, estudos e trabalhos com baixo custo de investimento.

“Uma placa de Arduino é, na verdade, uma placa de desenvolvimento baseada em microcontrolador, cujo projeto é independente e de hardware aberto (open source) “. (MONK, 2013, p. 7).

Por se tratar de hardware livre, são encontrados diversos tipos de Arduinos com diversas variações, mas sua base permanece, que são os Microcontroladores Atmega, a linguagem de programação em C, interface de desenvolvimento e os complementos. A placa por se só tem recursos limitados, mas devido ela trabalhar com múltiplas entradas e saídas de sinais tanto analógicos como digitais, fica fácil de implementar equipamentos e circuitos para melhorar e expandir o desempenho e a capacidade de projetos. Existem vários tipos de complementos montados chamados de Shields, onde estes são acoplados em cima da placa e permitem novas acoplagens, assim como também os complementos normais como sensores que são interligados por fios, circuitos eletrônicos ou outros meios que permitem a passagem de sinais elétricos. Na figura 8 observamos uma placa Arduino do modelo Mega 2560.

“Originalmente, o Arduino foi desenvolvido como recurso auxiliar no ensino dos estudantes. Mais adiante (em 2005), ele foi desenvolvido comercialmente por Massimo Banzil e David Cuartielles. Desde então ele se tornou um produto extremamente bem-sucedido junto a fabricantes, estudantes e artistas, devido à sua facilidade de uso e durabilidade”. (MONK, 2013, p. 10).

“Uma outra razão para o sucesso do Arduino é que ele não se limita a placas com microcontrolador. Há um número enorme de placas acessórias (denominadas shields) compatíveis com o Arduino”. (MONK, 2013, p. 11).



Figura 8: Arduino Mega2560.

Fonte: Arduino.cc.

A programação do Arduino tem como base a linguagem C. Este tipo de linguagem é considerado de nível médio devido à combinação de caracteres de baixo e alto nível, o que lhe permite criar programas simples e robustos. Para realizar esta programação e carregá-la para dentro do Microcontrolador, a plataforma Arduino tem um ambiente de desenvolvimento integrado (IDE) que permite a digitação e verificação do código em um editor de texto próprio. Esta Interface IDE ainda permite outros recursos como carregar a programação para dentro do Microcontrolador, incluir outros códigos de programação em forma de bibliotecas e a capacidade de monitorar e inserir valores através de uma ferramenta chamada monitor serial. A figura 9 mostra a IDE do Arduino, e a figura 10 mostrando a ferramenta monitor serial.

“A linguagem de programação usada para programar Arduinos é a linguagem C”. (MONK, 2013, p. 27).

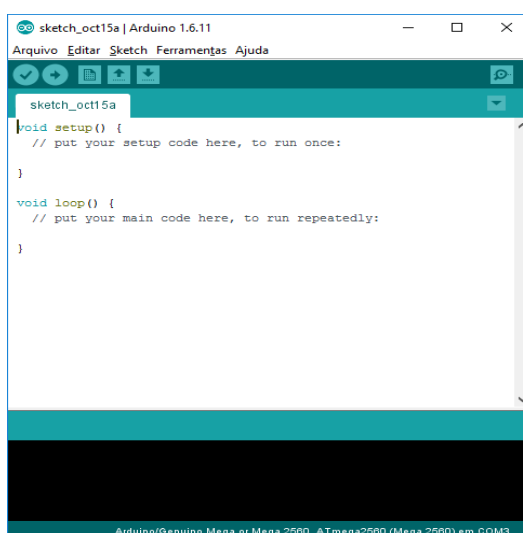


Figura 9: Interface Arduino programação e envio da mesma.

Fonte: Autoral.

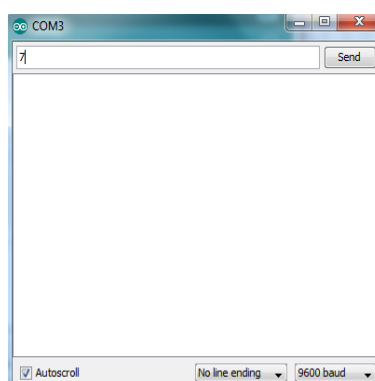


Figura 10: Monitor serial Arduino.

Fonte: Autoral.

Apesar da plataforma Arduino possuir múltiplas entradas e saídas de sinais, estas podem possuir recursos diferenciados, estas portas estão divididas em cinco grupos, que são eles:

Energia, neste grupo tem duas saídas de tensão que possuem valor de 3,3V e 5V. Possui também o GND que é entrada e saída simultaneamente, outra porta que faz parte deste grupo é a de alimentação da placa, que pode receber valor de tensão entre 7V e 12V para seu funcionamento.

“Ele recebe qualquer tensão (entre 7V e 12V) que esteja sendo fornecida pelo conector de alimentação e a converte em uma tensão constante de 5V”. (MONK, 2013, p. 7).

Entradas analógicas, neste grupo as portas recebem a descrição "A" junto ao número da porta, estas são responsáveis pelas entradas de informações analógicas. O sinal elétrico que elas captam podem conter informações de até 10 bits e variar um valor de tensão entre 0V e 5V. Caso a tensão tenha um valor superior poderá ocasionar danos físicos ao equipamento e através da percepção e análise da informação dessa entrada pode-se saber se um determinado equipamento está operando com potência entre 0% e 100%.

“Os seis pinos indicados como Analog In, de A0 a A5, podem ser usados para medir a tensão que está sendo aplicada a cada pino, de modo que seus valores podem ser usados em um programa (sketch)”. (MONK, 2013, p. 8).

Comunicação, neste grupo as portas recebem a descrição "TX" ou "RX" junto ao número da porta, estas são responsáveis por realizar transmissão e recepção de dados por meios de comunicação como ethernet e bluetooth.

“Os primeiros dois pinos (0 e 1), também denominados RX e TX, são para recepção e transmissão. Essas conexões estão reservadas para uso na comunicação”. (MONK, 2013, p. 9).

PWM, neste grupo em específico é responsável pela saída de sinais elétricos variáveis e recebe este nome devido a função de PWM que ele executa. Esta função se baseia em enviar sinais analógicos ter uma variação de 0V a 5V que é a tensão máxima de saída do Arduino. Por trabalhar em tensão contínua a leitura do sinal, não conseguiria interpretar mudança na frequência de valores em determinadas operações como em potência por exemplo. Dependendo do projeto o atuador nem sempre irá trabalhar com sua potência em 100%, então com variação no pulso entre 0V e 5V o Arduino consegue variar a potência do atuador entre 0% e 100%. (SOUZA, 2014).

Digital, neste grupo estão as portas que trabalham com informação digital, elas possuem o estado de ligado ou desligado, tanto emitem como recebem informações lembrando que sua análise e operação se dará penas em ligado ou desligado portanto qualquer uma das portas

pertencentes aos outros grupos com exceção do grupo de energia, possuem a mesma função. Basicamente através dela se consegue ligar ou desligar um atuador e saber se algum equipamento está ligado ou desligado.

“[...] Os pinos denominados Digital, de 0 a 13. Eles podem ser usados como entradas ou como saídas”. (MONK, 2013, p. 9).

2.1.5 CLP Versus Arduino

Criados com propostas diferentes, mas com características semelhantes, existem grandes diferenças entre estas plataformas.

O CLP foi desenvolvido a fim de controlar diversas máquinas e processos simultaneamente, o que o levou a ser um equipamento robusto com grande capacidade e alto custo para aquisição. Já o Arduino surgiu com a proposta de ser um equipamento para simulações e testes tendo um valor bem mais acessível, porém no desenvolver de tarefas e projetos, se percebe que ele é uma ferramenta com grande capacidade e eficiência, o que fez com que sua utilização e emprego aumentasse consideravelmente tanto em laboratórios quanto em uso prático em residências e diversos setores da indústria.

2.1.6 Sensor Ultrassônico

Sensor Ultrassônico é um equipamento utilizado para a identificação de objetos e sua distância, devido a sua velocidade e precisão o sensor se torna confiável e muito empregado na indústria e em projetos que necessitam desses tipos de informações como identificação de objeto e distância.

Seu funcionamento se baseia na reflexão de onda, onde o seu módulo emissor chamado de “trigger” envia um sinal em forma de onda senoidal. Quando se encontra com um objeto, esta onda é refletida sendo captada pelo módulo receptor denominado “echo”. Quando o módulo capta a informação, o gerenciador do sensor efetua análise do tempo de resposta, através desse tempo poderá encontrar a distância do objeto pela equação 2.2.

$$Distance = \frac{temporesposta * velocidade\ do\ som}{2} \quad (2.2)$$

A divisão por dois ocorre devido a onda ir e voltar no objeto, ou seja, ela percorre o caminho duas vezes. A velocidade de propagação do som varia de acordo com o meio físico onde é transportada e a temperatura ambiente. Com base nisso, a velocidade adotada para o cálculo de distância é de 340m/s. Respeitando as normas do sistema internacional de medidas, o tempo de resposta é dado em segundos e como a velocidade do som está em m/s, logo a resposta da distância será dada em metros. (ARÁUJO, 2014).

2.2 TEORIA DE CONTROLE

2.2.1 Sistemas Lineares

Sistemas Lineares são todos aqueles que se encaixam no princípio da superposição. Este princípio afirma que a resposta produzida pela aplicação simultânea de duas funções diversas é a soma de suas respostas individuais. Isso quer dizer que para conhecer a resultante de um Sistema Linear onde se recebe diversas entradas, basta realizar a soma do resultado de cada função individualmente.

Um sistema é dito como linear se o princípio da superposição se aplicar a ele. (OGATA, 2003, p.46)

2.2.2 Sistemas de Primeira Ordem

Sistemas de Primeira Ordem são caracterizados com as equações de primeiro grau, onde sua variável tem exponencial unitário. Estes tipos de sistemas são mais fáceis e rápidos para se controlarem e estabilizarem. (OGATA, 2003, p. 180).

2.2.3 Sistemas de Segunda Ordem

Sistemas de Segunda Ordem são aqueles que suas funções podem ser comparadas a equações de segundo grau onde uma de suas variáveis está elevada ao quadrado e a outra possui exponencial unitário. Este tipo de sistema possui um tempo de acomodação e estabilização mais demorada isso ocorre devido a existência de um sobressinal, o tempo de acomodação e amortecimento desse sobressinal irá influenciar diretamente no tempo de acomodação do sistema. (OGATA, 2003, p. 183).

2.2.4 Sinais de Entrada e Saída

São caracterizados como Sinais de Entrada todos aqueles que trazem informações para o sistema, e que serão processadas por ele, de acordo com a programação do sistema será emitido um Sinal de Saída que será responsável por ações como acionar, interromper, aumentar ou diminuir a potência de algum equipamento. Logo todo sinal que chega ao sistema é caracterizado como Sinal de Entrada e todo sinal que o sistema emite é caracterizado como Sinal de Saída.

2.2.5 Análise e modelagem matemática

Análise e modelagem matemática são a base para a eficiência do controle. Portanto deve-se levar em consideração todos os agentes no sistema, para se obter uma equação que

represente todo o sistema. Com isso podem ser criadas simulações gráficas em softwares de análises e simulações como o MATLAB, para determinar métodos de controle e seus ganhos para se obter um melhor resultado.

"Modelos matemáticos de sistemas físicos são elementos-chave no projeto e análise de sistemas de controle. [...] Para entender e controlar sistemas complexos deve-se obter modelos matemáticos quantitativos destes sistemas. É necessário, por tanto, analisar as relações entre as variáveis do sistema e obter um modelo matemático". (DORF e BISHOP, 2013, pag.36).

2.2.6 Função de Transferência

Função de Transferência é a representação matemática de todo um sistema, através dessa representação podemos relacionar a saída e entrada de um sistema e com esta relação fica fácil de localizar algum erro existente no sistema. Como todo o sistema está modelado torna-se fácil a identificação de onde ocorre o erro para buscar e implementar uma correção.

"Na teoria de controle, as funções de transferência são comumente utilizadas para caracterizar as relações de entrada e saída de componentes ou de sistemas, que podem ser descritos por equações diferenciais lineares invariantes no tempo". (OGATA, 2003, p.46)

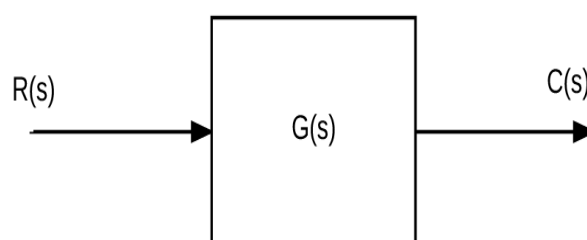


Figura 11: Diagrama de bloco Função de Transferência.

