

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**



IZAAC VIEIRA JARDIM

**SISTEMA DIDÁTICO DE CONTROLE DE NÍVEL EM
TANQUES ACOPLADOS**

VITÓRIA – ES
Julho, 2019

IZAAC VIEIRA JARDIM

**SISTEMA DIDÁTICO DE CONTROLE DE NÍVEL EM TANQUES
ACOPLADOS**

Parte manuscrita da Proposta de Projeto de Graduação do aluno **Izaac Vieira Jardim**, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para o grau de Engenheiro Eletricista.

Orientador: Prof. Dr. José Leandro Félix Salles

VITÓRIA – ES
Julho, 2019

IZAAC VIEIRA JARDIM

SISTEMA DIDÁTICO DE CONTROLE DE NÍVEL EM TANQUES ACOPLADOS

Parte manuscrita do Projeto de Graduação do aluno **Izaac Vieira Jardim**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovada em 20, de julho de 2019.

COMISSÃO EXAMINADORA:

Prof. Dr. José Leandro Félix Salles
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Walbermark Marques dos Santos
Universidade Federal do Espírito Santo
Examinador

Prof. Me. Rogério Passos do Amaral Pereira
Instituto Federal do Espírito Santo - Serra
Examinador

AGRADECIMENTOS

Agradeço sobre todas as coisas a Deus. Foi Ele que me sustentou até aqui e me proporcionou trilhar nesta etapa da minha vida sem me desviar dos meus princípios e propósitos. Toda glória e honra seja dada a Ele.

Toda a minha estima e consideração a Brenda Kelly Peres Franco, minha amada e querida esposa, que me incentivou e me apoiou durante toda essa trajetória. Agradeço pelo companherismo, paciência, persistência e por sua dedicação. Espero poder recompesá-la durante todos os anos da minha vida por tudo o que representa para mim.

A minha mãe, Valdirene Vieira e aos meus queridos irmãos, Kleber Henrique Vieira Jardim e Lorrainy Vieira Jardim, que me fizeram acreditar neste sonho e me deram todo apoio e recursos que foram precisos.

Aos professores Dr. José Leandro Félix Salles, Dr. Celso José Munaro e Dr. Alessandro Mattedi, que me deram todo embasamento técnico necessário, além de terem proporcionado o incentivo para o início e finalização deste projeto.

Aos meu sogro, Carlos Magno Franco, e minha sogra Márcia Andréia Bicalho Peres, que cuidaram de mim como filho. Minha gratidão é enorme.

Aos grandes amigos que oraram e me orientaram a permanecer firme, mesmo com tantas dificuldades. Agradeço a todos pela preocupação e apoio.

Este projeto não é apenas uma conquista pessoal, se tornou uma grande vitória para todos os que estão ao meu redor e que foram indispensáveis até aqui. Eu sou grato a todos!

RESUMO

O conhecimento é fundamentado em preceitos técnicos e experiências vividas durante todo o percurso de um estudante. O conteúdo teórico sempre é melhor absorvido quando aplicado em paralelo ao conteúdo prático. O processo de controle de nível é bastante adequado para ser utilizado como uma ferramenta didática, pois une o conhecimento teórico e prático, além de ser um mecanismo comum em diversos ramos industriais. O objetivo do trabalho foi a implementação de um sistema de controle de nível em tanques acoplados, utilizando o microcontrolador ATMEGA328, na plataforma ARDUINO UNO para as entradas analógica e digital e o *software* MATLAB®, onde foi implementado o controle PID digital, responsável pelo controle da planta, e uma tela de supervisão para visualização das variáveis do processo. O sistema foi linearizado devido as não-linearidades presentes em sistemas de controle de nível que proporcionam grandes desafios, necessitando assim de um controlador mais complexo. As simulações nortearam todo o desenvolvimento do projeto, tornando-o mais adequado para o objetivo de estudo de automação e controle.

Palavras-chave: Controle de nível, conhecimento teórico e prático, microcontrolador ATMEGA328, não-linearidades, tela de supervisão.

ABSTRACT

The knowledge is based on technical precepts and experiences lived throughout the course of a student. The theoretical content is always better absorbed when applied in parallel to the practical content. The level control process is well suited to be used as a didactic tool, as it unites theoretical and practical knowledge, as well as being a common mechanism in several industrial branches. The objective of the work was the implementation of a level control system in coupled tanks, using the ATMEGA328 microcontroller, the ARDUINO UNO platform for the analogue and digital inputs and the MATLAB® software, where the digital PID control was implemented, responsible for plant control, and a supervisory screen for visualization of process variables. The system was linearized due to the nonlinearities present in level control systems that provide great challenges, thus requiring a more complex controller. The simulations guided the entire development of the project, making it more suitable for the study object of automation and control.

Key words: Level control, theoretical and practical knowledge, ATMEGA328 microcontroller, nonlinearities, supervisory screen.

LISTA DE FIGURAS

Figura 1 - Sistema de tanques acoplados.....	14
Figura 2 - Comparação entre o valor desejado e o valor de processo.....	20
Figura 3 - Estrutura de controle proporcional e integral	22
Figura 4 - Estrutura de controle proporcional derivativo	22
Figura 5 – Estrutura do controlador PID.....	23
Figura 6 - Controlador em malha aberta	24
Figura 7 - Controlador em malha fechada.....	25
Figura 8 - Resposta de sistema de primeira ordem a uma entrada degrau	26
Figura 9 - Resposta de sistema de segunda ordem a uma entrada degrau	27
Figura 10 - Resposta de segunda ordem em função do fator de amortecimento	28
Figura 11 - Sistema em malha fechada com ganho K.....	29
Figura 12 - Diagrama de bloco da malha de controle	31
Figura 13 - Teste de resistência da válvula V12.....	34
Figura 14 - Diagrama de blocos do sistema para teste de resistência da válvula V ₁₂	34
Figura 15 - Comparação da curva de nível do líquido x tempo obtida via simulação e via testes experimentais	35
Figura 16 – Dinâmica do sistema monotanque para R12 igual a 3500.....	36
Figura 17 - diagrama de blocos do sistema monotanque	37
Figura 18 – Dinâmica do sistema de dois tanques acoplados para Q _p = 2L/min durante 40s.....	38
Figura 19 - diagrama de blocos do sistema de dois tanques acoplados	39
Figura 20 - Sensor de pressão MPX5010DP	40
Figura 21- Circuito filtro passa-baixa para saída do sensor MPX5010DP	40
Figura 22 - Sensor de vazão de 1/2" YF-S201	41
Figura 23 - Bomba submersa 240L/h.....	42
Figura 24 - CI L298N com dois circuitos ponte H	43
Figura 25 - Essquemático circuito de acionamento da bomba submersa.....	44
Figura 26 - Representação Sinal PWM.....	46
Figura 27 - ARDUINO UNO	47
Figura 28 - Diagrama de comunicação ARDUINO UNO - MATLAB®	47
Figura 29 - Tela de Supervisório do Sistema de tanques acoplados.....	50
Figura 30 - Protótipo desenvolvido.....	52
Figura 31 - Teste experimental do sistema para uma entrada de 2L/min	53

Figura 32 - Comparação entre teste experimental e simulação para uma entrada de 2L/min	54
Figura 33 - Resposta do sistema em malha aberta.....	55
Figura 34 - Resposta do sistema em malha fechada	56
Figura 35 - Lugar geométrico da raízes	56
Figura 36 – Projeto controlador PI com $M_p = 7-15\%$ e $t_s = 30-90s$	57
Figura 37 - Projeto do controlador PID.....	58
Figura 38 - Teste experimental 1 do controlador PI	59
Figura 39 - Teste experimental 2 do controlador PI	60
Figura 40 - Teste experimental do controlador PID	60
Figura 41 - Comparação entre as respostas dos controladores PI e PID.....	61

LISTA DE TABELAS

Tabela 1 - Variação do nível do líquido em função da área da base do tanque	33
Tabela 2- Especificações técnicas do microcontrolador ARDUINO UNO	48
Tabela 3- Relação do custo dos componentes	52
Tabela 4 - Dados para simulação	53

LISTA DE ABREVIATURA E SIGLAS

<i>A/D</i>	Conversor Analógico-Digital
<i>AC/DC</i>	Alternating Current/Direct Current
<i>AREF</i>	Analog Referenc
<i>CI</i>	Circuito Integrado
<i>DC</i>	Duty Cicle
<i>DP</i>	Differential Pressure
<i>E/S</i>	Entrada/Saída
<i>EEPROM</i>	Electrically-Erasable Programmable Read-Only Memory
<i>GND</i>	Ground
<i>GUIDE</i>	Graphical User Interface Development Environment
<i>IDE</i>	Integrated Development Environment
<i>IOREF</i>	Input Output Reference
<i>IPS</i>	Sensores de Pressão Integrados
<i>LED</i>	Light Emmitting Diode
<i>LGR</i>	Lugar Geométrico das Raízes
<i>MA</i>	Malha Aberta
<i>MF</i>	Malha Fechada
<i>MATLAB</i>	MATrix LABoratory
<i>PID</i>	Proporcional Integral Derivativo
<i>PWM</i>	Pulse Width Modulation
<i>SCL</i>	Serial Clock Line
<i>SDA</i>	Serial Data Line
<i>SP</i>	Set Point
<i>SPI</i>	Serial Peripheral Interface
<i>SRAM</i>	Static Random Access Memory
<i>TTL</i>	Transistor–Transistor Logic
<i>TWI</i>	Two-Wire Interface
<i>UART</i>	Universal Asynchronous Receiver/Transmitter
<i>UFES</i>	Universidade Federal do Espírito Santo
<i>USB</i>	Universal Serial Bus

SUMÁRIO

1	INTRODUÇÃO	12
2	OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS	13
2.1	Objetivo Geral	13
2.2	Objetivos Específicos	13
3	EMBASAMENTO TEÓRICO	14
3.1	Modelagem matemática	14
3.2	Modelagem do sistema monotanque	15
3.3	Modelagem do sistema de dois tanques acoplados	16
3.4	Linearização do sistema	18
3.5	Sistemas de Controle	20
3.5.1	Controle proporcional	21
3.5.2	Controle proporcional-integral	22
3.5.3	Controle proporcional-derivativo	22
3.5.4	Controle proporcional-integral-derivativo	23
3.6	Malha aberta e fechada de sistemas de controle	24
3.6.1	Sistema de controle em malha aberta	24
3.6.2	Sistema de controle em malha fechada	24
3.7	Resposta de sistemas de primeira e segunda ordem	26
3.7.1	Sistema de primeira ordem	26
3.7.2	Sistema de segunda ordem	27
3.8	PID digital	28
3.9	Lugar das raízes	29
4	ESTRUTURA DO SISTEMA	31
4.1	Malha de controle	31
4.2	Dimensionamento dos tanques	32
4.3	Testes iniciais	33
4.3.1	Resistência da válvula V_{12}	33
4.3.2	Simulação sistema monotanque	35
4.3.3	Simulação sistema de dois tanques acoplados	37
4.4	Hardware do sistema de tanques acoplados	39

4.4.1 Sensor de pressão MPX5010DP.....	39
4.4.2 Sensor de vazão	40
4.4.3 Bomba submersa 12V	42
4.4.4 CI L298N	43
4.4.5 Modulação PWM.....	45
4.4.6 ARDUINO UNO	46
4.5 <i>Software</i> do sistema de tanques acoplados	48
4.5.1 Interface Gráfica	50
5 RESULTADOS EXPERIMENTAIS	53
5.1 Validação do modelo do sistema.....	53
5.2 Projeto dos controladores	55
5.2.1 Projeto do controlador PI.....	57
5.2.2 Projeto do controlador PID	58
5.3 Resultados do sistema de controle	58
6 CONCLUSÃO.....	62
REFERÊNCIAS BIBLIOGRÁFICAS	64
APÊNDICE A – ALGORITMO DE AQUISIÇÃO DE DADOS - ARDUINO	66
APÊNDICE B – ALGORITMO DE CONTROLE E INTERFACE - MATLAB.....	70

1 INTRODUÇÃO

O processo de controle de um sistema é fundamental para se obter valores desejados de saída para uma determinada variável de entrada. Dentre muitos processos de controle pode-se destacar o de temperatura, vazão, velocidade e o de nível. Tais processos são, por muitas vezes, estudados nos cursos de engenharia elétrica e engenharia de automação. Isto ocorre porque estes sistemas estão presentes de forma ampla no setor industrial. O controle de nível, por exemplo, é bastante comum nas indústrias dos ramos químico, petroquímico, de celulose e de alimentos (GOSMANN, 2002).

Estudar estes processos e determinar a melhor forma de controle é, por muitas vezes, complexo e por isso necessita de um elevado conhecimento teórico e prático. Um grande limitador é o conhecimento prático adquirido por um aluno durante a graduação. Estes experimentos são importantes porque fazem com que “[...] os estudantes enfrentem situações como a presença de não-linearidade e sinais ruidosos, permitindo a eles adquirirem experiência no tratamento destes fenômenos” (RAMOS e WENSE, 2008).

O conhecimento prático é adquirido em cursos laboratoriais que, de forma geral, necessitam de uma enorme quantidade de recursos (RAMOS e WENSE, 2008). Isto inviabiliza, por muitas vezes, a instalação destes laboratórios em algumas instituições de ensino. Pode-se perceber o problema que diversos cursos de engenharia enfrentam, em especial as áreas de controle de sistemas. Nestas áreas os fenômenos e as idéias abordados possuem características complexas, o que determina a necessidade de serem demonstradas experimentalmente.

O processo de controle de nível é bastante adequado para ser utilizado como uma ferramenta didática, pois une o conhecimento teórico e prático, além de ser um mecanismo comum em diversos ramos industriais (GOSMANN, 2002). Este processo tem uma dinâmica não-linear, o que possibilita o estudante aplicar o conhecimento teórico em um sistema real, observando as possíveis limitações de técnicas de controle.

2 OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS

A implementação de um sistema didático de controle em tanques acoplados permite a aplicação de diferentes controladores de forma rápida e de fácil construção. Estes sistemas possuem modelagem simples e não necessitam de uma elevada quantidade de recursos materiais e financeiros. A utilização deste recurso para a experimentação presencial garante aos alunos interagirem diretamente com sistemas reais e adquirirem uma valiosa visão prática de engenharia de controle de processos.

2.1 Objetivo Geral

O objetivo geral do trabalho será a implementação de um sistema de controle de nível em tanques acoplados, com arquitetura de controle baseada em microcontrolador ARDUINO UNO, usando comunicação analógica e digital, através do *software* MATLAB®.

2.2 Objetivos Específicos

- Analisar a modelagem de sistemas de tanques acoplados e obter o modelo não-linear do processo que será implementado;
- Especificar as dimensões dos tanques para utilização do modelo em laboratório;
- Especificar os dispositivos que serão utilizados para se obter um sistema relativamente barato e suficientemente desejável para controle;
- Implementar a arquitetura de aquisição de dados baseada no microcontrolador ATMEGA328, na plataforma ARDUINO UNO;
- Especificar os módulos de controle e atuadores;
- Desenvolver uma tela de supervisor para visualização das variáveis do processo através do *software* MATLAB®;
- Aplicar os controladores PI e PID e comparar os seus resultados.