Fonte: Autoral.

Quando é obtida a Função de Transferência no sistema, a mesma pode ser representada em diagramas de blocos, que serão chamados de malhas, existem dois tipos de malhas, que são abertas ou fechadas, onde $G(s)$ é igual a Função de Transferência que está no bloco central do diagrama acima na figura 11, que representa matematicamente todo o sistema.

2.2.7 Malha Aberta

Malha Aberta é quando o sistema não possui uma realimentação. Ou seja, o sistema tem uma entrada de sinal e logo que ele processa este sinal, emite uma saída de sinal. Sistemas de malha aberta não efetuam uma comparação do valor de saída com o valor de entrada, o que o

torna um sistema desvantajoso em relação ao sistema de malha fechada que realiza a comparação entre a saída e entrada, esta comparação servirá para indicar a existência de um erro para que o sistema possa corrigi-lo.

"Os sistemas de controle com malha aberta são aqueles em que o sinal de saída não exerce nenhuma ação de controle no sistema. Isso quer dizer que em um sistema de controle de malha aberta o sinal de saída não é medido nem realimentado para comparação com a entrada". (OGATA, 2003, p.5).

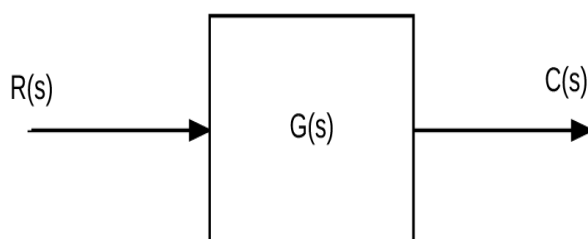


Figura 12: Diagrama de blocos sistema malha aberta.

Fonte: Autoral.

Na imagem a cima, o $R(s)$ representa o sinal de entrada no sistema, o bloco central $G(s)$ é onde ocorre o processo do sistema para a emissão da saída $C(s)$, que é a resultante do processo.

Um exemplo de sistema em Malha Aberta seria a máquina de lavar, apesar dela realizar vários processos ao longo do tempo conforme sua programação como, molho, bater, enxaguar e centrifugar, ela não analisa se a roupa está suja ela simplesmente fará as etapas em tempos determinados sem saber se está tendo o resultado esperado. (OGATA, 2003, p.5).

2.2.8 Malha Fechada

Diferentemente do sistema de Malha Aberta, o sistema em Malha Fechada possui uma realimentação, essa realimentação fará toda a diferença no sistema, tornando-o mais preciso e aumentando sua empregabilidade.

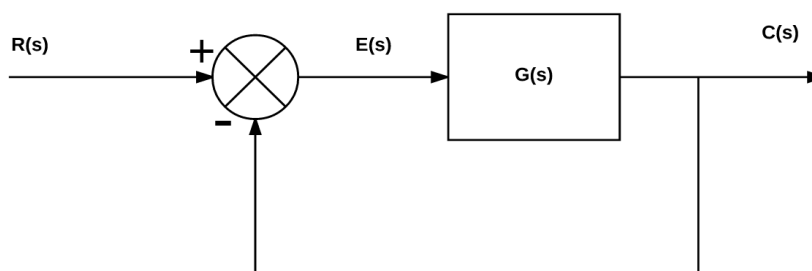


Figura 13: Diagrama de blocos sistema Malha Fechada.

Fonte: Autoral.

Através desse diagrama é possível relacionar diretamente a saída com a entrada $C(s)/R(s)$, obtendo-se nesta comparação um valor de erro $E(s)$ que é a diferença entre o setpoint e o estado atual do sistema, em cima deste erro o sistema irá trabalhar para realizar uma saída capaz de eliminar o erro e alcançar o setpoint desejado.

"Em um sistema de controle de malha fechada, o sinal de erro atuante, que é a diferença entre o sinal de entrada e o sinal de realimentação (que pode ser o próprio sinal de saída ou uma função do sinal de saída e suas derivadas e/ou integrais), realimenta o controlador, de modo que minimize o erro e acerte a saída do sistema ao valor desejado. O termo 'controle de malha fechada' sempre implica a utilização do controle com realimentação para efeito de reduzir o erro do sistema". (OGATA, 2003, p.5).

No diagrama abaixo podemos observar um símbolo constituído por um círculo com uma cruz dentro e sinais externos de soma (+) e subtração (-), este símbolo é chamado de somador, pois ele soma variáveis ao sistema. Como nem sempre o sistema é capaz de eliminar o erro sozinho, são implementados controladores ao sistema que irão direcionar e controlar seu trabalho para a eliminação do erro. Adicionando este controlador à representação do diagrama de blocos ficará assim.

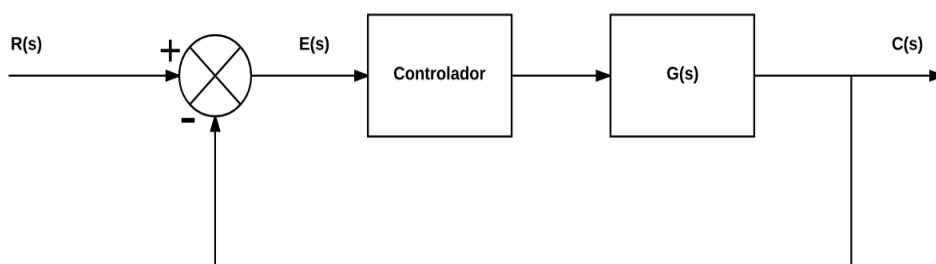


Figura 14: Diagrama de blocos sistema realimentado com controlador.

Fonte: Autoral.

Um exemplo simples de sistema em malha fechada é a geladeira, ela é configurada para deixar sua temperatura sempre em um determinado valor, ao colocar um alimento acima da temperatura interna da mesma, logo haverá um desequilíbrio entre a temperatura desejada e a temperatura presente, então a geladeira aciona o resfriamento até igualar a temperatura, zerando o erro existente entre o setpoint (temperatura desejada) e a temperatura no momento.

2.2.9 Transformada de Laplace

É uma transformação de equações criada por Pierre-Simon Laplace. Esse método permite a conversão de funções como as funções senoidais, as funções senoidais amortecidas e as funções exponenciais em funções algébricas com uma variável complexa "S". Ao realizar tal procedimento, a solução do problema se torna mais fácil e rápido, pois deixa de trabalhar no

domínio do tempo e passa-se a trabalhar no domínio da frequência. Outra vantagem é que essa transformação permitirá o uso de técnicas gráficas para prever o desempenho do sistema. Após a solução da problemática fica fácil de converte-la para o domínio de origem, a grande vantagem da Transformada de Laplace está em poder manipular as funções de forma que torna mais fácil e ágil sua resolução. Para a realização das conversões basta utilizar uma tabela já pré-definida que prevê diversos modelos de equações. Esta tabela estará disponível no anexo A.

"A transformada de Laplace é um método que pode ser usado de maneira proveitosa para solucionar equações diferenciais lineares. Por meio de sua utilização, podemos converter muitas funções comuns, como funções senoidais, funções senoidais amortecidas e funções exponenciais, em funções algébricas de uma variável complexa s ". (OGATA, 2003, p.8).

2.2.10 Controle

Sugestivamente pelo nome, controle quer dizer ter controle, poder controlar algo. Existem diversos métodos de controle onde são empregados de acordo com a necessidade, esses métodos variam sua complexidade de acordo com o sistema controlado.

"Como os avanços no controle automático, na teoria e na prática, vêm produzindo meios para otimizar o desempenho dos sistemas dinâmicos, melhorar a produtividade, diminuir o trabalho árduo de várias rotinas de operações manuais repetitivas, entre outros, a maioria dos engenheiros e dos cientistas devem ter agora bons conhecimentos nessa área". (OGATA, 2003, p.1).

Os controladores industriais podem ser classificados de acordo com suas ações de controle:

- Controlador de duas posições ou on-off;
- Controladores proporcionais;
- Controladores integrais;
- Controladores proporcional-integrais;
- Controladores proporcionais-derivativos;
- controladores proporcional-integral-derivativos.

2.2.10.1 Controle on-off

Um método de controle muito conhecido e utilizado é o on-off (ligado-desligado), este consiste em apenas duas ações que é ligar ou desligar o atuador no controle. Um exemplo que utiliza esse sistema de controle é o de encher uma caixa d'água. Pois bem, temos uma caixa

onde deverá estar sempre cheia e nunca transbordar, então colocamos nela uma boia, esta servirá para indicar mecanicamente se a caixa está cheia ou vazia e, a boia sempre estará no nível da água. Então, se a caixa esvaziar, a boia irá abaixar e liberar o fluxo de água para dentro da caixa e quando a água chegar em seu limite estipulado, a boia irá bloquear a passagem de água, impedindo então o fluxo e o transbordamento.

"Em um sistema de controle de duas posições, o elemento atuante tem somente duas posições fixas, que são, muitos caos, simplesmente on e off". (OGATA, 2003, p. 53).

2.2.10.2 Controle Proporcional (P)

Controle proporcional é a aplicação do ganho proporcional ao sistema. Dependendo do sistema, apenas o seu ajuste já é suficiente para eliminação do erro. Em maioria dos sistemas controlados, o ganho proporcional será empregado, podendo surgir métodos de controle proporcional, proporcional-integral, proporcional-derivativo e proporcional-integral-derivativo.

O ganho proporcional será simplesmente a multiplicação do valor proporcional com o erro existente naquele instante, isso dará um valor de ganho proporcional que é chamado de K_p . Com esta operação o sistema tende a diminuir o erro podendo até mesmo zerá-lo e realizar a estabilização e acomodação mais rápida do sistema. (OGATA, 2003, p.54).

Segundo Ogata, o valor proporcional se dará através da equação 2.3:

$$u(t) = K_p e(t) \quad (2.3)$$

ou, transformando por Laplace,

$$\frac{U(s)}{E(s)} = K_p \quad (2.4)$$

2.2.10.3 Controle Integral (I)

Controle Integral será a aplicação do ganho integral k_i no controle. A sua operação no sistema é diferente do proporcional, pois além de ser multiplicado pelo valor de erro no momento, ele multiplica também o tempo em que o erro persiste e por fim soma seu valor anterior ao resultado, quanto maior for o tempo da existência do erro no sistema maior será o valor do Integral, isso ocorre porque existe a necessidade do sistema alcançar o valor de setpoint com maior agilidade e como o ganho aumenta cada vez mais em relação ao tempo da existência de erro, rapidamente o sistema alcançará o valor de setpoint desejado.

De acordo com Ogata, o valor integral se dará através da equação 2.5:

$$\frac{du(t)}{dt} = K_i e(t) \quad (2.5)$$

$$u(t) = K_i \int_0^t e(t) dt$$

"onde K_i é uma constante ajustável. A função de transferência de um controlador integral é:

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.6)$$

Conforme Ogata, quando juntamos em um sistema o ganho proporcional mais o integral teremos um controlador proporcional-integral, a ação deste controle ao sistema é definida por:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2.7)$$

ou então, a função de transferência do controlador é:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right) \quad (2.8)$$

onde T_i é chamado de tempo integrativo.

2.2.10.4 Controle Derivativo (D)

Ao ajustar os ganhos no sistema pode se acelerar muito sua resposta, o que poderá ocasionar um valor de sobressinal e com isso pode vir acontecer distúrbios permanentes no sistema, então para que isso não ocorra e o sistema consiga se estabilizar rapidamente, o ganho derivativo dará uma reduzida no aumento do ganho ocasionado do controlador, o que irá proporcionar um amortecimento no sistema tendo pouco sobressinal e uma acomodação e estabilização mais rápida, o ganho derivativo é dado por K_d .

De acordo com Ogata, o valor à ação do controle proporcional-derivativo se dará conforme a equação 2.9.

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (2.9)$$

e a função de transferência é:

$$\frac{U(s)}{E(s)} = K_p (1 + T_d s) \quad (2.10)$$

onde T_d é chamado de tempo derivativo.

2.2.10.5 Controle Proporcional Integral Derivativo (PID)

O método de controle PID pode-se dizer que é o mais utilizado em fábricas. É um método simples e de alta confiabilidade. Sua forma de emprego é simples e seus ganhos podem facilmente serem ajustados no local sem grandes dificuldades e com bons resultados. Este emprego se tornou tão fácil que alguns equipamentos e recursos já conseguem ajustar seus ganhos automaticamente através de análises e testes pré-definidos. Mas para se obter um melhor resultado é preciso realizar projeções, simulações e testes a fim de se poder realizar o ajuste mais fino possível. Para implementá-lo precisa-se ajustar seus parâmetros, que são: ganho proporcional, ganho integral, ganho derivativo.