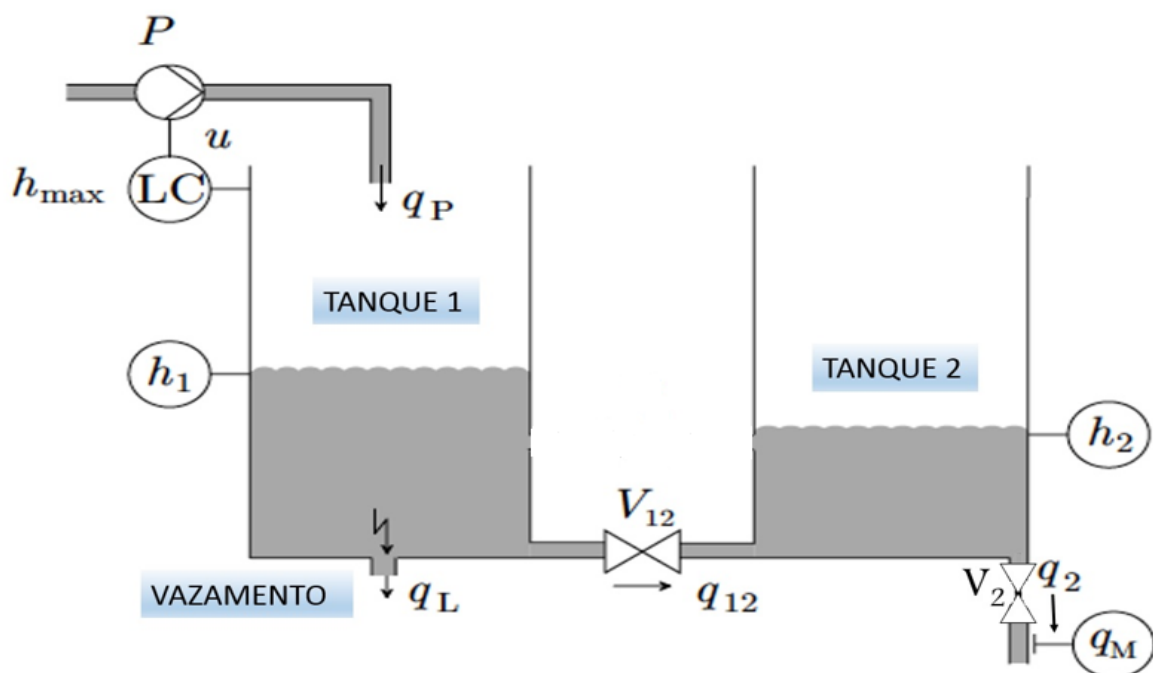
3 EMBASAMENTO TEÓRICO

Por se tratar de um procedimento comum em controle de níveis, o sistema de tanques acoplados será descrito a seguir, como uma abordagem no ensino e pesquisa de técnicas de controle de processos. Os modelos matemáticos que descrevem o sistema referido neste trabalho serão apresentados, assim como o método de linearização do modelo.

3.1 Modelagem matemática

O sistema a ser implementado neste trabalho é mostrado na figura 1. Este processo apresenta o acoplamento de dois tanques, onde será realizado o controle do nível do tanque 2.

Figura 1 - Sistema de tanques acoplados



Fonte: BLANKE, 2006.

Onde h_1 e h_2 são as alturas, em cm, do nível do líquido nos tanques 1 e 2, respectivamente. As vazões, em cm^3/s , são representadas por q_p , vazão de entrada no tanque 1, q_{12} , vazão de entrada no tanque 2, q_L representa um possível distúrbio como um vazamento de água do tanque 1 e q_2 é a vazão de saída do sistema. A válvula V_{12} é acionada para liberar a passagem de líquido entre os tanques. A válvula V_2 é acionada para liberar a saída do líquido do tanque 2 para o reservatório.

3.2 Modelagem do sistema monotanque

Considerando apenas um tanque do sistema descrito na figura 1, tem-se o balanço de massa para o fluxo laminar e turbulento, onde este primeiro fluxo descreve um modelo matemático mais simplificado, enquanto que o fluxo turbulento apresenta um modelo matemático mais complexo, evidenciando equações não-lineares.

Para o fluxo laminar é apresentado abaixo o modelo matemático do sistema com apenas um tanque.

$$Ar_1 \frac{dh_1}{dt} = q_p - q_{12} \quad (3.1)$$

Onde Ar_1 é a área da base do tanque. Para este caso, a relação entre a vazão de saída q_{12} e a altura h_1 é dada pela equação abaixo.

$$q_{12} = \frac{h_1}{R_1} \quad (3.2)$$

Onde R_1 é a resistência da válvula de saída. Realizando a substituição da equação 3.2 na equação 3.1 e aplicando a transformada de Laplace, tem-se a equação 3.3, que descreve a relação entre a entrada q_p e a saída h_1 .

$$\frac{H_1(s)}{Q_p(s)} = \frac{R_1}{R_1 Ar_1 s + 1} \quad (3.3)$$

Se a grandeza de entrada for q_p e a grandeza de saída for a vazão q_{12} , tem-se a equação 3.4.

$$\frac{Q_{12}(s)}{Q_p(s)} = \frac{1}{R_1 Ar_1 s + 1} \quad (3.4)$$

Considerando o fluxo como turbulento, a relação entre a altura do tanque e a vazão é dada pelo método de Bernoulli, descrito na equação 3.5.

$$q = k\sqrt{h} \quad (3.5)$$

Onde o valor k , em $\text{cm}^{2.5}/\text{s}$, depende de algumas variáveis, tais como o coeficiente de descarga do escoamento, da área de seção transversal do tanque, da área do orifício de saída e da aceleração da gravidade (GOSMANN, 2002).

Por se tratar de um fluxo turbulento, as equações diferenciais relacionadas a este sistema são não-lineares (OGATA, 2003, p.125), o que se pode observar na equação 3.5 descrita acima. A partir da equação 3.1, aplica-se a relação que existe entre a altura do nível do tanque h e a vazão de saída q . Obtem-se a equação 3.6 que descreve o sistema de 1 tanque com fluxo turbulento.

$$A_{r1} \frac{dh_1}{dt} = q_p - k_{12} \sqrt{h_1} \quad (3.6)$$

3.3 Modelagem do sistema de dois tanques acoplados

Considerando o sistema completo da figura 1, tem-se abaixo a modelagem com dois tanques acoplados para o fluxo laminar e turbulento. Utilizou-se as equações 3.1 a 3.6 para se obter as equações do sistema completo.

Para o fluxo laminar, tem-se as equações 3.7 e 3.8 que descrevem a modelagem do primeiro tanque e as equações 3.9 e 3.10 que descrevem a modelagem para o segundo tanque. Para este caso não considerou-se as perturbações.

$$A_{r1} \frac{dh_1}{dt} = q_p - q_{12} \quad (3.7)$$

$$\frac{h_1 - h_2}{R_1} = q_{12} \quad (3.8)$$

Para o tanque 2, tem-se que a vazão de entrada é q_{12} e a vazão de saída é q_2 . Logo, obtem-se as equações 3.9 e 3.10 que descrevem o sistema.

$$A_{r2} \frac{dh_2}{dt} = q_{12} - q_2 \quad (3.9)$$

$$\frac{h_2}{R_2} = q_2 \quad (3.10)$$

Aplicando a transformada de Laplace nas equações 3.7 a 3.10 e definindo a grandeza de entrada como a vazão q_p e a grandeza de saída como a altura h_2 , tem-se a equação 3.11 descrita abaixo.

$$\frac{H_2(s)}{Q_p(s)} = \frac{R_2}{R_1 Ar_1 R_2 Ar_2 s^2 + (R_1 Ar_1 + R_2 Ar_2 + R_2 Ar_1) s + 1} \quad (3.11)$$

Para o fluxo turbulento, tem-se a presença de equações diferenciais não lineares. Estas equações serão apresentadas abaixo. Realizando o balanço de massa do sistema apresentado na figura 1, obteve-se as equações 3.12 e 3.13, que consideram um possível distúrbio atribuído a variável q_L .

$$Ar_1 \frac{dh_1}{dt} = q_p - q_{12} - q_L \quad (3.12)$$

$$Ar_2 \frac{dh_2}{dt} = q_{12} - q_2 \quad (3.13)$$

Pela equação 3.5, tem-se a relação entre a vazão e a altura para um fluxo turbulento. Para este sistema, tem-se a partir da equação 3.5 as equações 3.14 e 3.15, que representam a relação da altura e vazão em cada tanque.

$$q_{12} = k_{12} \sqrt{h_1 - h_2} \quad (3.14)$$

$$q_2 = k_2 \sqrt{h_2} \quad (3.15)$$

As equações descritas acima representam a vazão de saída do tanque 1, q_{12} , e a vazão de saída do tanque 2, q_2 . Aplicando as equações 3.14 e 3.15 nas equações 3.12 e 3.13, obtem-se o modelo matemático para este sistema, representado pelas equações 3.16 e 3.17.

$$Ar_1 \frac{dh_1}{dt} = q_p - k_L \sqrt{h_1} - k_{12} \sqrt{h_1 - h_2} \quad (3.16)$$

$$Ar_2 \frac{dh_2}{dt} = k_{12} \sqrt{h_1 - h_2} - k_2 \sqrt{h_2} \quad (3.17)$$

Os coeficientes k_{12} , k_2 e k_L serão obtidos experimentalmente, uma vez que seus valores variam conforme a abertura de cada válvula. Neste trabalho, seus valores serão considerados constantes.

3.4 Linearização do sistema

Para que o controle PID seja aplicado se faz necessário a linearização do sistema, uma vez que este controlador não considera as não-linearidades de um sistema. Assim, considerando que o sistema não terá grandes variações na condição de operação é possível linearizar o sistema descrito nas equações 3.16 e 3.17. O modelo linearizado será baseado no controle do nível do tanque 2, h_2 , a partir da vazão de entrada no tanque 1, q_p . A equação 3.2 apresenta a relação entre a variação no nível do tanque e a variação na vazão. A equação 3.18 apresenta esta relação e a partir da equação 3.5 temos:

$$R = \frac{dh}{dq} \quad (3.18)$$

$$dq = \frac{K}{2\sqrt{h}} dh \quad (3.19)$$

Sendo $K = (\sqrt{h} / q)$ a equação 3.19 pode ser manipulada conforme a equação 3.20.

$$\frac{dh}{dq} = \frac{2\sqrt{h}\sqrt{h}}{q} = \frac{2 \cdot h}{q} \quad (3.20)$$

Relacionando as equações 3.18 e 3.20, obtem-se o valor da resistência R conforme a equação 3.21.

$$R = \frac{2h}{q} \quad (3.21)$$

Considerando a figura 1 em regime permanente, tem-se que a vazão de entrada q_p é igual a vazão de saída q_2 e a vazão entre os tanques q_{12} . A altura dos níveis nos tanques são iguais. Logo, $q_p = q_2 = q_{12} = \bar{q}$ e $h_1 = h_2 = \bar{h}$. Assumindo que há uma mudança na vazão de entrada de \bar{q} para $\bar{q} + q_p$, onde q_p é pequena, as mudanças nos níveis dos tanques e nas vazões q_2 e q_{12}

serão pequenas (GOSMANN, 2002). As equações abaixo apresentam a linearização do modelo para variações pequenas. Para o tanque 1 tem-se as equações 3.22 a 3.24.

$$A_{r1} \frac{dh_1}{dt} = q_p - q_{12} - q_L \quad (3.22)$$

$$\frac{h_1 - h_2}{R_1} = q_{12} \quad (3.23)$$

Relacionando as equações 3.22 e 3.23 teremos:

$$h_1 = -R_1 A_{r1} \frac{dh_1}{dt} - q_L R_1 + q_p R_1 + h_2 \quad (3.24)$$

Para o tanque 2 tem-se as equações 3.25 à 3.27.

$$A_{r2} \frac{dh_2}{dt} = q_{12} - q_2 \quad (3.25)$$

$$\frac{h_2}{R_2} = q_2 \quad (3.26)$$

Relacionando as equações 3.23, 3.25 e 3.26 tem-se:

$$h_1 = R_1 A_{r2} \frac{dh_2}{dt} + h_2 + h_2 \left(\frac{R_1}{R_2} \right) \quad (3.27)$$

Considerando que $A_{r1} = A_{r2} = A_r$, e eliminando h_2 das equações 3.24 e 3.27, obtém-se a equação 3.28, que é a equação do modelo linearizado do sistema com dois tanques acoplados, considerando a variável de entrada como a vazão q_p e a variável de saída como a altura h_2 . Para este primeiro caso considerou-se que no sistema não há distúrbio, ou seja, q_L é nulo.

$$R_2 R_1 A^2 \frac{d^2 h_2}{dt^2} + (2A R_2 + R_1 A) \frac{dh_2}{dt} + h_2 = q_p R_2 \quad (3.28)$$

Para o sistema de dois tanques acoplados em termos de pequenos sinais tem-se a equação 3.29 descrita abaixo, que é a função de transferência deste processo desconsiderando qualquer distúrbio, através da aplicação da transformada de Laplace no modelo da equação 3.28.

$$\frac{H_2(s)}{Q_p(s)} = \frac{R_2}{R_1 R_2 A^2 s^2 + (2AR_2 + R_1 A)s + 1} \quad (3.29)$$

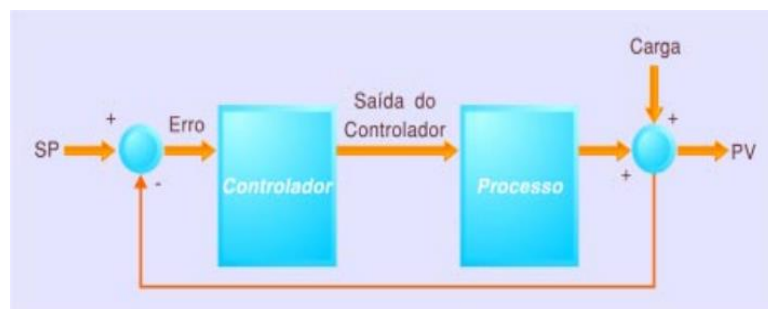
3.5 Sistemas de Controle

Existem basicamente dois tipos de natureza de controle: os auto-operados e os operados por alguma energia externa. Entre os auto-operados podemos citar o mais conhecido entre eles, o controle de nível por bóia, esse que existe em qualquer caixa d' água de nossas residências. Já os controladores baseados em energia externa podem ser dos tipos: controlador pneumático, controlador hidráulico, controlador elétrico ou eletrônico. De forma geral, temos que uma grandeza que precisa ser controlada (temperatura, nível, pressão, etc), e para mantermos essa grandeza sob controle precisamos de algumas informações (MATIAS, 2002):

- Valor desejado – Set-Point (SP).
- Valor real ou valor do processo (PV).
- Algoritmo de controle.