Segundo Dorf e Bishop, o sistema de controle PID possui a seguinte função de transferência:

$$G_c(s) = K_P + \frac{K_I}{s} + K_D s \quad (2.11)$$

"A popularidade dos controladores PID pode ser atribuída parcialmente ao seu desempenho robusto em uma larga faixa de condições de operação e parcialmente à sua simplicidade funcional, a qual permite os engenheiros operá-los de uma maneira simples e direta." (DORF e BISHOP, 2013, pag.701).

2.3 SOFTWARES

Softwares são aplicações desenvolvidas para a realização de tarefas, desde coisas simples como escrever um texto e imprimi-lo a mais complexas como o desenvolvimento de outros programas capazes de realizar cálculos, simulações e inúmeras outras aplicações, com intuito de agilizar e facilitar operações.

2.3.1 MATLAB

O MATLAB é um software de alta performance desenvolvido pela MathWorks para efetuação de diversos tipos de cálculos e simulações. Um dos recursos para as simulações do MATLAB é uma ferramenta chamada Simulink que pode criar e simular diversos ambientes dentro da automação industrial quanto em outros. O MATLAB não trabalha com representação de equipamentos como outros softwares de simulações de circuitos, ele trabalha com os valores matemáticos e suas representações matemáticas no sistema. A Interface de operação do MATLAB pode ser observado na figura 15, em seguida na figura 16, a Interface da ferramenta Simulink.

"O MATLAB é um programa interativo para cálculos científicos e de engenharia. A família de programas MATLAB inclui o programa base e mais uma variedade de toolboxes, uma coleção de arquivos especiais chamados de arquivos m (m-files) que estendem a funcionalidade do programa base. Juntos, o programa

base e o Control System Toolbox fornecem a capacidade de utilizar o MATLAB para o projeto e a análise de sistemas de controle". (DORF e BISHOP, 2013, pag.785).

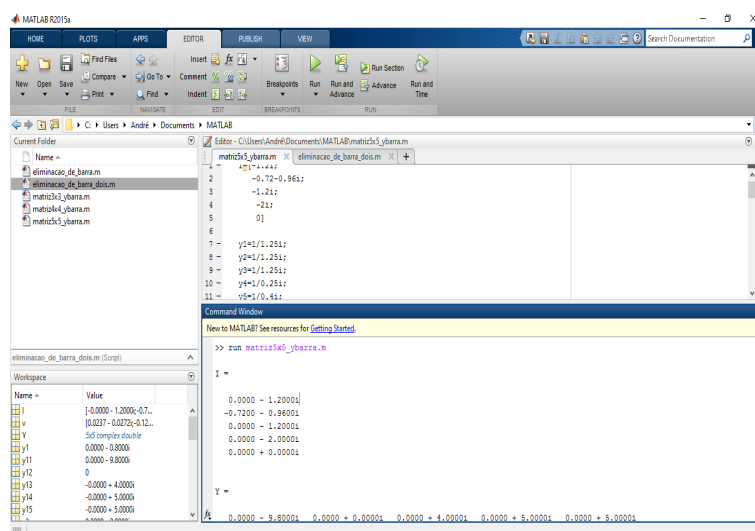


Figura 15: Interface Software MATLAB.

Fonte: Autoral.

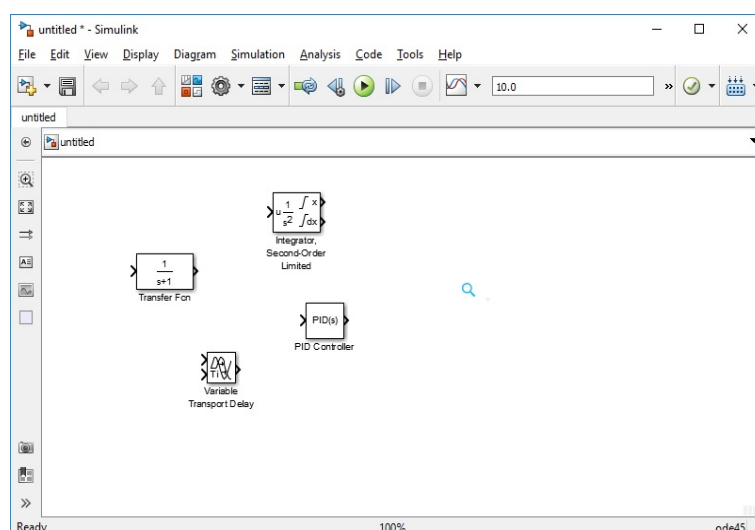


Figura 16: Interface simulação MATLAB Simulink.

Fonte: Autoral.

2.3.2 Programas Scada

Programas Scada são softwares supervisórios, desenvolvidos para monitorar e supervisionar variáveis em sistemas de controle, isso ocorre através de um protocolo de comunicação. Basicamente o programa Scada fica instalado em um computador recebendo informações de um controlador seja ele em plataforma Arduino ou CLP. A comunicação ocorre pelo protocolo de

comunicação, podendo ser o Modbus ou algum outro protocolo existente que seja reconhecido pelo sistema supervisor e pelo controlador empregado, esta pode ser realizada por diversos meios como ethernet, cabo coaxial, Wi-Fi, entre outros meios de transmissão de informação. (SILVA e SALVADOR, 2005).

2.3.2.1 Elipse Scada

Elipse SCADA é uma das aplicações SCADA desenvolvida pela Elipse Software, através deste software é possível desenvolver uma tela em um computador que será tido como servidor, através desta tela pode-se supervisionar, monitorar e controlar processos em uma ou mais plantas, desde que sejam corretamente configuradas e implementadas.(CARVALHO, 2012).

2.3.3 Modbus

Modbus é um protocolo de comunicação utilizado por softwares em sistemas de automação industrial, criado originalmente pela Modicon no ano de 1979. Foi e ainda é um dos protocolos mais utilizados. Desde 2004 é um protocolo livre que recebe alterações e melhorias por meio de diversos colaboradores.(FREITAS, 2014).

3 PROCEDIMENTOS METODOLÓGICOS

3.1 PLANTA DE CONTROLE

Para o desenvolvimento prático necessita-se de uma planta base, com esta serão realizados ajustes, testes e simulações. Ela poderia ser construída e montada, mas no caso foi adotado o kit para controle de nível da bit9, modelo existente em um dos laboratórios das Faculdades Integradas de Caratinga (FIC).

3.1.1 Kit para controle de nível da bit9

O kit para controle de nível da bit9, é um kit didático composto por dois reservatórios denominados A e B, além destes reservatórios o kit possui uma bomba e sensores capacitivo em ambos reservatórios.

A bomba possui um funcionamento entre 0V e 10V, ela enche o reservatório A enquanto esvazia o reservatório B.

Os sensores capacitivos nos reservatórios são duas barras paralelas em cada reservatório e à medida em que o nível aumenta ou diminui, o sensor emite um sinal de saída variando de 0V a 10V, sendo 0V o valor mínimo e 10V valor máximo do nível.

Ambos reservatórios possuem a capacidade aproximada de 1L. No reservatório A, observamos uma válvula mecânica com diâmetro de 0.009m que o esvazia jogando seu conteúdo no reservatório B. A bomba está na parte inferior do reservatório B onde ela capta o fluido e o envia por uma tubulação até a parte superior do reservatório A. Na figura 17 podemos visualizar o Kit mencionado.



Figura 17: Kit controle de nível Bit9.

Fonte: Autoral.

3.2 ANÁLISE DA PLANTA

Em primeiro contato com a planta foi observado as características citadas no capítulo, 3.1.1, em seguida teve-se início aos procedimentos necessários para a realização da modelagem matemática e simulações através do software MATLAB.

Para a realização da modelagem matemática precisou-se das seguintes informações:

- Área do reservatório;
- Volume do reservatório;
- Vazão da bomba d'água.

Para encontrar a área e o volume dos reservatórios, necessitou realizar medição de suas grandezas geométricas largura, profundidade e altura. A tabela 1 contém os valores medidos.

Tabela 1: Medidas do Reservatório

Grandeza	Medida (cm)
Largura	10
Profundidade	6.9
Altura	14.9

Fonte: Autoral.

Em seguida houve-se à análise da capacidade de vazão da bomba em diferentes níveis de tensão aplicada, para a realização dessa medição foi se verificado o tempo que a bomba levava para drenar de seu reservatório toda a quantidade de líquido existente nele. A tabela 2 mostra os resultados obtidos.

Para montagem da tabela 2 necessitou empregar a equação 3.2, esta equação nos dá o valor de vazão, relacionando o volume com o tempo e foi adotado um volume fixo de 0.5 litros de água para todas aferições.

$$Q = \frac{V}{t} \quad (3.1)$$

Q = Vazão;

V = Volume;

T = tempo em segundos;

Tabela 2: Tensão X Vazão

Tensão	Tempo (seg)	Vazão ($\mu\text{m}^3/\text{s}$)
1 Volt	17.33	28.85
2 Volts	12.34	40.52
3 Volts	10.60	47.17
4 Volts	9.73	51.39
5 Volts	9.11	54.88
6 Volts	8.33	60.02
7 Volts	7.85	63.69
8 Volts	7.49	66.76
9 Volts	6.80	73.53
10 Volts	6.14	81.43

Fonte: Autoral.

Com base na tabela 2 acima foi criado um gráfico que pode ser visualizado na imagem 17, que mostra a variação da vazão de acordo com a tensão aplicada. Notem que a variação está bem próxima de ser linear.

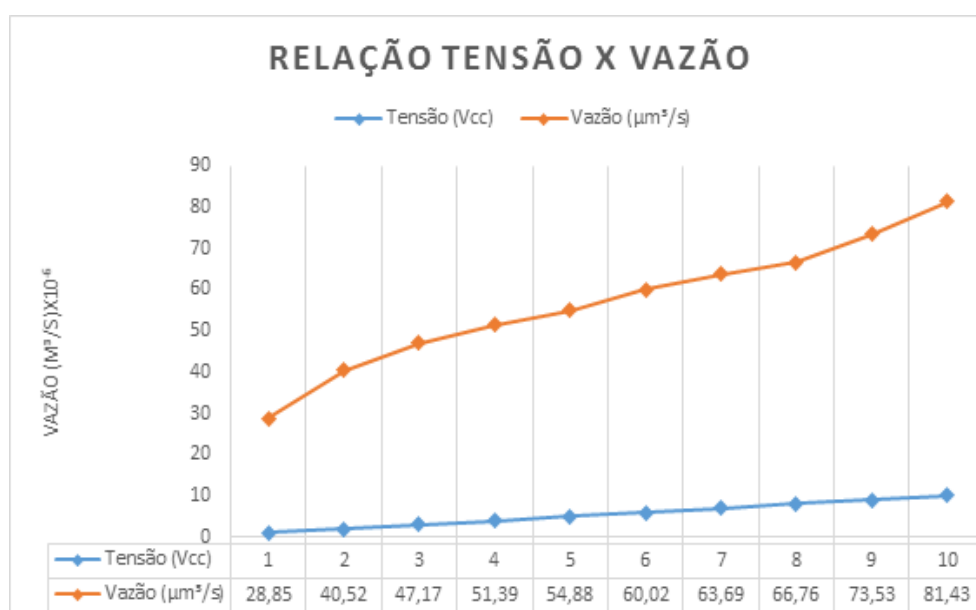


Figura 18: Gráfico tensão x vazão.

Fonte: Autoral.

3.3 MODELAGEM MATEMÁTICA

Tendo em posse todas as grandezas da planta podemos dar início a modelagem matemática da mesma. Aplicando o conceito de Laplace, para obtenção da função de transferência.