Com base nessas informações, o controlador compara o valor desejado (SP) com o valor do processo (PV) e determina, com base no algoritmo de controle, o valor de correção na saída do controlador para que o valor do processo (PV) se aproxime do valor desejado (SP), conforme ilustra a figura 2.

Figura 2 - Comparação entre o valor desejado e o valor de processo



Fonte: MATIAS, 2002.

Existem alguns algoritmos de controle que podem operar individualmente ou trabalhar em conjunto, conforme a precisão esperada do controle e também conforme o processo (MATIAS, 2002):

- Controle com ação proporcional (P).
- Controle com ação integral (I).
- Controle com ação derivativa (D).

O termo sistema é usado quando queremos descrever uma série de componentes que integram um determinado circuito e agem em conjunto para atingir um determinado objetivo. Um sistema é dito controlável quando a saída é controlada para obter um valor particular ou corrigir uma entrada (SERAPHIM, 2008). Assim, em um sistema tem-se uma série de componentes que podem ser descritos como:

- Variável controlada – condição que é medida e controlada, normalmente na saída do sistema.
- Variável manipulada – condição modificada pelo controlador, alterando o valor da variável controlada.
- Planta – conjunto dos elementos do sistema e que promove a interação entre eles.
- Distúrbio – sinal que tende a causar alterações indesejadas no valor da variável de saída.

3.5.1 Controle proporcional

Este controle é definido como a ação de controle mais simples em sistema. Neste controle os sinais de saída $U(s)$ e de entrada $E(s)$ são relacionados entre si por meio de um ganho proporcional K_p . A equação 3.30 apresenta a relação entre o sinal de entrada $E(s)$ e o sinal de saída $U(s)$.

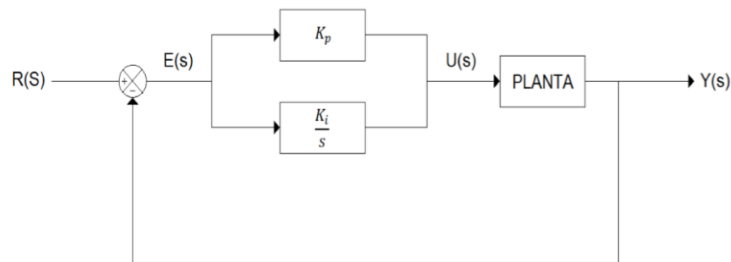
$$\frac{U(s)}{E(s)} = K_p \quad (3.30)$$

As vantagens da aplicação da ação proporcional no controle são a sua simplicidade, possibilitando a resposta rápida do sistema conforme o aumento do ganho K_p . Porém, se este ganho for muito elevado o sistema pode se tornar instável e sinais de ruído podem ser amplificados. Em se tratando de erro em regime, o controle proporcional pode apresentar uma resposta com um erro considerável em regime permanente, pelo fato de tal controlador não rejeitar distúrbios em regime estacionário (OGATA, 2003).

3.5.2 Controle proporcional-integral

A figura 3 apresenta a estrutura de controle proporcional e integral (PI) e a equação 3.31 mostra a função de transferência do controlador PI.

Figura 3 - Estrutura de controle proporcional e integral



Fonte: SILVA, 2014.

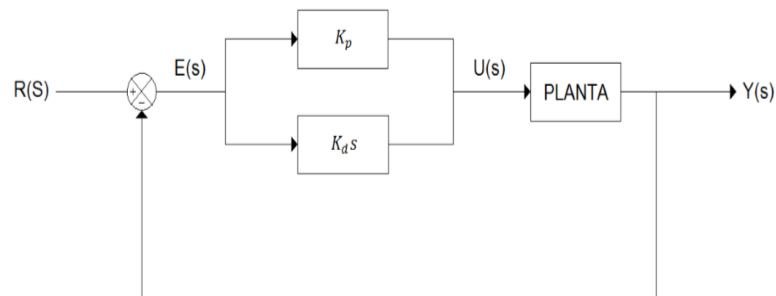
$$Gc(s) = k_p + \frac{k_i}{s} \tag{3.31}$$

Esse controlador aumenta o tipo do sistema em uma unidade e é utilizado para melhorar a resposta em regime estacionário. Tendo um pólo na origem e um zero em $s = -K_i/K_p$. Como o pólo está mais próximo da origem ele adiciona um ângulo negativo no lugar das raízes.

3.5.3 Controle proporcional-derivativo

A figura 4 apresenta a estrutura do controlador proporcional derivativo (PD) e a equação 3.32 mostra a função de transferência do controlador PD.

Figura 4 - Estrutura de controle proporcional derivativo



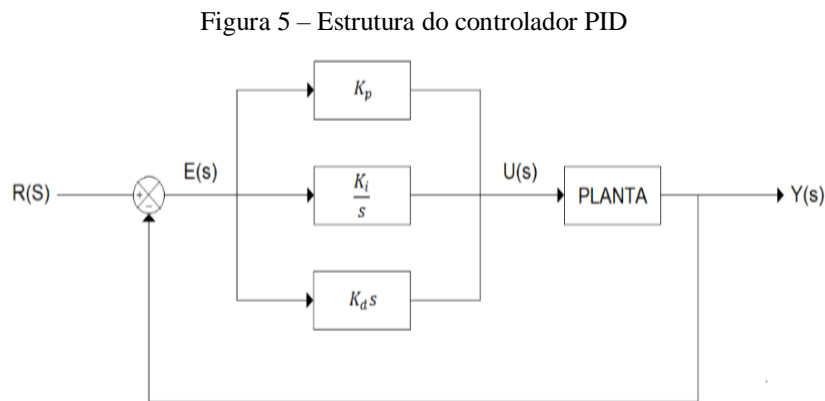
Fonte: SILVA, 2014.

$$Gc(s) = k_p + k_d s \tag{3.32}$$

este controlador tem um único zero em $s = -K_p/K_d$, adicionando um ângulo positivo do lugar das raízes. O controlador PD é um tipo de controlador de avanço de fase e melhora a resposta transitória do sistema.

3.5.4 Controle proporcional-integral-derivativo

A união das três ações de controle citadas acima é a ação de controle proporcional e integral derivativa (PID). Esta três ações (proporcional, integral e derivativa) são aplicadas ao sinal de erro $e(t)$, promovendo um sinal de saída $u(t)$ próximo ao valor de *set point*. A figura 5 apresenta a estrutura do controlador PID.



Fonte: SILVA, 2014.

A combinação destas três ações de controle promove uma redução das oscilações, redução do erro em regime e uma ação antecipada evitando que o erro se torne maior, portanto ele melhora tanto a resposta transitória quanto a resposta em regime estacionário. Este controlador utiliza as características particulares de cada ação de controle, obtendo uma melhora significativa do comportamento do sistema.

Os controladores PID, em geral são aplicáveis na maioria dos sistemas de controle, sendo utilizados em grande parte dos controladores industriais (SERAPHIM, 2008). A equação 3.33 apresenta a função de transferência do controlador PID.

$$G(s) = k_p + \frac{K_i}{s} + K_d s \quad (3.33)$$

3.6 Malha aberta e fechada de sistemas de controle

3.6.1 Sistema de controle em malha aberta

Em um sistema onde o sinal de saída não influencia no sinal de entrada, ou seja, não há realimentação (interação entre a saída e entrada), o sistema de controle é em malha aberta. Este tipo de sistema em malha aberta é relativamente simples e possui um custo relativamente baixo. A figura 6 apresenta um controlador em malha aberta.

Figura 6 - Controlador em malha aberta



Fonte: autoria própria.

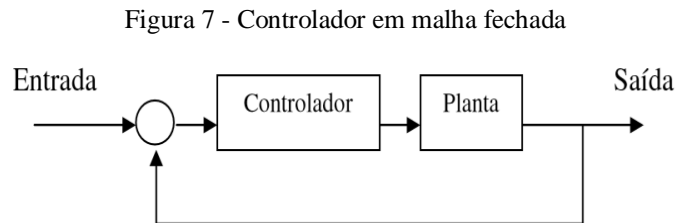
Como exemplo de um sistema em malha aberta, podemos utilizar a velocidade de rotação do eixo de um motor, pode ser determinada somente por um ajuste inicial da tensão aplicada na entrada do motor, Qualquer mudança na fonte de tensão ou nas características do motor, ocasionada por uma mudança na temperatura ou por uma variação na carga no eixo, provocará uma alteração na velocidade do eixo que não será compensada (SERAPHIM, 2008).

Algumas considerações para o sistema em malha aberta: o controle em sistema em malha aberta é impreciso, pois não há a possibilidade de efetuar correções automáticas a partir do sinal de saída. É necessário um conhecimento muito extenso da planta e das variáveis para que se possa obter um valor de saída desejado a partir do sinal de entrada. Portanto, o sistema de controle em malha aberta é mais utilizável em sistema onde são conhecidas as entradas e não há presença de perturbações.

3.6.2 Sistema de controle em malha fechada

Diferentemente do sistema de controle em malha aberta, o sistema em malha fechada é caracterizado pela existência de uma realimentação. Assim, é possível realizar correções no erro entre o valor desejado e o valor (SP) e o valor do processo (PV). Portanto, o sinal de erro é a diferença entre os sinais de entrada e o de realimentação, que normalmente é o sinal de saída

do sistema. Este sinal realimenta o controlador, que efetuará correções na entrada. A figura 7 apresenta um controlador em malha fechada.



Fonte: autoria própria.

Na maioria dos sistemas de controle existem variáveis simples que devem ser controladas, por exemplo, o nível da água de um reservatório, a velocidade de rotação de um motor ou até mesmo a posição de uma mão. Mas existem sistemas de controle onde mais de uma variável deve ser controladas (SERAPHIM, 2008).

Um dos problemas de sistema de malha fechada é o atraso de transporte, causado pela necessidade de igualar os valores reais com os desejados. Este atraso pode comprometer a estabilidade do sistema, ocasionar oscilações e dificultar a ação corretiva. A aplicação mais prática do sistema de controle em malha fechada é quando o sistema está sujeito a perturbações e variações que não podem ser previstas. Para que haja um controle satisfatório no processo em malha fechada é necessário uma análise minuciosa da planta e os controladores devem ser projetados com bastante cuidado para não ocasionarem resultados indesejáveis.

Em um sistema em malha fechada a uma entrada do tipo degrau, podemos adquirir especificações da dinâmica do sistema baseadas em alguns critérios a partir de parâmetros de interesse. Estes parâmetros são definidos, na maioria ds vezes, de acordo com a resposta padrão de sistemas de primeira e segunda ordem (GOSMANN, 2002):

- Tempo de subida (t_r) - é o tempo necessário para a resposta do sistema percorrer o intervalo entre 10% e 90% do seu valor final;
- Tempo de acomodação (t_s) - é o tempo necessário para que a resposta do sistema alcance e permaneça em uma faixa de 2% do seu valor final;

- Sobre-sinal máximo (M_p) - é o valor máximo da resposta do sistema com relação ao seu valor final em termos percentuais.

3.7 Resposta de sistemas de primeira e segunda ordem

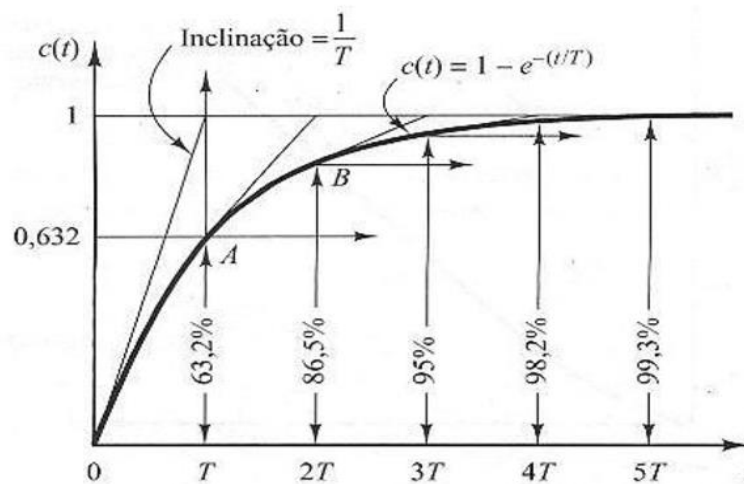
3.7.1 Sistema de primeira ordem

A função de transferência de um sistema de primeira ordem é descrito pela equação 3.34, onde K é o ganho do sinal, τ é a constante de tempo e L é o tempo de atraso. O termo exponencial representa o atraso de transporte.

$$\frac{c(s)}{U(s)} = \frac{K e^{-Ls}}{(\tau s + 1)} \quad (3.34)$$

A curva típica de um sistema de primeira ordem para uma entrada degrau assume em $\tau = t$ 63,2% da resposta em regime. Quanto menor a constante de tempo τ mais rápido o sistema atinge o valor degrau aplicado e sua entrada. A resposta em malha aberta de sistema de primeira ordem a uma entrada degrau $u(s) = 1/s$ é apresentada na figura 8.

Figura 8 - Resposta de sistema de primeira ordem a uma entrada degrau



Fonte: OGATA, 2003.

Para o sistema de primeira ordem o tempo de acomodação $t_s \geq 4T$ (ou 4τ). O tempo de subida t_r é igual a $2T$ (ou 2τ) e não existe sobre-sinal ($M_p = 0$).

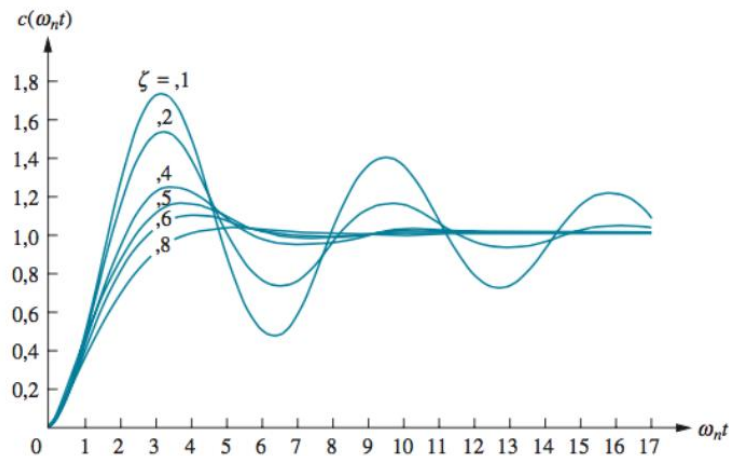
3.7.2 Sistema de segunda ordem

A função de transferência de um sistema de segunda ordem é descrito pela equação 3.35, onde ζ é o fator de amortecimento e w_n é a frequência natural não-amortecida.

$$\frac{C(s)}{U(s)} = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} \quad (3.35)$$

A resposta em malha aberta de sistema de primeira ordem a uma entrada degrau $u(s) = 1/s$ é apresentada na figura 9.