De acordo com Ogata (2003, p. 125), a resistência e capacitância de sistemas de nível de

Líquido podem ser dada em:

$$R = \frac{H}{Q} \quad (3.2)$$

Onde:

R = Fluxo de líquido;

H = Altura máxima do nível em m;

Q = Vazão máxima em m³/s;

De acordo com Halliday (2009, p. 75), para calcular a vazão do reservatório A para o B, o teorema de Bernoulli se dará conforme a equação 3.3.

$$\rho \cdot g \cdot h = \frac{1}{2} \rho \cdot v^2 \quad (3.3)$$

Logo:

$$v = \sqrt{2 \cdot g \cdot h} \quad (3.4)$$

Como:

$$Q = v * A \quad (3.5)$$

Então:

$$Q = A * \sqrt{2 \cdot g \cdot h} \quad (3.6)$$

Logo a resistência se dará por:

$$R = \frac{h}{A * \sqrt{2 \cdot g \cdot h}}$$

$$R = \frac{h * \sqrt{2 \cdot g \cdot h}}{A * 2 \cdot g \cdot h} \quad (3.7)$$

$$R = \frac{\sqrt{2gh}}{A2g}$$

De acordo com Ogata (2003, p. 127). “Com base na hipótese de que o sistema seja linear ou linearizado, a equação diferencial desse sistema pode ser obtida a seguir. Como o fluxo de entrada menos o fluxo de saída durante um pequeno intervalo de tempo dt é igual à quantidade adicional armazenada no reservatório, temos que”:

$$Cdh = (q_i - q_o)dt \quad (3.8)$$

“A partir da definição da resistência, a relação entre q_o e h é dada por”:

$$q_o = \frac{h}{R} \quad (3.9)$$

“A equação diferencial desse sistema para um valor constante de R torna-se”:

$$RA * \frac{dh}{dt} + h = Rq_i \quad (3.10)$$

Onde:

A = Área do reservatório em m^2 .

“Note que RA é a constante de tempo do sistema. Tomando a Transformada de Laplace de ambos os membros e considerando condições iniciais nulas, obtemos”:

$$(RA s + 1)H(s) = RQ(s) \quad (3.11)$$

$$\frac{H(s)}{Q(s)} = \frac{R}{RA s + 1} \quad (3.12)$$

Por fim, foi encontrada na equação 3.13 a Função de Transferência que representa o sistema.

$$G(s) = \frac{R}{RA s + 1} \quad (3.13)$$

Logo:

Com a equação 3.14 será identificada à área da passagem na válvula do reservatório "A" do kit para controle de nível da bit9.

$$A = \pi * r^2 \quad (3.14)$$

$$A = \pi * 20.25 * 10^{-6} = 63.62 * 10^{-6} m^2$$

Agora será encontrado o valor da resistividade por meio da equação 3.15.

$$R = \frac{\sqrt{2 * 10 * 0.149}}{63.62 * 10^{-6} * 2 * 10} \quad (3.15)$$

$$R = 1356.7$$

Na equação 3.16 será determinado à área do reservatório B do kit para controle de nível da nit9, que por sua vez é a mesma do reservatório A.

$$A_2 = 10 * 6.9 = 69cm^2 = 0.0069m^2 \tag{3.16}$$

Com todas as variáveis da equação 3.13 descobertas, sua resolução é mostrada na equação 3.17.

$$G(s) = \frac{1356.7}{1356.77 * 0.0069s + 1} \tag{3.17}$$

$$G(s) = \frac{1356.7}{9.36s + 1}$$

3.4 SIMULAÇÃO EM SOFTWARE

Após encontrar a equação que representa o sistema, deverá ser feita a simulação no MATLAB, através desta simulação será possível estipular os ganhos do controlador PI.

De acordo com com Ogata (2003, p.240), na tabela 5.1 é demonstrado que sistemas do tipo 1 que possuem o controlador proporcional-integral é capaz de eliminar o erro em sistemas que possuem entrada em degrau, tornando desprezível a utilização do ganho derivativo que pode ocasionar uma lentidão no sistema. Na figura 32 é visualizado a imagem da tabela 5.1.

	Entrada em degrau $r(t) = 1$	Entrada em rampa $r(t) = t$	Entrada em aceleração $r(t) = \frac{1}{2}t^2$
Sistema do tipo 0	$\frac{1}{1 + K}$	∞	∞
Sistema do tipo 1	0	$\frac{1}{K}$	∞
Sistema do tipo 2	0	0	$\frac{1}{K}$

Figura 19: Tabela erro regime permanente.

Fonte: Engenharia de Controle Moderno.

Para dar início a simulação é necessário realizar a montagem do diagrama de blocos, colocando em cada bloco sua função específica no sistema. Na figura 20 podemos visualizar a montagem básica do sistema no Simulink.

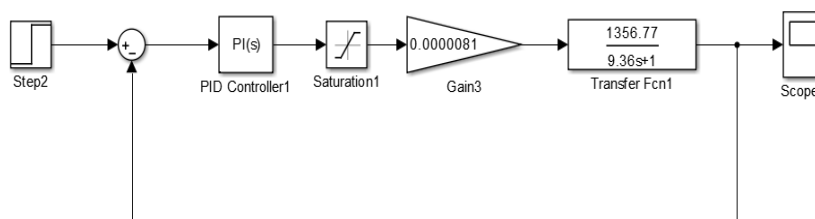


Figura 20: Diagrama de blocos do sistema.

Fonte: Autoral.

No quinto bloco do diagrama mostrado na figura 20 observamos um valor de ganho, este bloco e este valor representa o ganho da bomba d'água ao sistema. Através da equação 3.18 é possível encontrar o valor deste ganho.

$$ganho = \frac{vazomximadabomba}{tensoalimentaodabomba}$$

$$ganho = \frac{81.43 * 10^{-3}}{10} \quad (3.18)$$

$$ganho = 0.0000081$$

No Simulink temos uma ferramenta que ajusta os valores dos ganhos do controlador PI automaticamente e nesta mesma ferramenta podemos alterar os valores de acordo com a onda gráfica que se deseja.

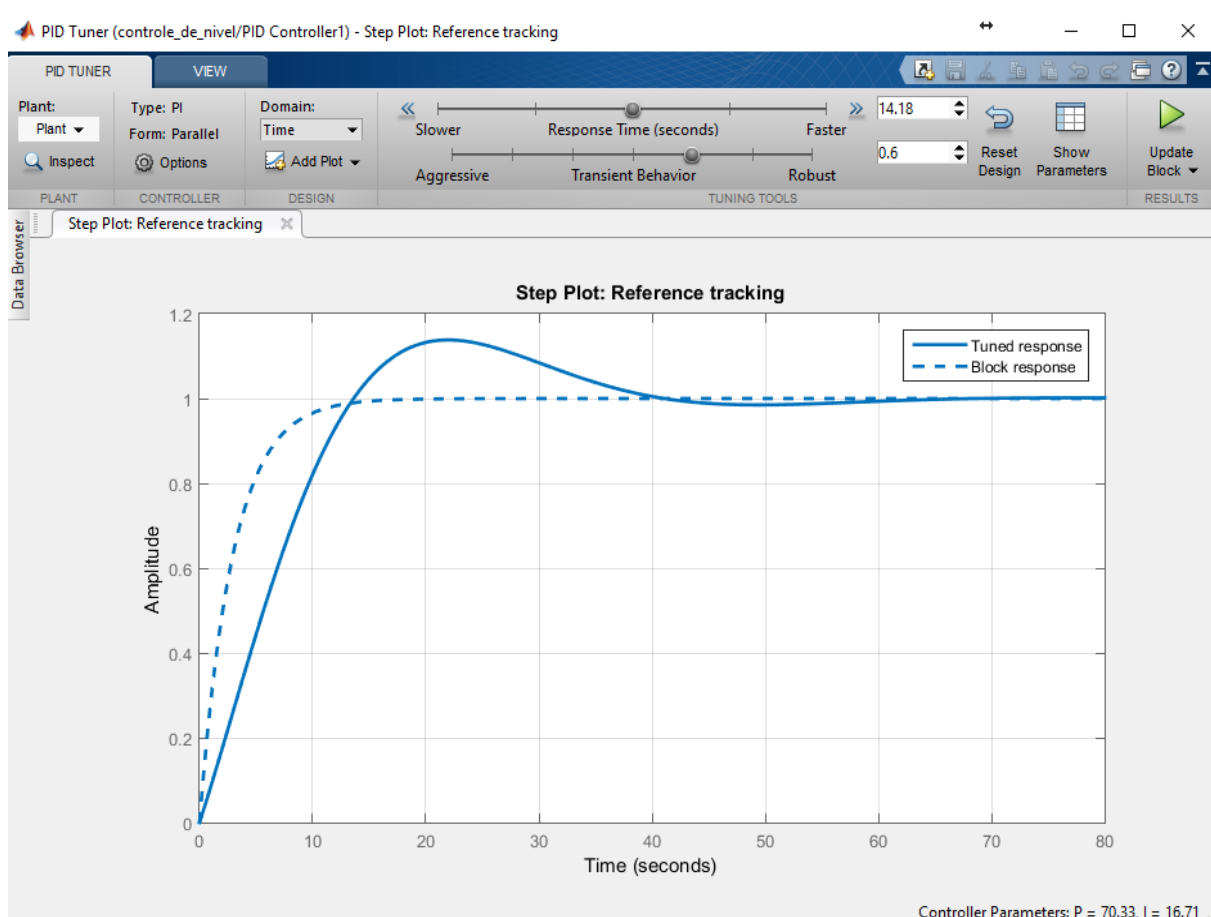


Figura 21: Ajustando Controlador PI.

Fonte: Autoral.

Na figura 21 é apresentado a saída atual e a previsão da resposta do sistema. A linha

pontilhada representa a saída com os ajustes atuais do controlador PI, a linha contínua é o valor ajustado automaticamente pelo sistema e para influenciar e modificar os valores estipulados automaticamente basta alterar o valor de resposta do sistema mais o valor transitório de comportamento.

É importante ressaltar que por se tratar de um sistema de controle de nível não é interessante ter valor de sobressinal, pois isso na prática irá ocasionar um transbordamento. Devido ao atuador só ter a capacidade de enviar líquido ou seja, a bomba não tem como sugar o líquido de volta, por esta razão devemos tomar medidas de segurança, que irão prevenir um possível transbordamento no reservatório. Essas medidas seriam as de não permitir um ajuste no controlador com sobressinal e não definir um setpoint no sistema muito próximo ou igual ao valor de nível máximo. No sistema descrito foi estipulado na configuração da programação que o valor de setpoint não poderá ser maior que 95% de sua capacidade, isso deve-se ao motivo que sempre existirá um erro devido a distúrbios ocasionados com a variação da vazão, até porque este é o motivo do controle, sempre deixar o nível no valor de setpoint ajustado independente da vazão com o menor valor de erro possível.

Na figura 22, é visualizado as propriedades do bloco do controlador PI, onde seus valores são implementados e ajustados.

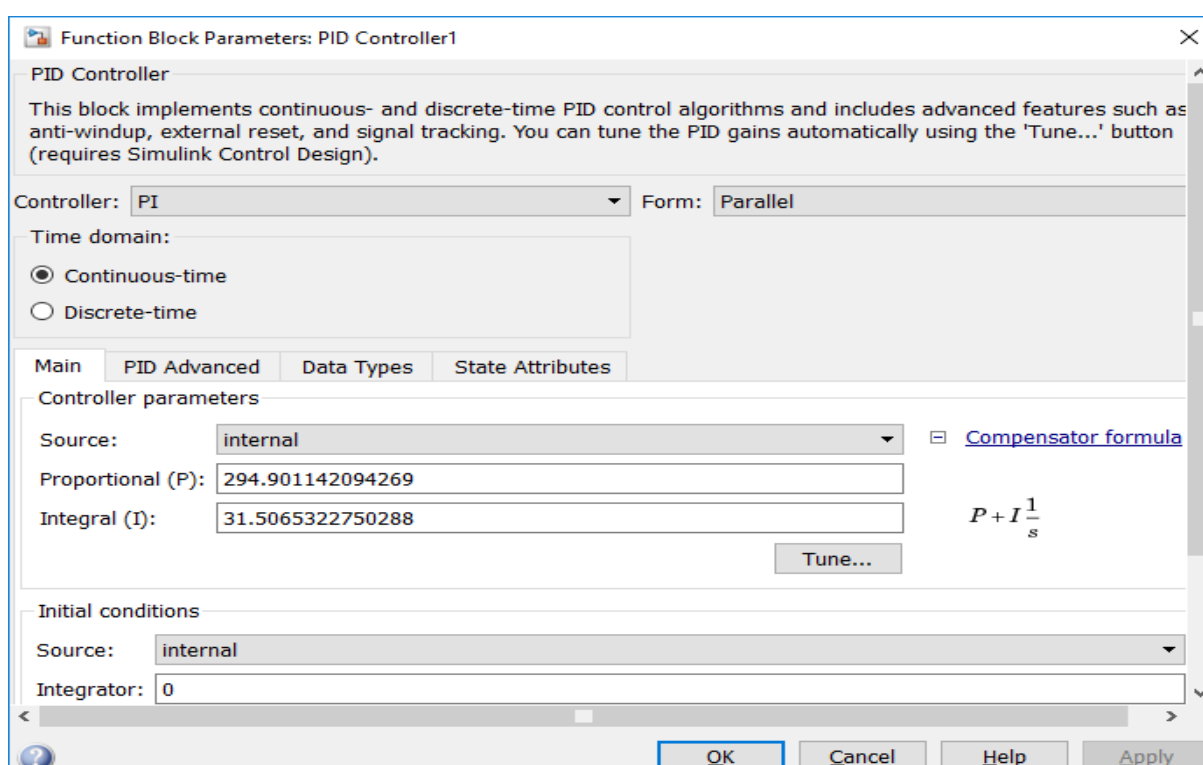


Figura 22: Valores do Controlador PI.

Fonte: Autoral.

Com os ganhos do controlador PI ajustado, o sistema apresentou a seguinte saída que é

mostrada na figura 23.

$$P = 294.9$$

$$I = 31.51$$

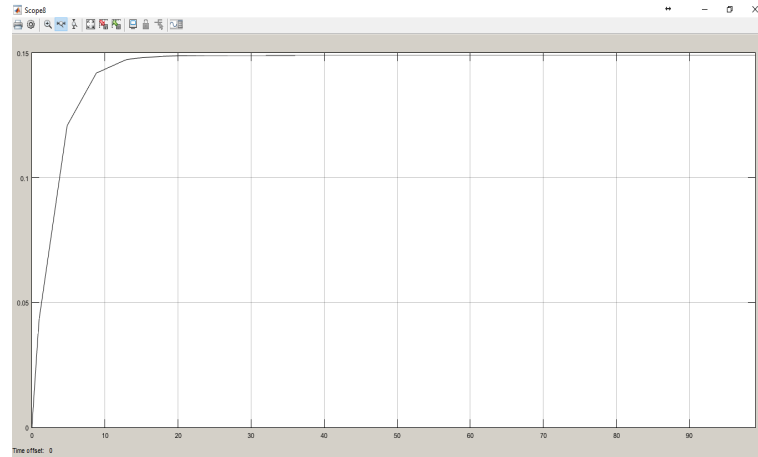


Figura 23: Resposta do sistema.

Fonte: Autoral.

3.4.1 Ajuste Fino da Simulação

Com os valores de ganho do controlador já ajustado, agora é hora de desmembrar o diagrama e acompanhar o comportamento individual dos componentes no sistema. Na figura 24 é mostrado o novo diagrama de blocos do sistema que permite este acompanhamento.

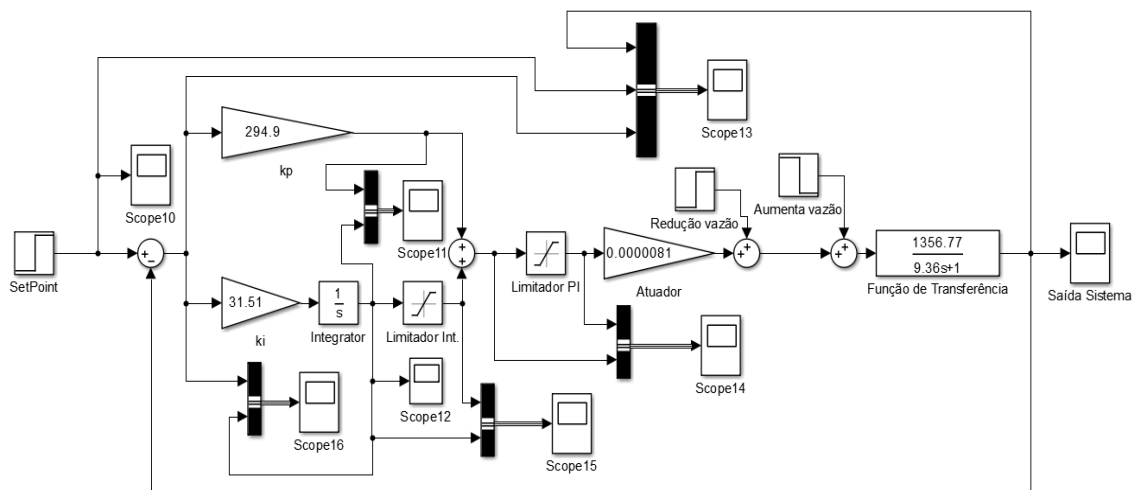


Figura 24: Diagrama de blocos modificado do sistema.

Fonte: Autoral.

O diagrama representado na figura 24 é a simulação final do sistema no MATLAB, com esta simulação podemos observar como o sistema irá reagir diante da variação da vazão. O

SetPoint foi ajustado com o valor correspondente a 8cm, em seguida ocorre a separação do bloco do controlador PI onde o intuito desta separação é observar a variação de cada ganho individualmente no sistema durante a simulação, no bloco representante do controle Integral há um ponto de saturação que não o permite ultrapassar o valor correspondente a 45% do valor máximo do controlador PI que se limita em 100%. Isso ocorre quando na programação do Arduino a função PWM mapeia o controlador PI de 0% a 100% em relação a potência aplicada no atuador, pois o atuador não pode receber mais do que 100% de potência, porque implicaria em sua danificação. Então como o controlador PI é limitado em 100%, logo ficou interessante limitar o controlador integral em 45%, devido o seu valor variar em relação ao tempo. Caso por algum motivo aconteça o desligamento indesejável da bomba, o erro continuaria a existir por um tempo indeterminado, o que resultaria no valor do integral tendendo ao infinito, resultando em um descontrole quando a bomba voltasse ao funcionamento.

Ainda na figura 24, após o bloco que representa o ganho do atuador, observamos dois somadores onde ambos estão ocasionando distúrbios ao sistema em tempos diferentes, o primeiro reduz a vazão e o segundo, aumenta. Na figura 25 é observado saída do processo de simulação do sistema com um resultado satisfatório.

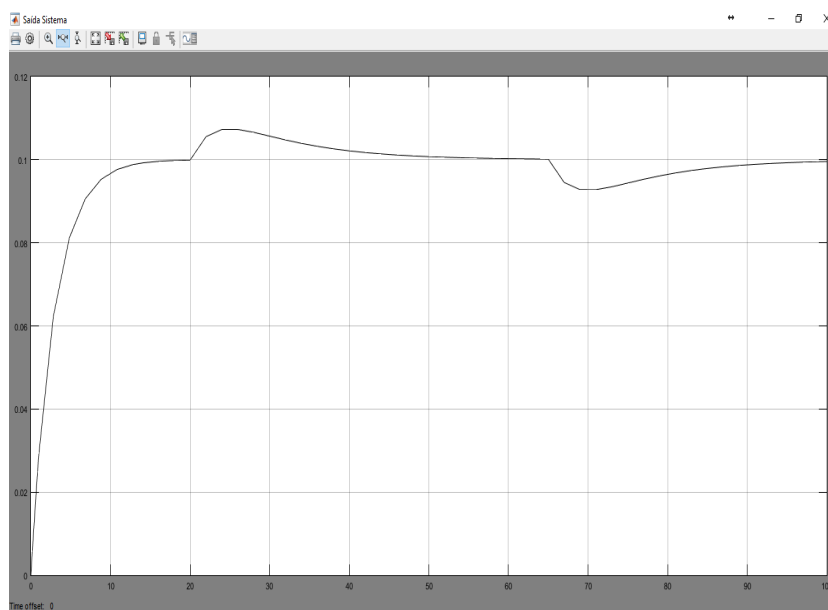


Figura 25: Resposta do sistema 2.

Fonte: Autoral.

3.5 IMPLANTAÇÃO DO CONTROLE E MONITORAMENTO DE NÍVEL

O controle e monitoramento de nível foi desenvolvido com o intuito de atender os seguintes critérios:

- Manter o nível sempre no valor indicado;

- Ter precisão e resposta rápida nas variações;
- Poder ser definido valor de setpoint no local e remotamente.

Para atender a esses critérios temos a necessidade de programar um Arduino para realizar o controle do sistema e junto a ele, a necessidade de desenvolver um ambiente gráfico que realize a interação entre o sistema e seu operador. Para tal finalidade foi desenvolvida uma tela de supervisão e monitoramento no aplicativo elipse scada.

3.6 PROGRAMAÇÃO ARDUINO

Para o bom funcionamento de todo o sistema envolvido é necessário uma programação que consiga atender todas as necessidades sem prejudicar uma das funções.

O primeiro desafio é. Como realizar a leitura do nível?

Para tal tarefa necessitamos adquirir um sensor capaz de identificar o nível. Na ocasião foi escolhido o sensor ultrassônico que é um equipamento rápido e preciso. O sensor utilizado é apresentado na figura 26, e na tabela 3 suas especificações.



Figura 26: Sensor ultrassônico HC-SR04.

Fonte: Vida de Silício.

Tabela 3: Especificações Sensor ultrassônico HC-SR04.

Especificações	
Alimentação	5V DC
Corrente em espera	< 2mA
Corrente de Operação	15mA
Ângulo de efeito	15°
Alcance	2cm a 4m
Precisão	3mm
Dimensão	45x20x15mm

Fonte: Vida de Silício.

A leitura do nível é de grande importância pois é somente através dela que será possível identificar o erro entre o nível ajustado e o nível atual do líquido. O sensor se mostrou muito preciso tendo uma variação mínima em suas medições e para tornar essa leitura ainda mais

precisa, foi declarado no código que leitura deveria ser feita cinco vezes com intervalo de 3 milissegundos, então essas leituras são somadas e por fim é feita uma média que a torna mais precisa e confiável.

Com a leitura da distância realizada, o programa a compara com o valor de setpoint para verificar o erro e efetuar o controle. O setpoint em questão poderá ser controlado tanto remotamente pela tela de supervisão e monitoramento quanto localmente. Remotamente basta informar no campo específico o valor em porcentagem do reservatório que se deseja, exemplo 50 para 50% da capacidade do reservatório e localmente basta girar o botão indicado para setar o valor desejado. Por medidas de segurança, para evitar um possível transbordamento do líquido devido a um possível sobressinal, foi definido na programação que o valor de setpoint nunca poderá ultrapassar o valor de 95% da capacidade de armazenamento do reservatório.

Após todos os processos, surge então o valor de erro e através deste valor o programa irá processar e ajustar o valor do controlador PI para atuar e tentar eliminar o erro. Com o valor de PI determinado, o programa realiza um mapeamento relacionando proporcionalmente o valor mínimo a máximo de PI com o valor mínimo a máximo de tensão que o Arduino consegue emitir, logo o Arduino irá emitir um sinal que varia de 0V a 5V que representa a potência que o atuador deverá atuar para eliminar o erro. Como a tensão do sinal de saída do Arduino é muito baixa e não possui potência para alimentar o atuador foi necessário o desenvolvimento de um circuito que amplie este sinal. Na planta em questão a amplificação deste sinal foi suficiente para alimentar o atuador, mas isso ocorre devido a baixa potência do mesmo. Em caso de uma planta maior onde exista uma bomba d'água que necessite de uma potência maior, o correto é jogar este sinal amplificado em um equipamento que fará a leitura e interpretação deste sinal e o controle potência do atuador de acordo com a variação do sinal.

No decorrer do código desenvolvido e utilizado que está no Apêndice A, percebemos que as variáveis possuem valores diferentes dos ajustados na modelagem do sistema, isso ocorre devido que na modelagem os valores adotados principalmente das medidas estavam em metros e na programação para facilitar e obter um melhor resultado, as unidades estão em centímetros. Por esse motivo é observado que os valores dos ganhos do controlador proporcional e integral são cem vezes menores do que os obtidos na modelagem. Outra coisa que será notado é uma diferença no valor do controlador proporcional e integral, houve uma pequena alteração nos valores, pois na prática eles mostraram um melhor resultado do que os valores obtidos na modelagem, este ajuste seria o chamado ajuste fino ou ajuste em prática. O valor proporcional que era de 294.9 passou para 295 e o valor integral que era 31.51 passou para 79.20.

Na figura 27 é visualizado parte do código de programação, onde os valores do controlador proporcional e integral são definidos, e que o valor do integral é limitado em 4.5, isso ocorre devido a limitação de 45% que o controlador integral recebe em relação ao valor máximo do controlador PI.



```
control_monit_elipse | Arduino 1.6.11
Arquivo Editar Sketch Ferramentas Ajuda
control_monit_elipse

/*****
          CONTROLE PID
*****/
double deltaTime = (millis() - (lastProcess));
lastProcess = millis();

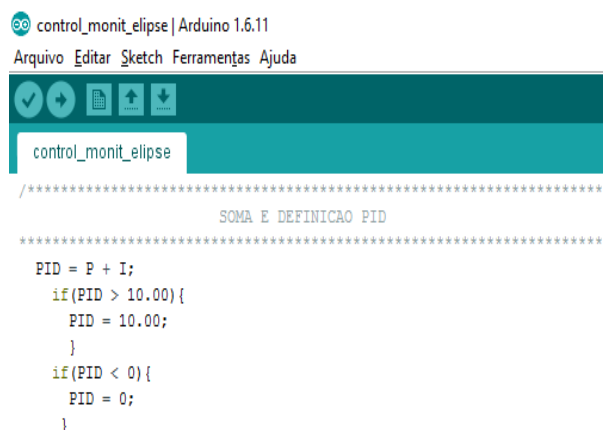
//P
P = error * kP;

//I
I = I + (error * kI) * (deltaTime/1000.0);
if(I > 4.5){
  I = 4.5;
}
if(I < 0){
  I = 0;
}
```

Figura 27: Definindo valor Proporcional e Integral.