Figura 9 - Resposta de sistema de segunda ordem a uma entrada degrau



Fonte: autoria própria

Para o sistema de segunda ordem o tempo de acomodação t_s é $\geq 4\tau$ (como no caso do sistema de primeira ordem), o tempo de subida é igual a $1.8/w_n$, o sobre-sinal máximo para o caso em que $\zeta \geq 1$ é igual a 0 ($M_p = 0$) e para o caso em que $0 \leq \zeta < 1$ é dado pela equação 3.36.

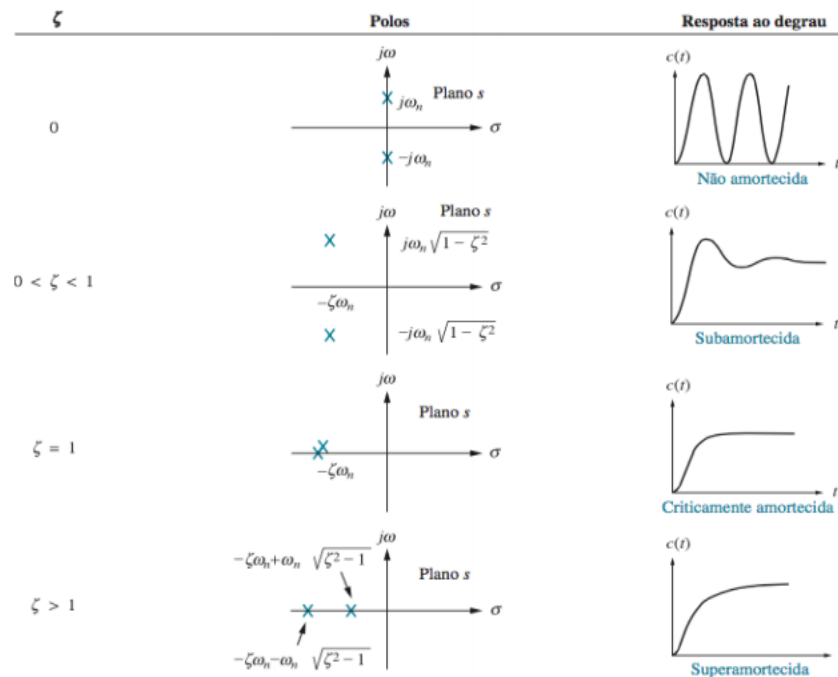
$$M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}} \quad (3.36)$$

Os sistemas de segunda ordem tomam as seguintes designações em função do valor de ζ :

- Sistema sub-amortecido, se $0 < \zeta < 1$.
- Sistema criticamente amortecido, se $\zeta = 1$.
- Sistema sobre-amortecido, se $\zeta > 1$.

A figura 10 apresenta as condições e a resposta de um sistema de segunda ordem a uma entrada degrau para os valores admissíveis do fator de amortecimento.

Figura 10 - Resposta de segunda ordem em função do fator de amortecimento



Fonte: OGATA, 2003

3.8 PID digital

Por se tratar de um controlador muito usado nos processos industriais e por sua relevância no ensino e pesquisa de sistemas de controle, o controlador PID será implementado neste trabalho. Porém existem algumas dificuldades em realizar a discretização de funções de tempo contínuo que possuem partes integrativas e derivativas. Para a solução deste problema, assume-se que o controlador PID tem uma forma de equação diferencial, e aproximando os termos P(s), I(s) e D(s) para equações discretas, tem-se um controlador PID que pode ser implementado em *hardware* digital.

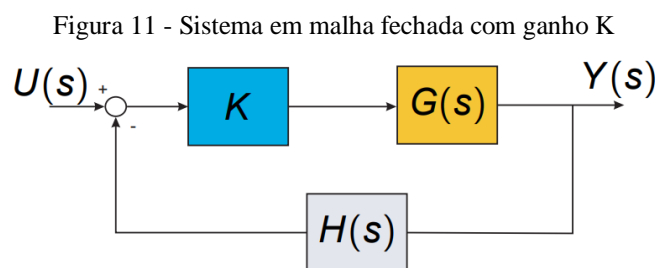
Uma maneira de definir a equação discreta de um controlador PID é aplicando métodos de aproximação, como o métodos trapezoidal e *backward*. O método trapazoidal é aplicável na parte integrativa, pois estabelece aproximação de integrais. Já o método *backward* é aplicável na parte derivativa, pois é um método que não tem a tendência de levar o sistema a instabilidade.

Para uma análise mais detalhada e mais especificações dos métodos citados acima, consulte SILVA, 2014. A equação 3.37 apresenta a forma discreta do controlador PID implementado no algoritmo de controle na plataforma de *script* do MATLAB®.

$$u[n] = u[n - 1] + K_p[e[n] - e[n - 1]] + \frac{K_i T_s}{2}[e[n] + e[n - 1]] + \frac{K_d}{T_s}[e[n] - 2e[n - 1] + e[n - 2]] \quad (3.37)$$

3.9 Lugar das raízes

Para realizar a sintonia do controlador PID, optou-se pelo método do lugar das raízes, por se tratar de um método muito estudado no processo de aprendizagem do aluno no curso de graduação em engenharia elétrica da UFES. Este método analisa a localização dos pólos de um sistema em malha fechada, variando assim apenas um parâmetro (em geral, o ganho), que podemos definir como K. A figura 11 apresenta um modelo de um sistema em malha fechada com o ganho K. Desta maneira, é possível aplicar pólos e zeros no sistema em malha fechada, de acordo com a representação de cada controlador (P, PI, PD e PID) e observar a resposta do sistema.



Fonte: ANDREA, 2011.

Para a visualização dos pólos e zeros, utiliza-se o lugar geométrico das raízes, que é um gráfico construído a partir do conhecimento dos pólos e zeros do sistema em malha aberta. Portanto, o LGR é o conjunto dos pontos no plano complexo que correspondem aos polos do sistema em malha aberta.

A equação 3.38 apresenta a função de transferência em malha fechada de um sistema genérico.

$$\frac{C(s)}{U(s)} = \frac{G(s)}{1+G(s)*H(s)} \quad (3.38)$$

Os pólos do sistema em malha fechada que determinam as características da resposta do sistema são as raízes do denominador da equação 3.38. A partir dessas raízes, juntamente com os critérios de módulo e ângulo, pode-se definir a região no plano complexo dos pólos do sistema. Para encontrar os zeros do sistema, determina-se os valores para os quais a função de transferência apresentada na equação 3.38 é nula, à saber, $G(s) = 0$.

De forma geral, podemos utilizar o método do lugar das raízes para analisar em quais situações de parâmetros K_p , K_i e K_d o sistema não se torna instável, com um valor de sobre-sinal dentro da faixa desejada e com um tempo de amortecimento relativamente pequeno.

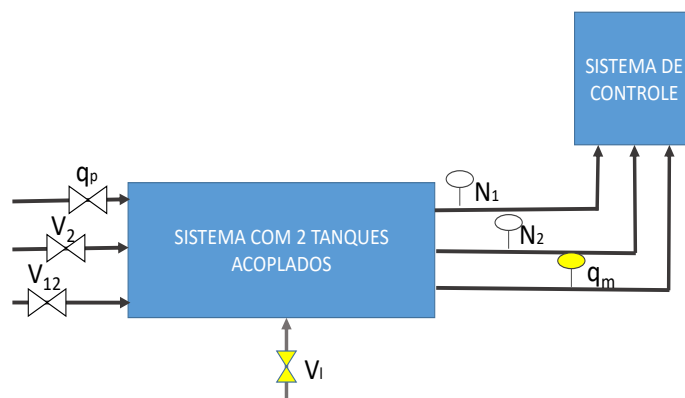
4 ESTRUTURA DO SISTEMA

Na indústria, uma das aplicações de sistemas digitais está ligada a processos que dependem do controle de nível de líquidos armazenados em tanques (SILVA, 2014). Para que se tenha um controle satisfatório, é necessário um estudo sobre a planta a qual se deseja controlar e o conhecimento sobre a dinâmica do processo se torna indispensável para a melhor escolha dos sensores e atuadores. Assim sendo, deve-se definir a malha de controle e dimensionar corretamente os tanques do sistema.

4.1 Malha de controle

Neste sistema ter-se-ão componentes de medição e atuadores que fornecerão dados para o controle do processo. A figura 12 apresenta o diagrama de blocos deste sistema com a representação dos instrumentos atuadores e de medição que são necessários para este sistema.

Figura 12 - Diagrama de bloco da malha de controle



Fonte: autoria própria.

Utilizou-se como instrumentos atuadores 1 (uma) bomba de água q_p para fornecer a vazão de entrada no primeiro reservatório, 2 (duas) válvulas manuais V_{12} e V_2 para definir a vazão de saída de cada tanque e 1 (uma) válvula manual V_L que representa um possível distúrbio com a liberação de líquido do tanque 1.

Como instrumentos de medição, utilizou-se 2 (dois) sensores de nível N_1 e N_2 para medir a altura do líquido em cada tanque e 1 (um) sensor de vazão q_m para medir a vazão de saída do tanque 2.

Na figura 12 observa-se a presença do instrumento atuador (vazão de entrada q_p) e dos instrumentos de medição (sensor de nível do tanque 1 N_1 , sensor de nível do tanque 2 N_2 e sensor de vazão q_m). As válvulas V_{12} , V_2 e V_L são válvulas manuais e são manipuladas para representar variações no processo de operação da planta. Os sensores enviam sinais para o sistema de controle, que por sua vez aciona a bomba que aplica a entrada de líquido no sistema de tanques acoplados.

4.2 Dimensionamento dos tanques

Neste trabalho é considerado que os dois tanques acoplados possuem a mesma capacidade de armazenamento e foram fabricados em material de acrílico para possibilitar a visualização do nível de líquido em tempo real e por ser um material de baixo custo.

Para que a dinâmica do processo de controle seja rápida é necessário que o volume dos tanques obedeça uma faixa de valores. Neste trabalho, admitiu-se que a variação de nível nos tanques fosse de 0,6 a 0,7cm/s para que o tempo de resposta do controlador seja satisfatório, uma vez que utilizaremos válvulas manuais de 1/2" (meia polegada).

As bombas de valor mais acessível no mercado tem a capacidade de vazão na ordem de 80 e 120L/h, para bombas de 3 a 6V, e de 240L/h, para bombas de 12V. Por se tratar de um sistema didático, os tanques possuem uma altura máxima para o nível do líquido na ordem de 5 a 15cm. A partir da equação 4.1, que representa o volume de um tanque em formato paralelepípedo, necessita-se definir a área da base (A_r) dos tanques obedecendo a faixa de variação do nível do líquido e a faixa de vazão das bombas.

$$V_{\text{paralelepípedo}} = (\text{área da base}) * h \quad (4.1)$$

A tabela 1 apresenta alguns valores para a área da base dos reservatórios e a variação do nível do líquido para cada valor de vazão das bombas presentes no mercado com valor mais acessível.

Tabela 1 - Variação do nível do líquido em função da área da base do tanque

Altura (cm)	Área da base (cm ²)	Volume (cm ³)	Variação do nível (cm/s)			Tempo para encher (s)		
			80L/h	120L/h	240L/h	80L/h	120L/h	240L/h
05	51,84	0,259*10 ³	0,4287	0,643	1,286	11,66	7,77	3,88
	225	1,125*10 ³	0,0988	0,148	0,296	50,62	33,75	16,87
10	51,84	0,518*10 ³	0,4287	0,643	1,286	23,32	15,55	7,77
	225	2,250*10 ³	0,0988	0,148	0,296	101,25	67,50	33,75
15	51,84	0,777*10 ³	0,4287	0,643	1,286	34,99	23,33	11,66
	225	3,375*10 ³	0,0988	0,148	0,296	151,87	101,25	50,62

Fonte: autoria própria.

Conforme a altura máxima do tanque aumenta, tem-se que o tempo para encher cada tanque aumenta, tornando o processo mais lento. A variação do nível é dependente apenas da área da base do tanque, e cresce conforme a vazão da bomba aumenta. Para que se tenha uma variação de nível em torno de 0,6 e 0,7cm/s, utiliza-se uma bomba de 120L/h (2L/min) e cada tanque terá a área de base (A_T) igual a 51,84cm². Para uma altura máxima de líquido de 10cm, tem-se que o tempo para encher o tanque é de aproximadamente 16s e para uma altura máxima de 15cm, o tempo para encher o tanque é próximo de 23s.

Para que haja a escolha correta das dimensões dos tanques, foi necessário realizar alguns testes e simulações para validação dos valores de resistência das válvulas V_{12} e V_2 , a variação do nível de líquido nos tanques e o tempo para encher e esvaziar cada tanque.

4.3 Testes iniciais

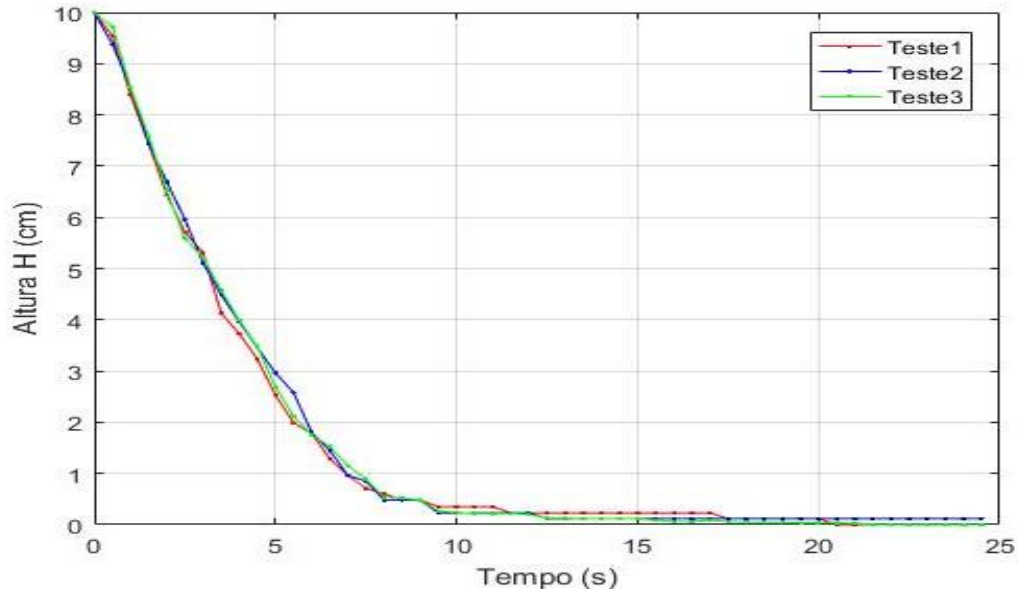
Inicialmente realizou-se os testes com apenas um tanque e definiu-se o valor da resistência da válvula de acoplamento dos dois tanques (V_{12}). Posteriormente, realizou-se as simulações com os dados mais relevantes da tabela 1 e analisou-se os resultados obtidos.