Fonte: Autoral.

Na figura 28 é visualizado parte do código responsável por somar os valores dos controladores proporcional e integral. Onde a soma se limita a um valor igual a 10, isto ocorre devido a este valor representar o atuador com operação em 100% de sua capacidade.



```
control_monit_elipse | Arduino 1.6.11
Arquivo Editar Sketch Ferramentas Ajuda
control_monit_elipse

/*****
          SOMA E DEFINICAO PID
*****/
PID = P + I;
if(PID > 10.00){
  PID = 10.00;
}
if(PID < 0){
  PID = 0;
}
```

Figura 28: Definindo valor controlador PI.

Fonte: Autoral.

3.7 CIRCUITO ELETRÔNICO

Para o funcionamento do atuador, foi preciso a criação de um circuito capaz de pegar o sinal de saída do Arduino e acionar à bomba d'água em mesma proporção, isso ocorre devido o Arduino não ter potência suficiente para fornecer alimentação ao atuador.

Como a bomba d'água opera em corrente contínua e com tensão de 10V, o circuito desenvolvido com componentes eletrônicos básicos como Resistor, Transistor, Diodo e uma fonte de alimentação com capacidade limitada de potência foi suficiente para atender a performance máxima do atuador. Caso o mesmo fosse uma bomba de corrente alternada com alta potência,

bastaria redirecionar o sinal de saída do circuito desenvolvido para um outro equipamento que fosse capaz de gerenciar a performance do atuador de acordo com a faixa de operação estipulada pelo Arduino.

Na figura 29 podemos visualizar o esquemático do circuito desenvolvido. Que funciona da seguinte forma: Ao alimentar a base do Transistor Q1 passando pelo Resistor R2 de 10Kohms com o sinal emitido pelo Arduino que pode chegar a 5V, o coletor de Q1 capta a tensão da fonte de alimentação passando também por R1. A base do Transistor Q4 está sendo alimentada após R1 percebemos um Diodo D1 entre seu coletor e R3 e a existência deste Diodo se dá para que não haja retorno de corrente em R3 pois poderia ocasionar danos ao Resistor R3, por ultimo temos o Transistor Q2 recebendo sinal em sua base após o Diodo D1 e seu pino coletor após o Diodo D2, a existência de D2 se dá pela mesma função de D1, para que não ocorra retorno de corrente, o que poderia ocasionar danos aos componentes do circuito e no pino emissor de Q2 será ligado o polo positivo da bomba d'água, para sua alimentação e operação.

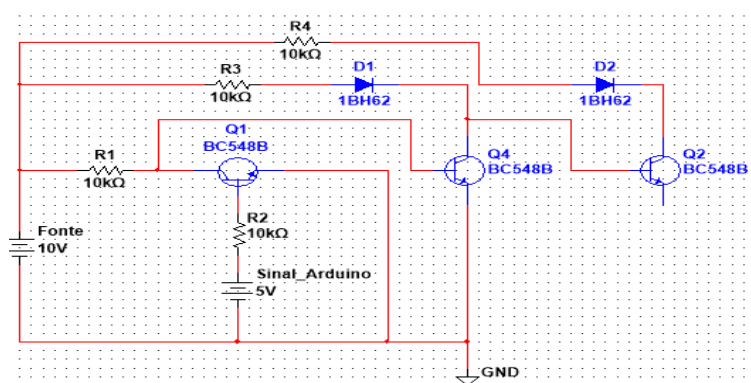


Figura 29: Circuito alimentação do Atuador.

Fonte: Autoral.

3.8 MONITORAMENTO

Para supervisionar e monitorar o sistema, foi desenvolvida uma tela que permite acompanhar e monitorar os processos no sistema. Além disso ela possui funções que permitem ajustar o valor de Setpoint e determinar se o mesmo poderá ser controlado remotamente ou localmente com um simples apertar de botão é possível alternar da função "local" para "remoto" onde "remoto" permite ao operador alteração no valor de Setpoint através da tela que estará configurada em seu computador. Já a opção local seria diretamente no ambiente controlado, por meio de um botão giratório o operador pode aumentar ou diminuir o valor de Setpoint acompanhado em um display o valor de Setpoint e o valor de nível, no mesmo display é possível saber se o ajuste do Setpoint está sendo realizado localmente ou remotamente, caso esteja na opção "remoto" e o operador queira ajustar localmente, basta pressionar o botão de emergência que o ajuste do Setpoint passa a ser realizado no local.

O computador não precisa estar conectado diretamente ao Controlador para que funcione a comunicação entre a tela e o Arduino, pois o protocolo de comunicação Modbus permite esta interação tanto conectado diretamente por um cabo ou por meios de transmissão como Ethernet, desde que esteja corretamente configurado. No caso desse projeto, a comunicação é realizada diretamente entre o Arduino e o computador onde a tela está instalada.

Na figura 30 é visualizado a tela de supervisão e monitoramento. Percebemos na imagem diversos blocos com seus títulos, onde o bloco de "Nível %"exibe o valor medido de nível no reservatório em porcentagem como a maioria dos blocos, ao lado está o bloco que representa o Setpoint, este mostra o valor ajustado de setpoint, tanto localmente quanto remotamente e para ajustar remotamente basta ativar a função remoto no botão "Ger. SetPoint"em seguida digitar o valor desejado no bloco "SetPoint %"que está logo acima do bloco "Bomba %"que mostra a porcentagem de operação do atuador que no caso é o da bomba d'água. No canto direito da imagem estão os blocos que exibem os valores obtidos dos controladores proporcional e integral junto ao bloco que mostra o valor do controlador PI.

Na tela ainda é possível visualizar dois gráficos, onde o gráfico "Erro %"exibe também em porcentagem o valor de erro existente. Nota-se que este varia de -100 a 100 e esta variação ocorre de acordo com o nível, se está abaixo ou acima do valor indicado no SetPoint. O outro gráfico é uma relação entre o nível e o Setpoint e através deste, é possível acompanhar a diferença de valores e o tempo que o sistema leva para acomodar e ajustar o nível controlado com o nível desejado.

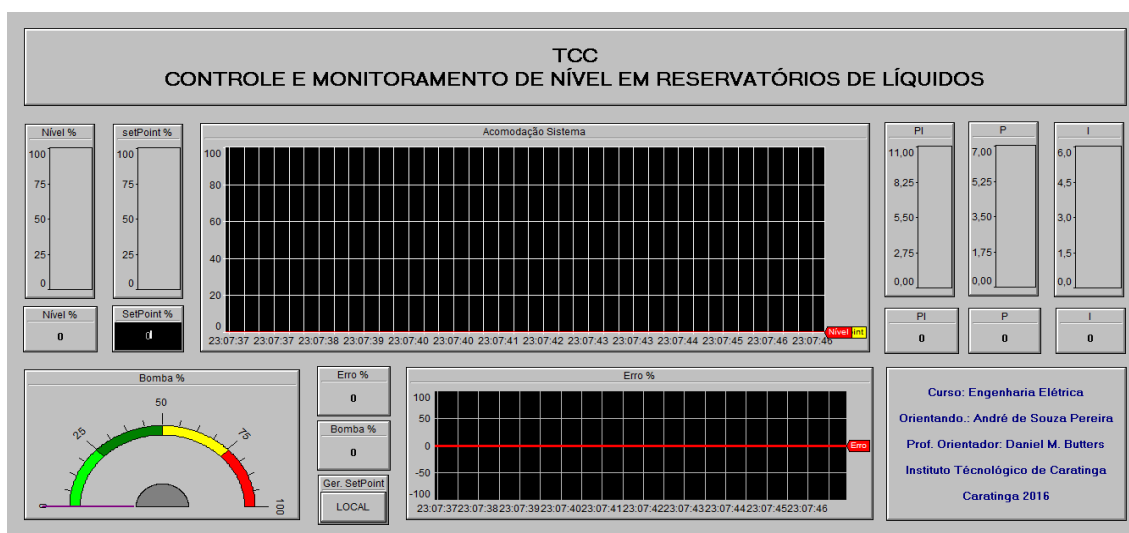


Figura 30: Tela de supervisão e monitoramento.

Fonte: Autoral.

Na figura 31 podemos observar a foto do display, exibindo valor de Setpoint, de nível e tipo de ajuste "LOCAL", na figura 32 podemos observar as mesmas informações, porém em vez de "LOCAL" temos "REMOTO".



Figura 31: Display Ajuste Local.



Figura 32: Display Ajuste Remoto.

Fonte: Autorais.

4 RESULTADOS E APLICAÇÕES

4.1 RESULTADOS

Após todo o processo teórico desenvolvido, foram realizadas práticas com o sistema projetado e no decorrer destas com o ajuste fino do ganho integral, o sistema ficou mais preciso e estável, respondendo de acordo com o previsto nas simulações do MATLAB.

Para realização do teste deixou-se o reservatório B com líquido em um nível próximo ao máximo e a válvula presente no reservatório A ficou totalmente aberta. O teste será a realização da drenagem do líquido presente no reservatório B para o A, enquanto o reservatório A abastece o B com capacidade máxima de vazão. Na figura 33 observamos no bloco "Acomodação Sistema" a inserção do degrau na posição 21 horas 09 minutos e 48 segundos, já na figura 34 observamos no mesmo bloco a eliminação do erro no sistema na posição 21 horas 10 minutos e 11 segundos, cerca de 23 segundos, foi o tempo acomodação do sistema, valor bem próximo ao da simulação no MATLAB.

Aplicação Elipse SCADA - Tela

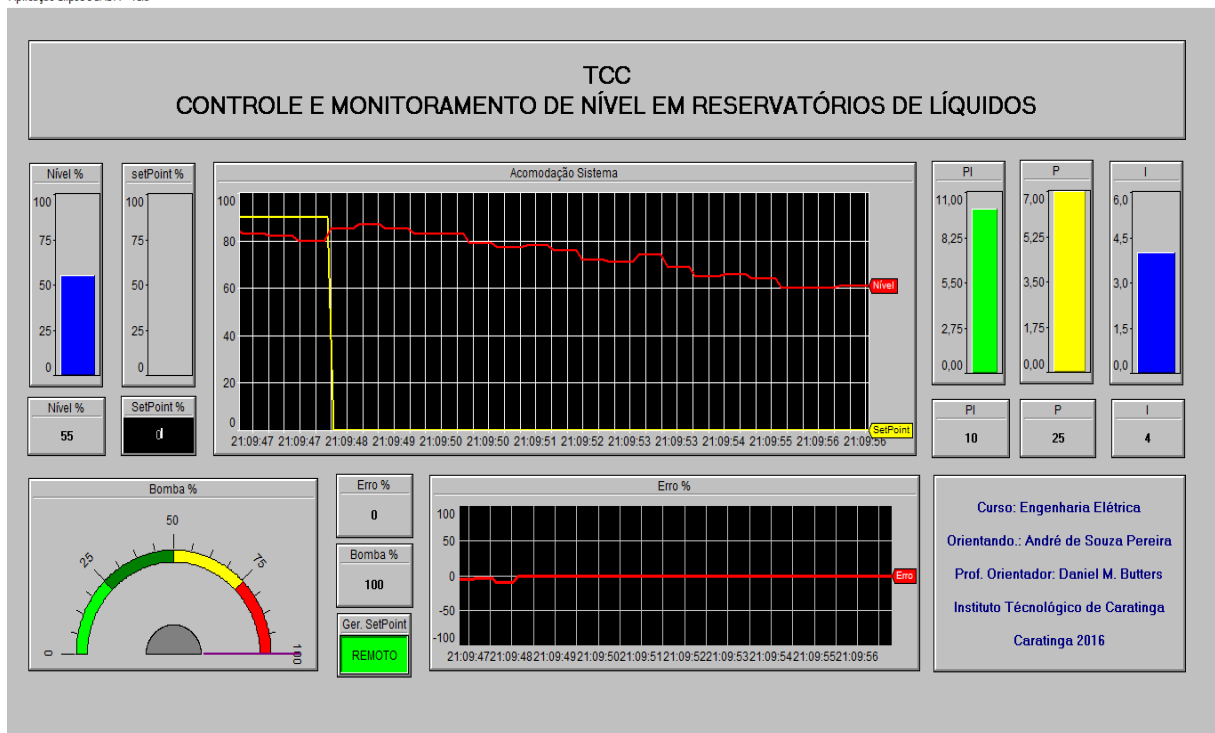


Figura 33: Início de teste.

Fonte: Autoral.

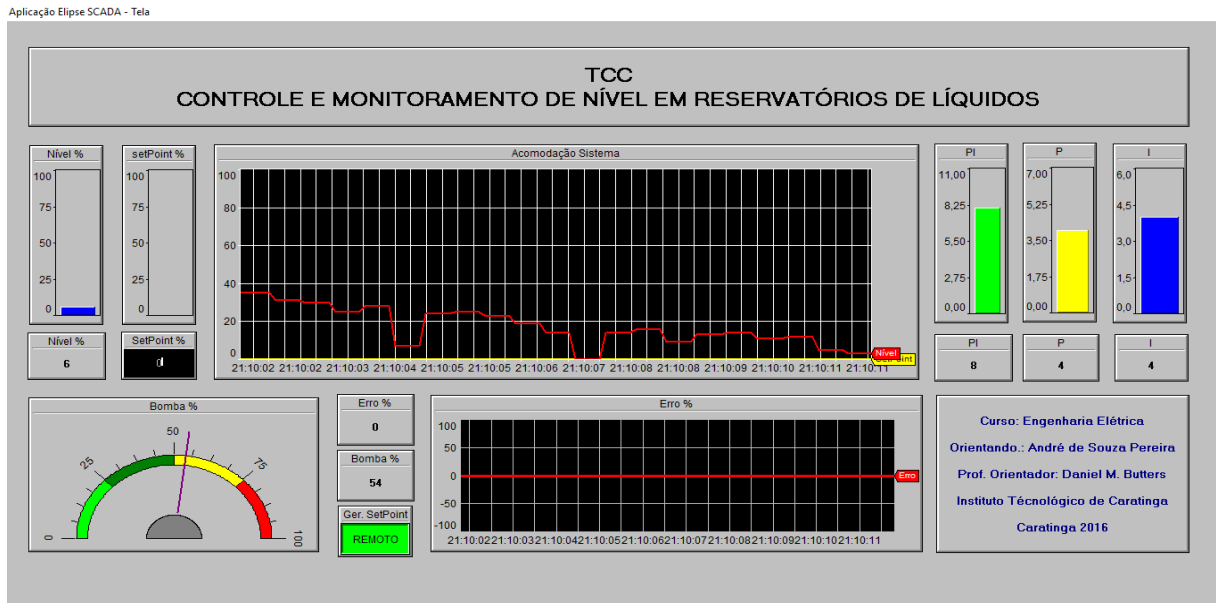


Figura 34: Final de teste.

Fonte: Autoral.

Na figura 35 podemos observar o sistema em funcionamento em regime permanente, observa-se que sua variação é mínima e sutil mantendo o nível de acordo com o ajustado.

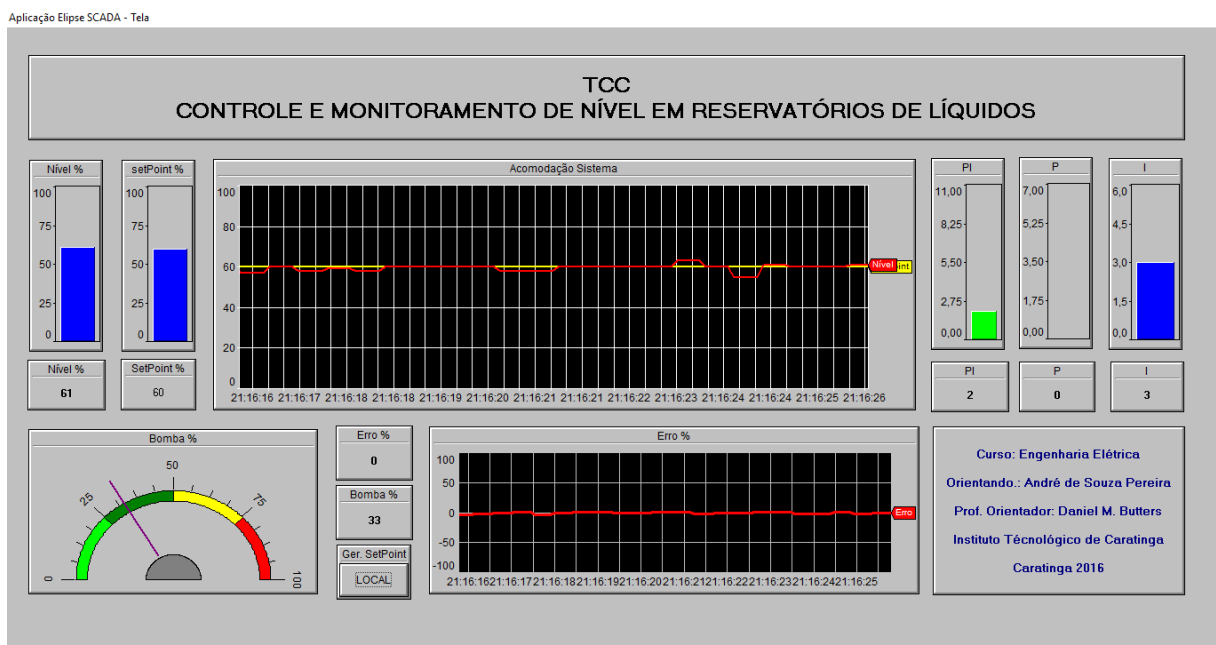


Figura 35: Teste em regime permanente.

Fonte: Autoral.

4.2 APLICAÇÕES

O sistema abordado teve sua utilização como controlador de nível em um reservatório de líquido, porém sua aplicação se estende além por causa de sua robustez, capacidade e precisão.

4.2.1 Saneamento urbano

Aplicando sistema de controle em estações de tratamento de água e/ou esgoto realizando o controle da vazão dos reservatórios conforme a demanda.

Em um sistema não controlado corretamente a bomba seria acionada para abastecer ou drenar líquido, como ela não é controlada por um sistema de controle sua partida será em 100% de sua capacidade desde o início até conseguir atingir seu valor de Setpoint onde seu interrompimento será de imediato, esta partida direta com força total e parada repentina durante operação causam danos ao equipamento reduzindo sua produtividade junto a sua vida útil gradativamente.

Com o sistema em funcionamento o controle aumentaria a potência do atuador gradativamente tentando alcançar o valor de Setpoint e reduz a mesma em medida que o erro diminui, tendo uma maior eficiência na distribuição do líquido e consumo de energia, evitando desperdício de potência e diminuindo o desgaste do atuador com consequência do aumento de sua vida útil.

4.2.2 Pequena Central Hidrelétrica

Pequena Central Hidrelétrica (PCH), é uma usina geradora de energia de pequeno porte, ela existe para ajudar a suprir a demanda energia elétrica, são caracterizadas como PCHs as usinas que possuem uma geração de energia entre 1 MW e 30 MW. Existem inúmeras destas centrais espalhadas por todo território nacional. Uma das característica destas centrais é operar a fio d'água, trabalhando de acordo com a vazão do rio onde estão localizadas.

Como a vazão dos rios variam de acordo com a época do ano, a geração destas usinas também irão variar, já que dependem diretamente dessa capacidade de vazão. A maioria destas PCHs tem um reservatório e de acordo com o nível do mesmo elas manipulam sua geração, podendo até mesmo cessar a geração a qualquer momento devido a baixa do nível do reservatório ocasionado pela diminuição da vazão do rio.

Como não podemos modificar a vazão do rio que abastece o reservatório, mas podemos controlar a vazão de saída que alimenta as turbinas, para obter um melhor aproveitamento na capacidade de geração. Ao controlar esta vazão de saída consequentemente alteramos a geração de energia elétrica, se a vazão ocasionada for superior a vazão do rio, logo o nível do reservatório irá abaixar, ao perceber isso o sistema de controle reduzirá a vazão de saída de acordo que o nível

estabilize no Setpoint e da mesma forma quando o nível aumentar a vazão de saída aumentará até igualar a de entrada para que o nível possa ficar no valor de Setpoint.

Claro que na usina existe limite de operação, tendo uma faixa de operação de trabalho que as turbinas podem suportar, os valores máximo e mínimo devem ser dimensionados como críticos no sistema de controle e quando estiverem próximo de serem alcançados, o sistema deve tomar as providências cabíveis. Por exemplo em um nível baixo a comporta deve fechar totalmente impedindo o fluxo de água até a turbina ocasionando seu interrompimento e parando a geração até que a vazão do rio esteja propensa novamente para seu religamento e ativação. Em caso da vazão ser superior a suportada pela turbina, o ideal é que a comporta pare de abrir deixando a turbina em uma operação segura e caso o nível continue a subir a integridade física da turbina não será comprometida, pois o fluxo de água pode sair pelas zonas de escoamento emergenciais das PCHs.

5 CONCLUSÃO

O trabalho mostrou que o emprego de Microcontrolador é eficiente para controle e monitoramento de nível em reservatórios além de outros processos e áreas como mencionado no capítulo 4.

A tela desenvolvida no Elipse Scada demonstrou ser de suma importância para controle. Poder acompanhar os processos e manipular variáveis ajuda muito no acompanhamento e gerenciamento do sistema, a possibilidade de poder alternar o ajuste de Setpoint entre local e remoto também se mostrou útil, principalmente em situações emergenciais que podem vir a surgir, outro recurso interessante do projeto é poder acompanhar o valor de nível e Setpoint por um display no local.

Foi observado durante a prática que a precisão do controle é ligada diretamente ao sensor de nível, no caso o sensor ultrassônico, este sensor demonstrou uma precisão boa em alturas mais elevadas níveis, mas em níveis baixos mostrou uma variação na leitura tendo menor precisão.

No geral o projeto conseguiu abranger diversas áreas e recursos, correspondendo de acordo com o esperado pela simulação. Vendo estes resultados percebemos a importância de poder dimensionar e modelar sistemas, realizando isso fica fácil de identificar falhas e corrigi-las, além de facilitar o emprego e dimensionamento de recursos que farão parte do sistema. Ressalvo que este trabalho trata-se de um protótipo e que o Arduino é apenas para utilização de estudos e teste pois não foi projetado para suportar longos períodos de trabalho. Para utilização e aplicação industrial deve ser construído e montado um circuito para acoplagem do Microcontrolador e os demais componentes necessários respeitando todas as normas vigentes sobre instalações industriais.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- BOYLESTAD, Robert L.; NASHELSKY, Louis. **Dispositivos eletrônicos e teoria de circuitos**. 8ª ed. São Paulo. Pearson, 2004.
- JOHNSON, David E.; Et al. **Fundamentos de análise de circuitos elétricos**. Rio de Janeiro. Prentice Hall, 1994.
- ELECTRONICA-PT. **Transistores**. [sd] Disponível em: <<http://www.electronica-pt.com/componentes-eletronicos/transistor-tipos>>. Acesso em 26 de Setembro 2016.
- JÚNIOR, Joab Silas da Silva. **Resistores**. 2016. Disponível em: <<http://mundoeducacao.bol.uol.com.br/fisica/resistores.htm/>>. Acesso em: 26 de Setembro de 2016.
- ELECTRONICA-PT. **Resistores**. [sd]. Disponível em: <<http://www.electronica-pt.com/codigo-cor-resistores-online>>. Acesso em 26 de Setembro de 2016.
- ELECTRONICA-PT. **Diodo**. [sd]. Disponível em: <<http://www.electronica-pt.com/diodo>>. Acesso em 26 de Setembro de 2016.
- NENOTECH. **Microcontrolador Atmega 2560**. [sd]. Disponível em: <<http://nenotecheletronica.com.br/Produto-CI-PLCC-ATMEGA2560-16AU-Microcontrolador-ATMEGA2560-16AU-TQFP100-Atmel-REF336-versao-579-579.aspx>>. Acesso em 01 de Outubro de 2016.
- PALMIERE, Sérgio Eduardo. **CLP versus Microcontrolador**. 2016. Disponível em: <<https://www.embarcados.com.br/clp-versus-microcontrolador/>>. Acesso em: 01 de Outubro de 2016.
- MONK, Simon. **Programação com Arduino**. 1ª ed. Porto Alegre. Bookman, 2013.
- ARDUINO. **Arduino MEGA 2560**. [sd]. Disponível em: <<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>>. Acesso em 02 de Outubro de 2016.
- SOUZA, Fábio. **Arduino - Saídas PWM**. 2014. Disponível em: <<https://www.embarcados.com.br/arduinosaídas-pwm/>>. Acesso em: 02 de Outubro de 2016.
- ARÁUJO, Thayron. **Sensor Ultrassônico com Arduino**. 2014. Disponível em: <<http://blog.fazedores.com/sensorultrassonico-com-arduino/>>. Acesso em: 27 de Setembro de 2016.
- OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 4ª ed. São Paulo. Prentice Hall, 2003.
- DORF, R. C.; BISHOP, R. H. **Sistemas de Controle Modernos**. 12ª ed. Rio de Janeiro. LTC, 2013.
- SILVA, Ana Paula Gonçalves da; SALVADOR, Marcelo. **O que são sistemas supervisórios?**. 2005. Disponível em: <http://www.wectrus.com.br/artigos/sist_superv.pdf>. Acesso em: 15 de Outubro de 2016.
- CARVALHO, Gregor Gama. **Supervisório ELIPSE SCADA - Introdução**. 2012. Disponível

em: <<http://automacao-criativa.blogspot.com.br/2012/07/supervisorio-ellipse-scada-introducao.html>>. Acesso em: 15 de Outubro de 2016.