4.3.1 Resistência da válvula V_{12}

Para a realização dos testes de resistência da válvula V_{12} utilizou-se uma válvula manual em aço com registro esfera em borboleta de 1/2" (meia polegada). No experimento foi adotado a

abertura máxima da válvula representando o seu menor valor de resistência. A figura 13 apresenta a curva característica dos testes de resistência.

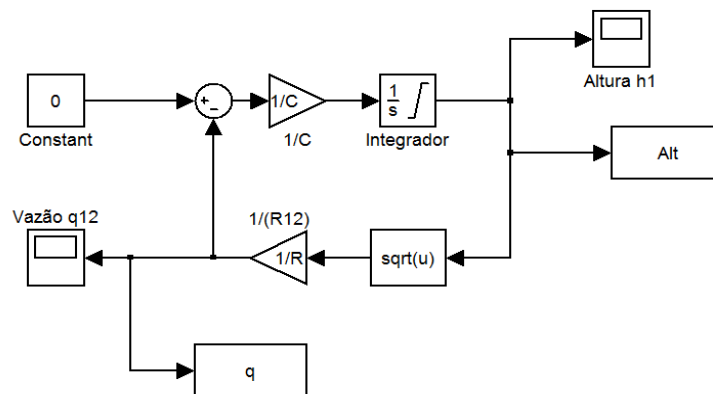
Figura 13 - Teste de resistência da válvula V12



Fonte: autoria própria.

Para validar estes testes foram realizadas simulações na plataforma *SIMULINKTM* afim de encontrar o valor da resistência da válvula V_{12} via diagrama de blocos (MARGON, 2010). A figura 14 apresenta o diagrama de blocos utilizado para a realização da simulação. Foi definido como condição inicial o valor de 10cm no bloco do integrador, admitindo que o tanque estava com o nível máximo. Foram utilizados alguns valores para o valor de R_{12} (resistência da válvula V_{12}) de modo a obter a mesma curva de nível do líquido x tempo da figura 13.

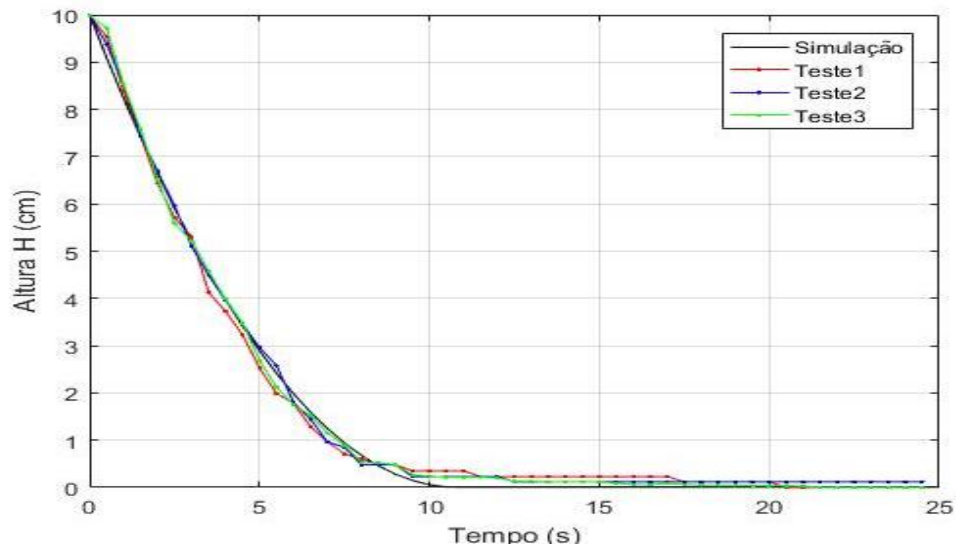
Figura 14 - Diagrama de blocos do sistema para teste de resistência da válvula V_{12}



Fonte: MARGON, 2010.

A figura 15 apresenta a curva característica do nível de líquido x tempo obtida via simulação na plataforma *SIMULINKTM* e as curvas obtidas via testes experimentais.

Figura 15 - Comparação da curva de nível do líquido x tempo obtida via simulação e via testes experimentais



Fonte: autoria própria.

A partir da comparação feita entre a curva de nível do líquido x tempo obtida via simulação e via testes experimentais pode-se determinar o valor da resistência R_{12} . Para a curva obtida por meio de simulação da figura 15 utilizou-se o valor de $R_{12} = 3500$. Desta forma, para um tanque com área da base igual a $51,84\text{cm}^2$ e altura do líquido de 10cm , tem-se que a resistência da válvula de $1/2''$ (meia polegada) é igual a 3500 . Utilizando este valor para V_{12} , simulou-se o tempo para encher e esvaziar o tanque 1 e obteve-se a curva de vazão q_{12} no tempo.

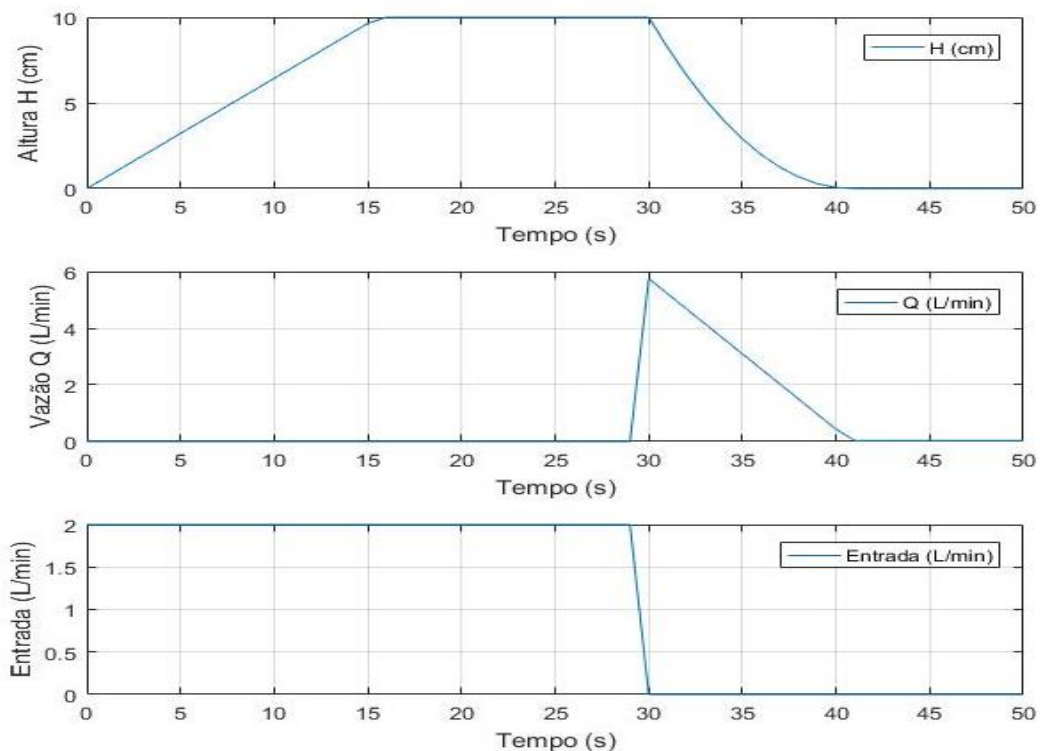
4.3.2 Simulação sistema monotanque

O valor da resistência da válvula V_{12} depende do coeficiente de descarga do escoamento, da área de seção transversal do tanque, da área do orifício de saída, da altura do tanque e da aceleração da gravidade (GOSMANN, 2002). Desta maneira, foi feita a simulação do sistema com as dimensões admitidas nos testes experimentais, à saber, área da base do tanque 1 igual a $51,84\text{cm}^2$, altura máxima do líquido igual a 10cm e R_{12} igual a 3500 .

A figura 16 apresenta o comportamento do sistema para as dimensões citadas com uma vazão de entrada de $2\text{L}/\text{min}$ e resistência da válvula V_{12} igual a 3500 . O sinal de entrada tem duração

de 30s, representando a vazão de entrada q_p durante este intervalo de tempo. A partir do instante 30s a válvula V_{12} é aberta e tem-se a dinâmica de escoamento do líquido através da válvula e a variação do nível no tempo.

Figura 16 – Dinâmica do sistema monotanque para R_{12} igual a 3500

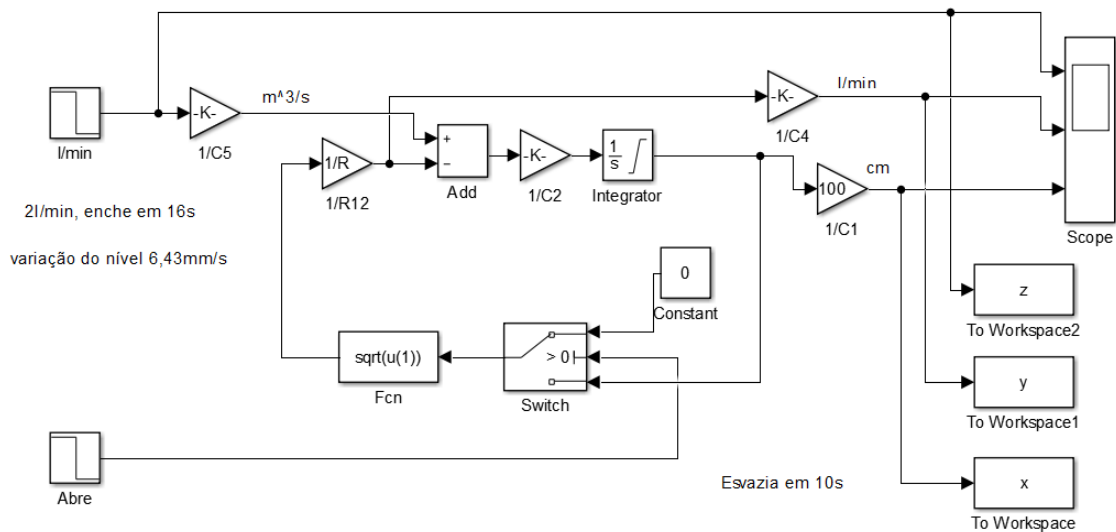


Fonte: autoria própria.

Para validar os dados da tabela 1 foi necessário analisar cada gráfico em particular. A figura 16 mostra que o tempo para o preenchimento do nível do tanque 1 (altura máxima de 10cm) é de aproximadamente 16s, que é um valor muito próximo do valor calculado para um reservatório com $51,84\text{cm}^2$ área de base e 10cm de altura máxima, conforme tabela 1.

O tempo para que o tanque 1 esteja totalmente vazio é de aproximadamente 10s. Este tempo para o esvaziamento é menor do que o tempo para encher o tanque 1, uma vez que a vazão q_{12} está com um valor de pico (5.75 L/min), que é um valor mais elevado do que a vazão de entrada q_p (2L/min). O comportamento da vazão q_{12} , para o valor de R_{12} igual a 3500, possui um valor de pico de vazão elevado, dando uma dinâmica rápida para o escoamento do líquido através da válvula V_{12} . A figura 17 apresenta o diagrama de blocos para o sistema monotanque.

Figura 17 - diagrama de blocos do sistema monotanque



Fonte: autoria própria.

4.3.3 Simulação sistema de dois tanques acoplados

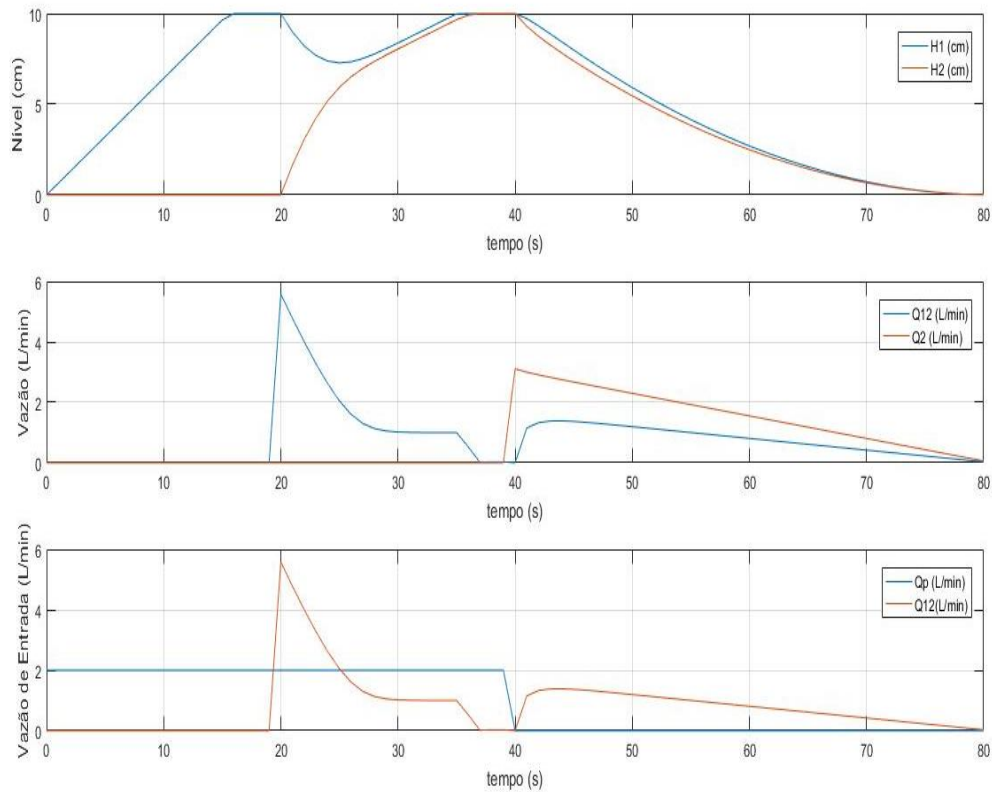
Considerando a figura 1 e os resultados apresentados nas seções anteriores pode-se obter um panorama do comportamento do sistema com dois tanques acoplados. A figura 18 apresenta a dinâmica do processo a partir de uma entrada degrau (2L/min) durante 40s. Admitindo que os dois tanques possuem as mesmas dimensões, tem-se que o nível máximo em cada tanque é igual a 10cm. Realizando os mesmos teste para a válvula V_2 , obteve-se os valores de resistência R das duas válvulas, a saber, $R_{12} = 3500$ e $R_2 = 6050$.

A simulação apresentada na figura 18 mostra a dinâmica do processo durante um período de 70s. No instante 20s a válvula V_{12} é aberta, permitindo que o líquido escoe do tanque 1 para o tanque 2. A válvula V_2 é mantida fechada até o instante 40s, após este período ela é aberta, permitindo a passagem do líquido. A entrada do sistema é uma vazão $Q_p = 2L/min$ que tem duração de 40s.