FREITAS, Carlos Márcio. **CLP versus Microcontrolador**. 2014. Disponível em: <<https://www.embarcados.com.br/protocolo-modbus/>>. Acesso em: 01 de Outubro de 2016.

HALLIDAY, David.; Et al. **Fundamentos de física - Volume 2**. 8^a ed. Rio de Janeiro: LTC, 2008.

EMPRESA VIDA DE SILÍCIO. **Sensor Distância Ultrassônico HCSR04**. [sd]. Disponível em: <<http://www.vidadesilicio.com.br/ultrassom-hc-sr04.html>>. Acesso em: 07 de Outubro de 2016.

MATLAB. **MATLAB**. [sd]. Disponível em: <<https://www.mathworks.com/products/matlab/>>. Acesso em 20 de Setembro de 2016.

ELIPSE SOFTWARE. **Downloads**. [sd]. Disponível em: <<http://www.ellipse.com.br/downloads/>>. Acesso em 24 de Setembro de 2016.

UNICAMP. **Tabela: Transformada de Laplace**. [sd]. Disponível em: <<http://www.ime.unicamp.br/~msantos/tab-laplace>>. Acesso em 24 de Outubro de 2016.

APÊNDICE A APÊNDICE

CÓDIGO ARDUINO: Programação de própria autoria.

```
#include <Modbusino.h>

#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>

Adafruit_PCD8544 display = Adafruit_PCD8544(8, 9, 10, 11, 12);

/*****
  ARDUINO – COMUNICACAO SERIAL (MODBUS)
  https://github.com/stephane/modbusino
  *****/

/* IDENTIFICACAO DO DISPOSITIVO NO PROTOCOLO MODBUS*/
ModbusinoSlave modbusino_slave(1);

/*****
  NUMERO DE REGISTRADORES MODBUS
  *****/
uint16_t tab_reg[10];

/*****
  CONFIGURACOES SENSOR ULTRASONICO
  *****/
#include <SoftwareSerial.h>
#define rxPin 0
#define txPin 1
#define ledPin 13
byte pinEstado = 0;
SoftwareSerial miSerial = SoftwareSerial(rxPin, txPin);
const int trigger=53;
const int echo=52;
double distance;

/*****
  GANHOS
  *****/
double kP = 2.95,
       kI = 0.792,
       kD = 0;
//kD = 0.0348;

/*****
  VARIAVEIS DE PROCESSO
  *****/

double pCONTROLE=5,
       erro,
       error,
       nivell=0.16,
       lasterror=0,
```

```

    setPoint ,
    setPoint2 ,
    setPoint3 ,
    lastProcess=0,
    deltaTime ,

    nivelr1 ,

    pPOTENC=A1,
    potenc=0,

    GR,

    rxloc=37,

    P = 0,
    I = 0,
    D = 0,

    PID = 0,
    PID2 = 0,
    soma ,
    controlePwm = 0;

    int i;

/*****
                                VARIAVEIS TELA DE MONITORAMENTO
*****/
double setPoint1 ,
    nivelr ,
    error1 ,
    P1 ,
    I1 ,
    D1 ,
    PID1 ,
    controlePwm1 ,
    gerSetPoint;

/*****
                                SETUP
*****/
void setup(){
    modbusino_slave . setup(9600);

//ULTRASONICO RESEVATORIO CONTROLADO
    pinMode( trigger ,OUTPUT);
    pinMode(echo ,INPUT);

    pinMode(rxPin , INPUT);
    pinMode(txPin , OUTPUT);

//CONTROLE PID
    pinMode(pCONTROLE, OUTPUT);

    Serial . begin(9600);

```

```

miSerial.begin(9600);

//POTENCIOMETRO NIVEL
pinMode(pPOTENC, INPUT);

//BOTAO LOCAL
pinMode(rxloc, INPUT);

/*****
                                DISPLAY
*****/
display.begin();
// AJUSTA O CONTRASTE DO DISPLAY
display.setContrast(48);
// APAGA O BUFFER E O DISPLAY
display.clearDisplay();
// DEFINE TAMANHO E COR DO TEXTO
display.setTextSize(1);
display.setTextColor(BLACK);

// RETANGULO TEMPERATURA
display.drawRoundRect(0,0, 44,24, 3, 2);
// TEXTO INICIAL TEMPERATURA
display.setCursor(11,3); // Seta a posição do cursor
display.println("SetP");
display.setCursor(5,14);
display.println("----");
display.setCursor(29,14);
display.drawCircle(31, 15, 1,1);
display.println("%");

// RETANGULO UNIDADE
display.drawRoundRect(45,0, 39,24, 3, 2);
// TESTO INICIAL NIVEL
display.setCursor(52,3);
display.println("NIVL");
display.setCursor(50,14);
display.println("----");
display.setCursor(75,13);
display.println("%");

// RETANGULO AJUSTE
display.drawRoundRect(0,25, 84,23, 3, 2);
// TEXTO INICIAL AJUSTE
display.setCursor(22,28);
display.println("AJUSTE");
display.setCursor(11,38);
display.println("-----");
display.display();
}

/*****
                                INICIO LOOP
*****/
void loop(){
  while(1)
  {

```

```

/*****
MEDINDO A DISTANCIA COM SENSOR ULTRASSONICO
*****/

    for (i=0; i<=5; i++){
        digitalWrite (trigger ,LOW);
        delayMicroseconds (5);
//ENVIANDO SINHAL
        digitalWrite (trigger ,HIGH);
        delayMicroseconds (10);
        digitalWrite (trigger ,LOW);
// RECEBENDO SINHAL E CONVERTENDO EM METROS
        distance=pulseIn (echo ,HIGH); // MEDIDA EM ECHO POR PULSO

        distance=(distance*0.0001657) + 0.001; //CONVERTENDO EM METROS

            soma += distance;
            delay (3);
        }

        distance = soma/i;
        soma=0;

/*****
SETPOINT
*****/
gerSetPoint = tab_reg [0];
GR = gerSetPoint;

if (digitalRead (rxloc) == HIGH){
    GR = 0;
    gerSetPoint = GR;
    tab_reg [0] = gerSetPoint;
}

if (GR == 0 || GR == 1){
if (GR == 1){
    setPoint1 = tab_reg [1];
    display.fillRect (4, 37, 46 , 10, 0);
    display.setCursor (11,38);
    display.println ("REMOTO");
    if (setPoint1 >= 95){
        setPoint1 = 95;
        tab_reg [1] = setPoint1;
    }
    setPoint = (setPoint1*nivell)/100;
}

else if (GR == 0){
    gerSetPoint = 0;
    tab_reg [0] = gerSetPoint;
    display.fillRect (4, 37, 46 , 10, 0);
    display.setCursor (11,38);
    display.println ("LOCAL");
    potenc=analogRead (pPOTENC);
    setPoint2 = map (potenc , 0.0, 1023.0, 0.0, 10000.0);
}
}

```

```

    setPoint = setPoint2/62500;
    if (setPoint >= 95){
    setPoint = 95;
    }
    setPoint1 = (setPoint/nivel1) * 100.0;
    tab_reg[1] = setPoint1;
}
}

/*****
                DEFININDO VALOR DE NIVEL E ERRO
*****/
//NIVEL
    nivelr1 = (nivel1 - distance);

//ERRO – DIFERENCA ENTRE SETPOINT E NIVEL/////////
    erro = setPoint - nivelr1;
    error = (erro * 100.0)* (-1);

/*****
                CONTROLE PID
*****/
    double deltaTime = (millis() - (lastProcess));
    lastProcess = millis();

//P
    P = error * kP;

//I
    I = I + (error * kI) * (deltaTime/1000.0);
    if(I > 4.5){
        I = 4.5;
    }
    if(I < 0){
        I = 0;
    }

//D
    D = ((error - lasterror) * kD) / (deltaTime/1000.0);
    lasterror = error;

/*****
                SOMA E DEFINICAO PID
*****/
    PID = P + I;
    if(PID > 10.00){
        PID = 10.00;
    }
    if(PID < 0){
        PID = 0;
    }

/*****
                CONTROLE PWM
*****/
    PID2 = PID*1000.00;

```

```

controlePwm = map(PID2,0.00,10000.00,0.00,255.00);
  if(controlePwm <= 21){
    controlePwm = 21.0;
  }
  if(controlePwm >= 255.0){
    controlePwm = 255.0;
  }
  analogWrite(pCONTROLE, controlePwm);

/*****
          INFORMACOES PARA TELA DE MONITORAMENTO
*****/
setPoint3 = setPoint1;

nivelr = (nivelr1/nivel1) * 100.0;
  if(error <= 0){
    error1 = (erro /setPoint) * -100.0;
  }
  else {
    error1 = (((erro* (-1))/setPoint) * 100.0);
  }

P1 = P;
I1 = I;
D1 = D;
PID1 = PID;

  if(controlePwm > 21){
    controlePwm1 = (controlePwm/255.0) * 100.0;
  }
  else{
    controlePwm1 = 0;
  }

tab_reg[2] = error1;
tab_reg[3] = P1;
tab_reg[4] = I1;
tab_reg[5] = D1;
tab_reg[6] = PID1;
tab_reg[7] = controlePwm1;
tab_reg[8] = nivelr;
tab_reg[9] = setPoint3;

//ATUALIZA O VALOR DE SETPOINT NO DISLAY
display.fillRect(4,13, 25 , 10, 0);
display.setCursor(5,14);
  display.println(setPoint3 ,1);

//ATUALIZA O VALOR DE NIVEL NO DISLAY
display.fillRect(50,13, 23 , 10, 0);
display.setCursor(50,14);
  display.println(nivelr ,1);

  display.display();
/*****

```

FIM DOS PROCESSOS


```
*****/
  modbusino_slave.loop(tab_reg , 10); //IGUAL A NUMERO DE REGISTRADORES MODBUS
}
}
```

ANEXO A TABELA DE LAPLACE

1	$\frac{1}{s}$
e^{at}	$\frac{1}{s - a}$
t^n	$\frac{n!}{s^{n+1}}$
$\text{sen } at$	$\frac{a}{s^2 + a^2}$
$\text{cos } at$	$\frac{s}{s^2 + a^2}$
$\text{senh } at$	$\frac{a}{s^2 - a^2}$
$\text{cosh } at$	$\frac{s}{s^2 - a^2}$
$e^{at} \text{sen } bt$	$\frac{b}{(s - a)^2 + b^2}$
$e^{at} \text{cos } bt$	$\frac{s - a}{(s - a)^2 + b^2}$
$t^n e^{at}$	$\frac{n!}{(s - a)^{n+1}}$
$U_c(t)$	$\frac{e^{-cs}}{s}$
$U_c(t)f(t - c)$	$e^{-cs}F(s)$
$e^{ct}f(t)$	$F(s - c)$
$f(ct)$	$\frac{1}{c}F\left(\frac{s}{c}\right)$
$\int_0^t f(t - \tau)g(\tau)d\tau$	$F(s)G(s)$
$\delta(t - c)$	e^{-cs}
$f^{(n)}(t)$	$s^n F(s) - s^{(n-1)}f(0) - \dots - f^{(n-1)}(0)$
$(-t)^n f(t)$	$F^{(n)}(s)$

Figura 36: Tabela de Laplace.

Fonte: Unicamp.