Nota-se que o nível do tanque 1 alcança a sua altura máxima aproximadamente no instante 16s e tem uma variação a partir do instante da abertura da válvula V_{12} , retornando para a altura máxima de 10cm após 15s. Esta variação ocorre devido ao fato da vazão q_{12} possuir um valor instantaneo maior que a vazão q_p , proporcionando assim esta dinâmica para o processo.

No instante 40s a válvula V_2 é aberta, e a dinâmica de escoamento dos tanques possui uma pequena diferença entre si devido ao atraso de transporte do sistema.

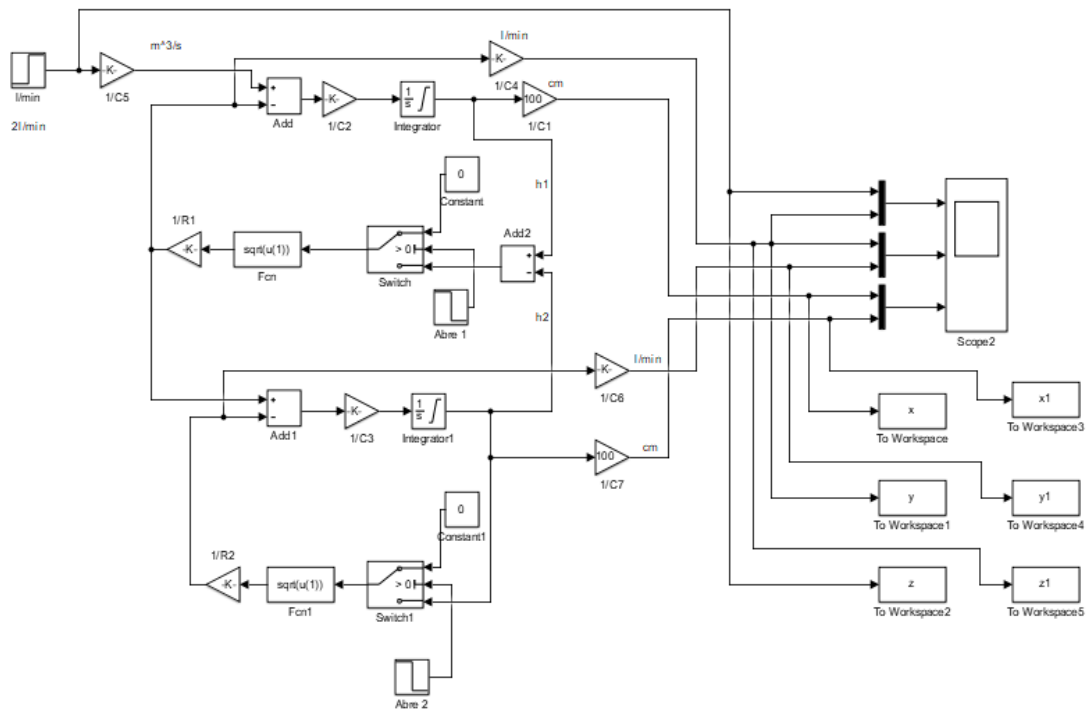
Figura 18 – Dinâmica do sistema de dois tanques acoplados para $Q_p = 2\text{L/min}$ durante 40s



Fonte: autoria própria.

A vazão q_{12} tem um valor elevado no instante em que a válvula V_{12} é aberta. Este valor decresce até o instante em que a altura do nível do tanque 1 se aproxima da altura do nível do tanque 2. A partir deste instante o valor da vazão q_{12} permanece constante até que os níveis dos tanques alcancem o valor máximo. O tempo para que os dois tanques estejam completamente vazios é de aproximadamente 35s. Este valor representa um tempo satisfatório para que ocorra um controle do processo. A figura 19 apresenta o diagrama de blocos para o sistema com dois tanques acoplados.

Figura 19 - diagrama de blocos do sistema de dois tanques acoplados



Fonte: autoria própria.

4.4 Hardware do sistema de tanques acoplados

Conforme foi analisado o sistema de tanques acoplados desenvolvido neste trabalho possui uma dinâmica rápida, considerando as dimensões dos tanques e o propósito de se obter um sistema de fácil controlabilidade. As simulações apresentadas facilitam a escolha e a aplicação dos componentes eletrônicos que serão utilizados no protótipo. Apresenta-se os sensores e atuadores do sistema, a plataforma do microcontrolador ARDUINO UNO, circuitos de acionamentos e a comunicação entre o *hardware* e o *software* MATLAB®.

4.4.1 Sensor de pressão MPX5010DP

Os modelos de sensores com sufixo DP (*Differential Pressure*) medem a diferença de pressão entre as suas duas entradas. Deixando a entrada de vácuo aberta, o valor da pressão medida é a diferença para a pressão atmosférica (RYAN, 2013). Neste trabalho utilizou-se o modelo MPX5010DP, que tem uma resolução satisfatória de até 1 litro. A saída deste sensor é analógica, de 0V a Vcc, proporcional à pressão aplicada e sua alimentação é de 5V. A figura 20 apresenta o sensor de pressão e a identificação dos seus pinos.

Figura 20 - Sensor de pressão MPX5010DP

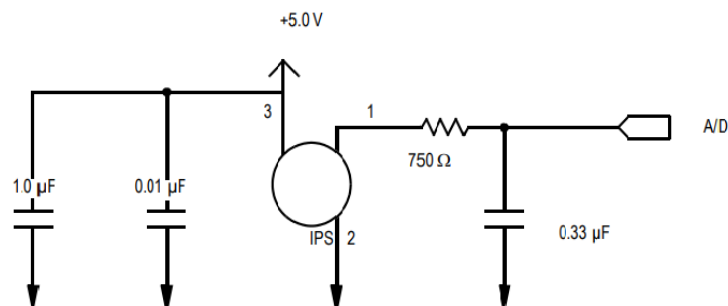


Fonte: autoria própria

Os Sensores de Pressão Integrados (IPS) possuem um aparador de saída, compensação de temperatura incorporada e amplificação do sinal de saída que os tornam compatíveis com conversores analógicos-digitais (A/D) em micro-controladores de baixo custo. Embora os conversores A/D de 8 bits sejam mais comuns, a maior resolução nesses conversores está se tornando cada vez mais disponível. Com estes sinais de maior resolução em conversores A/D, o ruído se torna inerente aos componentes resistivos (REODIQUE, 2005).

A figura 11 apresenta o circuito passa-baixa para diminuir o ruído da saída do sensor MPX5010DP e para fornecer um sinal mais refinado para o microcontrolador ARDUINO UNO.

Figura 21- Circuito filtro passa-baixa para saída do sensor MPX5010DP



Fonte: REODIQUE, 2005.

4.4.2 Sensor de vazão

O sensor de fluxo de água trabalha com uma turbina (no seu interior existe um tipo de hélice que ao receber o atrito de água gira) e junto a esta turbina existe um ímã fixado e paralelamente ao ímã existe um sensor de efeito hall que detecta cada volta dada por ele. Ou seja, quando a água passa pela hélice, são gerados pulsos proporcionais a velocidade da hélice. Para aplicação

prática é necessário a utilização do sensor de fluxo de água em conjunto com uma plataforma de prototipagem, o Arduino, por exemplo, utilizando as portas digitais (PORTA, 2016).

Conforme a equação 3.4, pode-se controlar a vazão de saída através da vazão de entrada no sistema. Observando esta possibilidade, utilizou-se um sensor de fluxo de água de 1/2" (meia polegada) na saída do tanque 2 para medir a vazão q_2 . A figura 22 apresenta o modelo do sensor de vazão de 1/2" utilizado neste trabalho.

Figura 22 - Sensor de vazão de 1/2" YF-S201



Fonte: autoria própria

Especificações técnicas do sensor de vazão 1/2":

- 3 fios: vermelho (5V), preto (GND), amarelo (sinal de saída).
- Material: nylon.
- Modelo: YF-S201.
- Rosca de 1/2".
- Sinal de saída: pulsos conforme a velocidade do rotor interno.
- Pulsos: nível alto 4,5V e nível baixo 0,5V.
- Tensão nominal: 5Vcc.
- Faixa de vazão: 1 a 30L/min.
- Pressão de operação máxima: 1,75Mpa.
- Dimensões: 58x37x37mm.

As simulações realizadas para o sistema de dois tanques acoplados, apresentaram uma vazão máxima de saída do tanque 2 (q_2) próximo a 6L/min. Admitindo uma vazão máxima de entrada igual a 2L/min, observamos que a faixa de vazão admissível pelo sensor de fluxo de líquido é confortável para obtermos uma leitura sem erro e perda de dados.

4.4.3 Bomba submersa 12V

Conforme os dados da tabela 1 e as simulações realizadas, foi definido uma bomba que forneça 120L/h (2L/min) para que o sistema possua uma dinâmica rápida. Para o sistema apresentado neste trabalho, as simulações demonstraram que uma bomba de 120L/h proporcionaria um período curto para que o nível nos tanques chegasse em seu limite máximo (aproximadamente 30s, considerando os dois tanques).

Os dados obtidos na tabela 1 admitem que para uma bomba de 120L/h o sistema apresentado teria uma variação de nível do líquido igual a 0.643cm/s, porém, pode-se afirmar que a bomba funcionaria em seu regime máximo de operação (corrente máxima), ou seja, o motor trabalharia perto da região de saturação. Como este trabalho tem a finalidade de desenvolver um sistema robusto, escolheu-se uma bomba de 240L/h como atuador do sistema, garantindo assim que o motor não estará saturado e operará em condições lineares de trabalho.

A figura 23 apresenta o modelo da bomba utilizada neste trabalho. Para que os resultados obtidos nas simulações sejam validados, trabalhou-se com a metade da vazão máxima fornecida por esta bomba ($Q_{max} = 4\text{L/min}$ ou 240L/h), aplicando um limitador de 50% para a tensão de saída do circuito de acionamento da bomba submersa.

Figura 23 - Bomba submersa 240L/h



Fonte: autoria própria

Especificações técnicas da bomba submersa:

- Modelo: AD20P-1230A.
- Tensão nominal: 12Vcc.
- Potência: 4,8W.

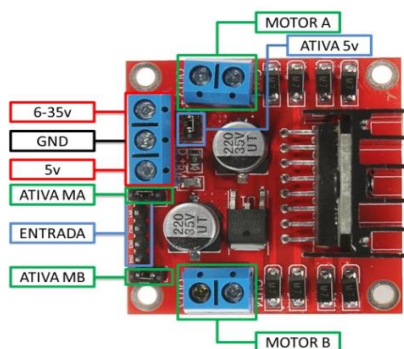
- Corrente nominal máxima: 400mA.
- Temperatura de trabalho: 0 ~ 75°.
- Vazão máxima: 4L/min (240L/h).
- Altura máxima de elevação: 3 metros.
- Ruído: menor que 40dB.
- IP68 (submersível).
- Dimensões: 5,5x3,5x4,5cm

4.4.4 CI L298N

O ARDUINO UNO fornece uma tensão de saída de 5Vcc e corrente máxima de 40mA (ARDUINO, 2019) e estes valores limitam a operação da bomba submersa. Devido à magnitude da corrente e da tensão solicitadas pela bomba em relação às correntes e tensões do circuito de controle, é necessário o uso de um circuito de acionamento capaz de alimentar a bomba e também isolar eletricamente circuitos onde circulam altas correntes de circuitos de comando (SILVA, 2014).

Foi definido como componente do circuito de acionamento o CI L298N, fabricado pela *ST Electronics*. Este circuito funcionou juntamente com uma fonte de alimentação de 12V. A figura 24 apresenta o modelo e a descrição de cada pino do módulo L298N. Este circuito integrado (CI) é composto por dois circuitos ponte H que são independentes. Circuitos em ponte H são circuitos compostos por transistores dispostos de maneira que possibilitem a inversão da rotação de motores de corrente contínua simplesmente invertendo o sentido da corrente que percorre os enrolamentos da máquina (SILVA, 2014).

Figura 24 - CI L298N com dois circuitos ponte H

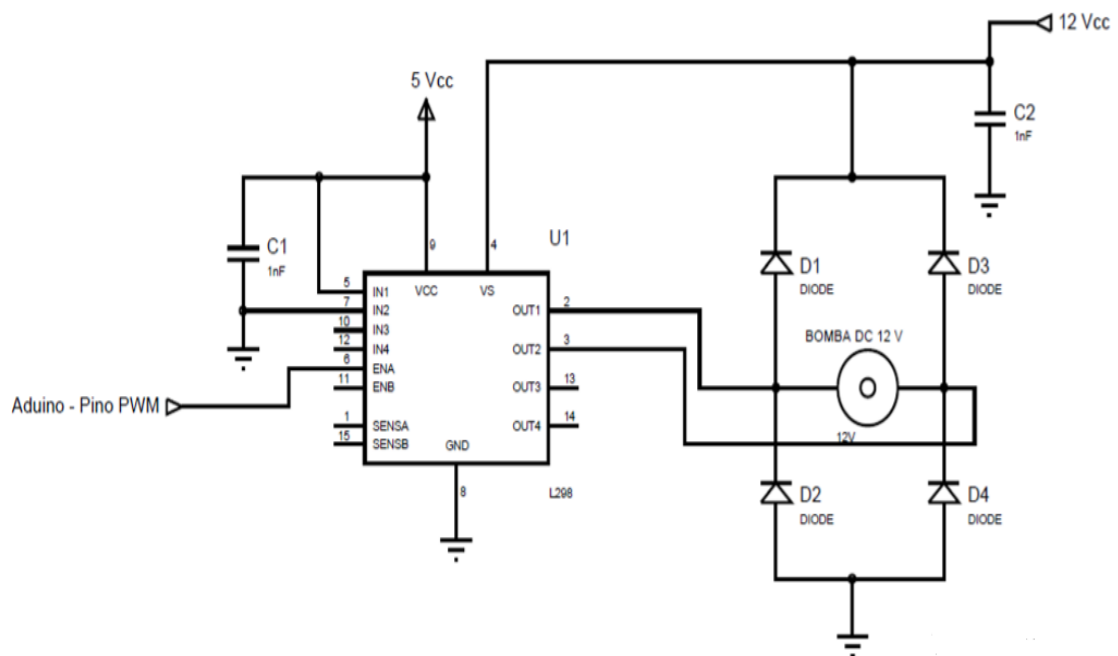


Especificações técnicas da Ponte H L298N:

- Tensão de Operação: 4~35 V.
- Chip: ST L298N.
- Controle de 2 motores CC ou 1 motor de passo.
- Corrente de Operação máxima: 2A por canal ou 4A max.
- Tensão lógica: 5 V.
- Corrente lógica: 0~36 mA.
- Limites de Temperatura: -20 a +135°C.
- Potência Máxima: 25W. • Dimensões: 43 x 43 x 27 mm.
- Peso: 30g.

É possível controlar a velocidade do motor a partir de um sinal PWM no pino de entrada do CI. A figura 25 apresenta o esquemático para a utilização de apenas um circuito ponte H do CI L298N. Este circuito foi utilizado neste trabalho para o acionamento da bomba submersa de 12Vcc a partir de um sinal PWM proveniente do ARDUINO UNO.

Figura 25 - Essquemático circuito de acionamento da bomba submersa



Fonte: SILVA, 2014.

4.4.5 Modulação PWM

O controle de velocidade dos motores CC é realizado por meio da tensão de alimentação aplicada em seus enrolamentos, ou seja, quanto maior a tensão aplicada, mais rápido o motor CC gira. Um método relativamente simples para realizar o controle de velocidade de um motor CC, é chamado de modulação por largura de pulso (PWM - Pulse Width Modulation) que mantém o torque da máquina constante além de ser um sinal muito menos sensível a ruídos, além de ser muito mais barato de outras formas de controle de tensão (ZHAN, 2007).

O ARDUINO possui somente saídas digitais, porém através da função *analogWrite* é possível gerar um sinal PWM, e este sinal varia o ciclo de trabalho conforme um valor de entrada através da comunicação serial ou por um valor que pode ser pré-definido. Na plataforma ARDUINO UNO o sinal PWM assume valores na faixa de 0 – 255 e tensões de 0 a 5V. Este sinal gerado é enviado para o CI L298N, que proporcionalmente controla o sinal de tensão aplicado ao motor da bomba submersa.

A modulação PWM consiste em gerar um sinal que possua amplitude constante, entretanto, com larguras de pulso que variam de 0 a 100% do período do sinal. A variação da largura de pulso é denominada ciclo de trabalho (*Duty Cycle - DC*), que corresponde à relação entre o tempo em que o sinal se encontra em nível lógico alto e o período, e está relacionada diretamente com a tensão entregue a carga. Neste caso: $DC = 0\%$ equivale a $V_L = 0V_{cc}$ e $DC = 100\%$ equivale a $V_L = 12 V_{cc}$, podendo DC e V_L assumir qualquer valor entre seus respectivos limites (SILVA, 2014). A equação 4.2 apresenta o cálculo do DC .

$$Duty\ Cycle = \frac{T_{na}}{T_{na} + T_{nb}} = \frac{T_{na}}{T} \quad (4.2)$$

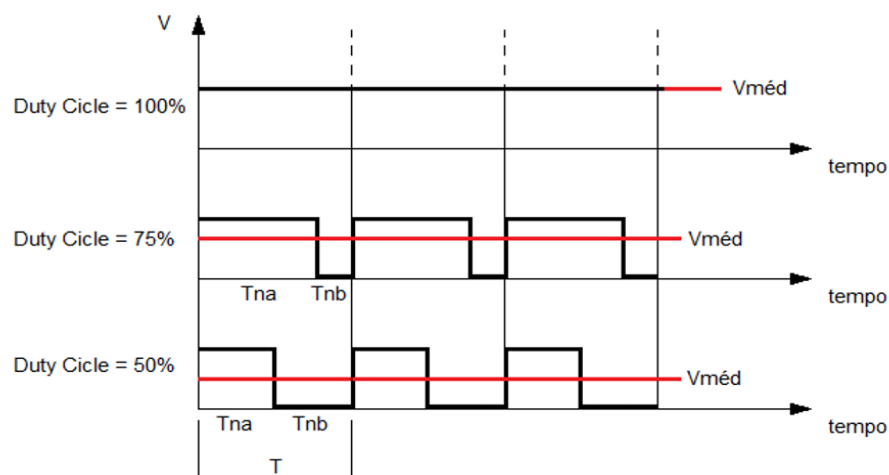
Onde:

- T_{na} : Tempo em que o sinal se encontra em nível lógico alto.
- T_{nb} : Tempo em que o sinal se encontra em nível lógico baixo.
- T : Período do sinal PWM

Em se tratando do microcontrolador ATMEGA utilizado no ARDUINO UNO, o sinal PWM é gerado em uma porta digital e seu *Duty Cycle* é calculado automaticamente, subtraindo a necessidade de implementação em forma de linhas de códigos da equação 4.2 (SILVA, 2014).

O motor CC interpreta o sinal PWM gerado como sendo um sinal completamente analógico, ou seja, mesmo havendo um “chaveamento” no sinal de PWM o motor funciona como se estivesse recebendo uma tensão analógica constante que é exatamente a tensão média resultante do ciclo de trabalho do sinal PWM. Logo, para controlar a velocidade de motores CC basta apenas controlar o valor de *Duty Cycle* do sinal. A Figura 26 mostra a relação entre tensão média ($V_{méd}$) aplicada ao motor e o ciclo de trabalho *Duty Cycle* (SILVA, 2014).

Figura 26 - Representação Sinal PWM

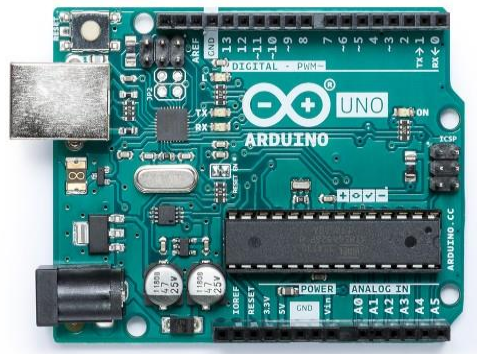


Fonte: SILVA, 2014.

4.4.6 ARDUINO UNO

O ARDUINO UNO é uma plataforma que tem sua construção baseada em um microcontrolador ATMEGA328 de 8 bits, fabricado pela ATMEL. Esta plataforma possui um *software* livre Arduino IDE, onde foi implementado um código para a aquisição dos dados lidos pelos sensores MPX5010DP e pelo sensor de fluxo de água, além de enviar o sinal PWM para o CI L295N, responsável pelo acionamento da bomba de 12V. A figura 27 apresenta o modelo utilizado neste trabalho.

Figura 27 - ARDUINO UNO

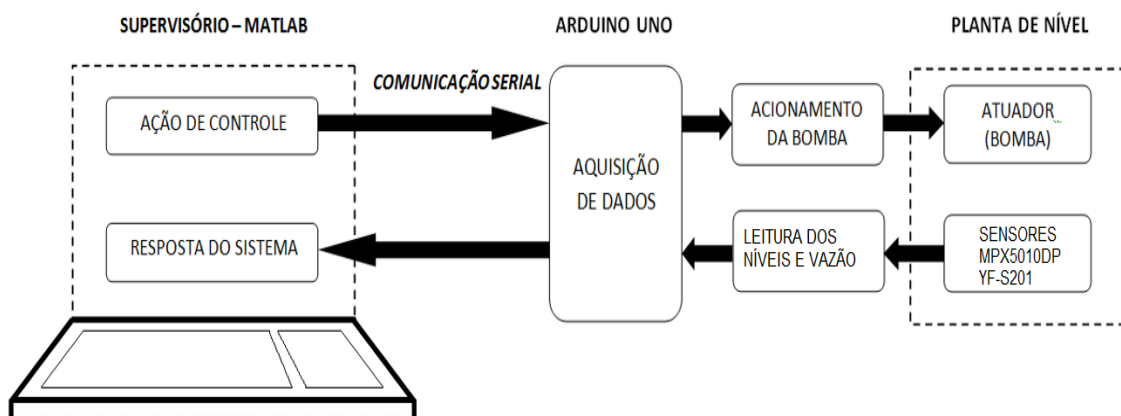


Fonte: ARDUINO, 2019.

A comunicação entre o sistema supervisorio desenvolvido neste trabalho e a plataforma ARDUINO UNO é realizada pelo *software* MATLAB®, desenvolvido pela MATHWORKS. Neste *software* está implementado o código de controle da planta de tanques acoplados, bem como o sistema supervisorio através da ferramenta *GUIDE*.

O ARDUINO UNO envia os dados obtidos pelos sensores para o *software* MATLAB®, que trata esses dados, os apresenta na tela supervisorio, realiza o controle do erro de sinal lido pelo sinal *set point* e envia um sinal na faixa de 0 – 255 para o ARDUINO UNO. A figura 28 apresenta o diagrama do funcionamento do sistema de comunicação.

Figura 28 - Diagrama de comunicação ARDUINO UNO - MATLAB®



Fonte: autoria própria.

A tabela 2 apresenta o resumo das especificações técnicas do ARDUINO UNO.

Tabela 2- Especificações técnicas do microcontrolador ARDUINO UNO

Tensão de operação	5Vcc
Tensão de entrada	6 a 20Vcc (Recomendado 7 a 12Vcc)
Entradas e saídas digitais	14 pinos (Incluindo 6 pinos PWM)
Entradas analógicas	6 pinos (A0 a A5)
Corrente CC entrada e saída	40mA
Saídas de tensão	3,3Vcc e 5Vcc
Memória flash	32Kb (0,5Kb dedicados ao <i>bootloader</i>)
SRAM	2Kb
EEPROM	1Kb
Velocidade de clock	16MHz
Conexão PC	Via USB

Fonte: autoria própria.

4.5 Software do sistema de tanques acoplados

O desenvolvimento dos códigos nos *softwares* Arduino IDE e MATLAB® foi baseado em linguagem C/C++. Estes códigos processam o sistema, controlam e monitoram o acionamento dos elementos de *hardware* e apresetam os dados em uma tela de supervisão. Para que haja a interação entre este dois códigos, foi desenvolvido no *software* MATLAB® linhas de comando para a leitura e escrita na porta serial, a qual o *hardware* ARDUINO UNO está conectado. A comunicação via porta serial é necessária para que seja possível enviar os comandos realizados pelo usuário a partir da tela de supervisão do MATLAB® para o ARDUINO, quanto receber os dados dos sensores existentes no sistema de tanques acoplados adquiridos pelo ARDUINO durante a execução do programa.

O *software* Arduino IDE é extremamente prático de se trabalhar [...], programado de forma intuitiva e suas configurações de portas são feitas de forma rápida, além de possuir funções e bibliotecas destinadas a vários tipos de aplicações. estrutura básica de programação do kit é composta por quatro etapas fundamentais: inclusão de bibliotecas, declaração de variáveis, laço void setup() para determinação das condições de operação das portas digitais e analógicas e laço void loop() onde a rotina principal do código é implementada. Em se tratando de ambiente

de programação, o *hardware* ARDUINO UNO, e demais kits disponibilizados pela fabricante ARDUINO, possui uma IDE (*Integrated Development Environment*) executável destinada à elaboração e compilação dos programas escritos para o equipamento (SILVA, 2014).

O código implementado no *software* Arduino IDE é responsável por receber e tratar os sinais provenientes dos sensores MPX5010DP e YF-S201, além de enviar o sinal PWM para o circuito de acionamento da bomba submersa. Para os sinais adquiridos pelos sensores de pressão, o código utiliza a implementação da função *map* () para converter os valores lidos (tanques completamente vazios, tanques completamente cheios e valores atuais) em valores de altura do líquido em cm. Para os sinais provenientes do sensor de fluxo de água, o código utiliza a implementação de uma interrupção para a leitura dos giros/ciclos, por meio da função *attachInterrupt* (). Os valores serão enviados para o MATLAB® por meio da comunicação serial, através da função *Serial.print* (). O sinal PWM é enviado para o CI L295N através da função *analogWrite* (). O valor deste sinal é proveniente do código implementado no MATLAB®. Para evitar o transbordamento de líquido, foi implementado linhas de comando que limitam o acionamento da bomba submersa caso o limite de altura do líquido nos tanques seja alcançado.

O código implementado no *software* MATLAB® é responsável pela arquitetura de controle e de interface gráfica, por meio da ferramenta *GUIDE*. O controle foi implementado através de um PID digital, e a partir do valor dos sensores recebido pelo *hardware* ARDUINO UNO se faz o ajuste do erro e um sinal na faixa de 0 – 255 é enviado para o código implementado no *software* Arduino IDE. Através da interface gráfica o usuário pode interagir com o sistema de tanques acoplados, modificando e visualizando as variáveis do processo de controle de nível em tanques acoplados.

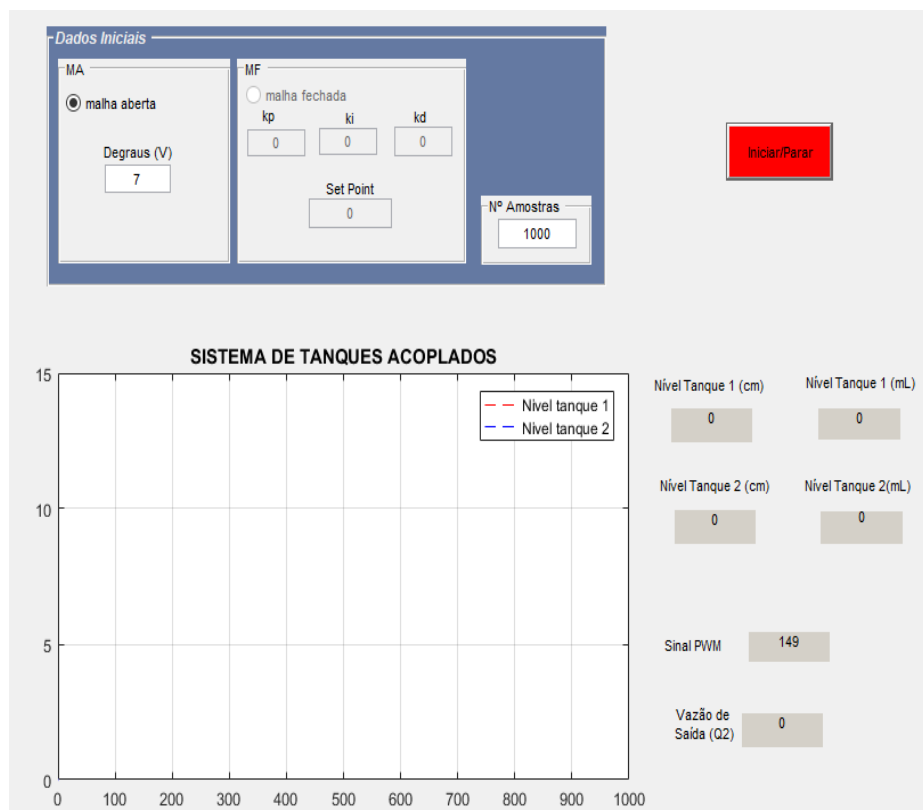
O MATLAB® (de MATrix LABoratory) é um programa produzido pela Mathworks, Inc., que consiste de uma ferramenta de computação numérica de análise e visualização de dados. Possui uma linguagem de programação de alto nível que proporciona um ambiente fácil de usar, já que os problemas e soluções são expressos em linguagem matemática, e não na linguagem de programação tradicional. A ferramenta MATLAB tem como principais objetivos a construção de gráficos, compilação de funções, manipulação de funções específicas de cálculo e variáveis simbólicas (MATHWORKS, 2019).

Os apêndices A e B apresentam as linhas de comando dos códigos implementados nos *software* Arduino IDE e MATLAB®, respectivamente. Os códigos foram baseados em linguagem de programação Arduino, que é uma linguagem de programação baseada em C/C++, com algumas pequenas modificações, para o *software* Arduino IDE e uma linguagem de alto nível baseada em C/Java para o *software* MATLAB®.

4.5.1 Interface Gráfica

A interface gráfica proporciona ao usuário a inicialização e finalização da comunicação serial, a escolha do modelo de teste realizado, o envio de dados de controle e a visualização da dinâmica do processo de controle do sistema tanques acoplados. A tela de supervisão foi desenvolvida utilizando a ferramenta GUIDE do *software* MATLAB®. A figura 29 apresenta a tela de supervisão desenvolvida neste trabalho.

Figura 29 - Tela de Supervisão do Sistema de tanques acoplados



Fonte: autoria própria.

A interface possibilita ao usuário a opção de realizar ensaios na planta em malha aberta (MA) ou malha fechada (MF). O programa inicia a comunicação serial entre o ARDUINO UNO e o computador através do botão “Iniciar/Parar”. Este botão também finaliza o programa, encerrando a comunicação serial e cria um arquivo “.txt” que contem os dados do sistema. O gráfico disponibiliza os dados obtidos do *software* Arduino IDE representando os níveis dos tanques. Os *displays* mostram os valores dos níveis dos tanques 1 e 2 em mL e cm, o sinal PWM que é enviado para o Arduino IDE e o valor da vazão de saída do tanque 2 (q_2). Quando um dos dois modos é selecionado, o outro modo de operação é bloqueado, evitando assim que valores indesejáveis sejam inseridos no algoritmo e invalidem o ensaio escolhido pelo usuário.

O painel de controle para o modo em MA será preenchido com o valor de tensão que deverá ser aplicado na bomba submersa, através do circuito de acionamento. O algoritmo implementado no MATLAB recebe este valor e o disponibiliza para o Arduino IDE, convertendo-o em um faixa de 0 a 255, onde 0 representa 0Vcc para a bomba e 255 representa 12V (tensão máxima fornecida pelo circuito de acionamento). O Arduino IDE transforma o valor recebido em um valor PWM, enviando-o para o CI L295N. Como medida de proteção para os componentes eletrônicos, o algoritmo envia uma mensagem ao usuário caso o valor preenchido seja maior que 12 ou menor que 0. Além do valor de tensão, este modo possibilita preencher o número de amostras que se deseja obter no ensaio em questão.

Para o modo em MF, o painel disponibiliza as opções de preenchimento para as variáveis de controle e o valor em que se deseja obter o nível do tanque 2 (*set point*). Neste modo também é possível ajustar o valor do número de amostras que se deseja obter para no ensaio. O algoritmo identifica os níveis dos tanques 1 e 2, compara o nível do tanque 2 (altura h_2) com o valor de *set point* e calcula o erro do sistema. Os valores definidos para as variáveis de controle K_p , K_i e K_d são enviados para o algoritmo que realiza o ajuste do erro e envia um sinal na faixa de 0 a 255 para o Arduino IDE conforme seja a implementação do controle. Durante toda a execução do programa é disponibilizado no gráfico os valores dos níveis dos tanques, possibilitando ao usuário a visualização do comportamento do controlador.

4.6 Protótipo desenvolvido

A figura 30 apresenta o protótipo do sistema de tanques acoplados desenvolvido neste trabalho.

Figura 30 - Protótipo desenvolvido



Fonte: autoria própria.

A tabela 3 abaixo relaciona o custo total dos componentes utilizados na confecção da planta de tanques acoplados.

Tabela 3- Relação do custo dos componentes

Descrição do componente	Valor (r\$)
2 tanques em acrílico	120,00
2 válvulas manuais ½" pvc	13,00
1 válvula manual ½" aço	8,00
2 sensores de pressão mpx5010dp	100,00
1 sensor de vazão de 1/2" yf-s201	25,00
1 ci l298n	20,00
1 bomba dc 12V - 240l/h	40,00
1 arduino uno	45,00
1 fonte 12V/10Aa colmeia	39,00
2 régua em acrílico	2,00
Total	412,00

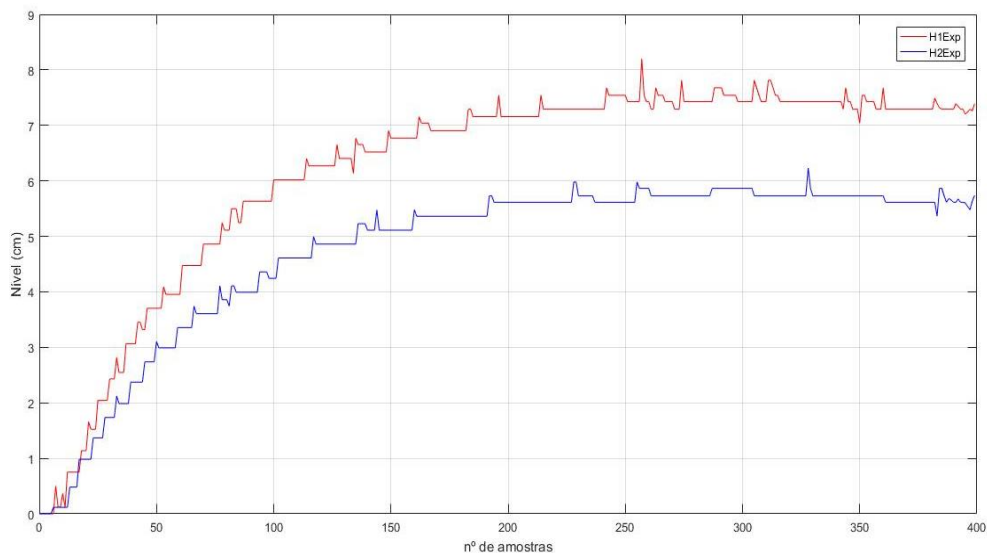
Fonte: autoria própria.

5 RESULTADOS EXPERIMENTAIS

5.1 Validação do modelo do sistema

Inicialmente, realizou-se testes experimentais para validar o modelo da planta obtido pelas equações 3.16 e 3.17. A partir de simulações no *SIMULINKTM* pode-se comparar o modelo da planta com os testes experimentais. Para a realização do teste experimental em malha aberta, aplicou-se um degrau na entrada no sistema de amplitude igual a 6V. Este valor representa uma vazão de entrada de 2L/min. A figura 31 apresenta os níveis dos tanques 1 e 2 obtidos via teste experimental.

Figura 31 - Teste experimental do sistema para uma entrada de 2L/min



Fonte: autoria própria.

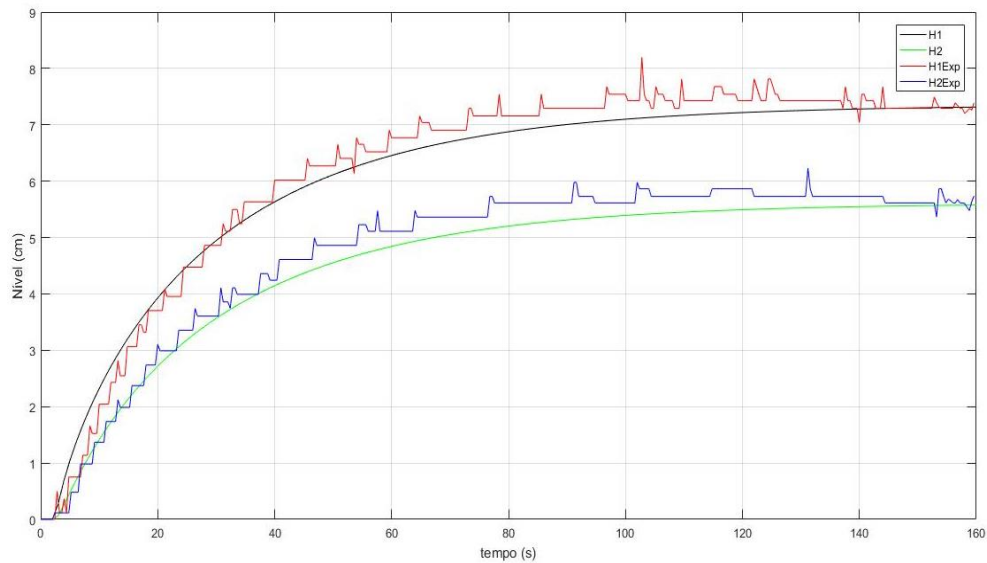
Para realizar a simulação do sistema, utilizou-se o diagrama de blocos apresentado na figura 19. As definições para as variáveis estão apresentadas na tabela 4. A figura 32 mostra a comparação entre a curva obtida para o teste experimental e para a simulação do sistema de tanques acoplados.

Tabela 4 - Dados para simulação

Degrau de entrada	2L/min	Tempo de amostragem	0.4
Abertura da válvula V_{12}	0 – 160s	Resistência da válvula V_{12}	3500
Abertura da válvula V_2	0 – 160s	Resistência da válvula V_2	6050

Fonte: autoria própria.

Figura 32 - Comparação entre teste experimental e simulação para uma entrada de 2L/min



Fonte: autoria própria.

Substituindo os valores de resistência das válvulas definidos na tabela 5 e o valor da área da base A_r dos tanques na equação 3.29, obteve-se a função de transferência $G(s)$ do sistema linearizado. A equação 6.1 apresenta planta $G(s)$. Considerando que esta equação representa o sistema linearizado (sem a presença das não-linearidades típicas de um sistema de nível), a simulação da dinâmica dos níveis dos tanques se aproxima muito da dinâmica dos níveis do sistema real, obtidos através de testes experimentais.

$$G(s) = \frac{10.63}{s^2 + 0.1421s + 0.001757} \quad (6.1)$$

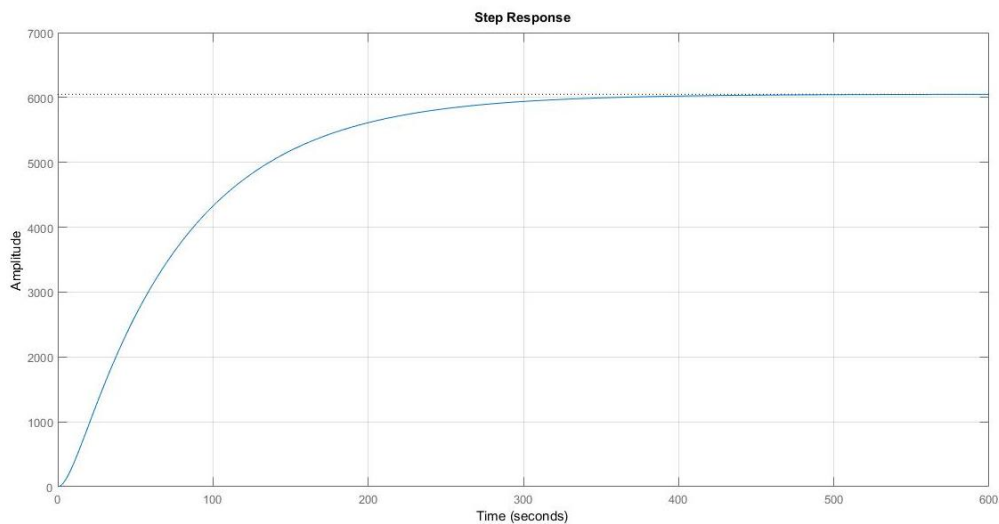
Analisando a função de transferência $G(s)$, obtemos o valor do fator de amortecimento ζ do sistema. Realizando os cálculos necessários, encontra-se o valor de $\zeta = 1.695$, caracterizando a resposta do sistema a uma entrada degrau como sobre-amortecida. Portanto, os pólos da planta $G(s)$ são reais, negativos e distintos, dando uma característica não oscilatória ao sistema.

5.2 Projeto dos controladores

Utilizando a equação 6.1 que representa a função de transferência do sistema de tanques acoplados, realizou-se simulações da resposta do sistema, para uma entrada degrau unitário, em malha aberta e em malha fechada.

Para o teste da resposta em malha aberta, utilizou-se a função $step(G)$ no MATLAB®, onde $G(s)$ representa a função de transferência do sistema. A figura 33 apresenta a resposta do sistema em malha aberta.

Figura 33 - Resposta do sistema em malha aberta

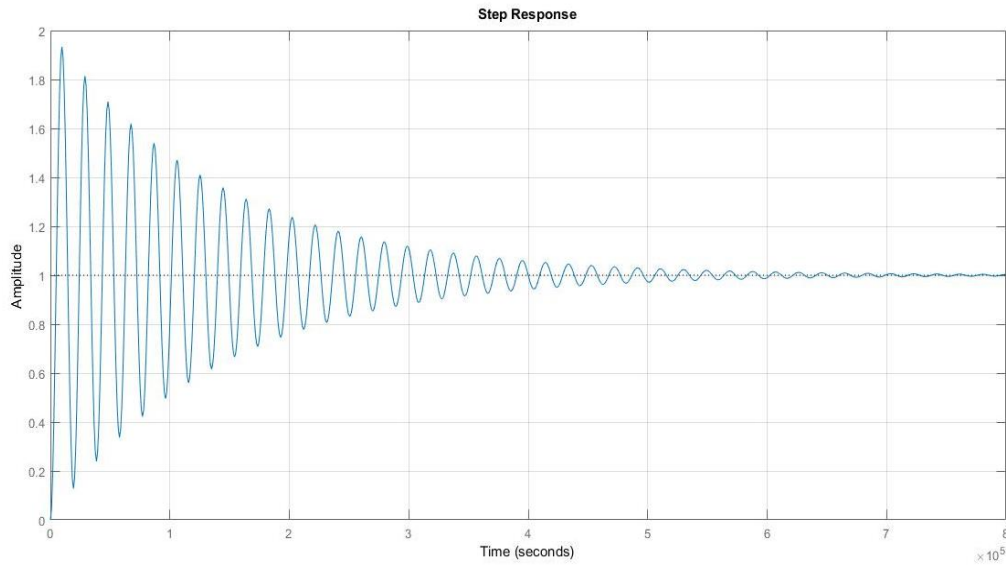


Fonte: autoria própria.

A resposta do sistema tem o comportamento de um sistema de segunda ordem sobre-amortecido, assim como demonstrado na simulação para validação do sistema. O sinal de amplitude atinge o seu valor de regime em um intervalo de tempo elevado. Este fato ocorre pois aplica-se um degrau unitário, que representa 1.33L/min. Como as dimensões dos tanques estão em metros, a amplitude assumirá valores elevados.

Para o teste da resposta em malha fechada, utilizou-se a função $feedback(G,H)$, onde $H(s)$ representa um ganho unitário e $G(s)$ representa a função de transferência do sistema, juntamente com a função $step(G_{mf})$, onde $G_{mf}(s)$ representa a função de transferência em malha fechada. A figura 34 apresenta a resposta do sistema em malha fechada.

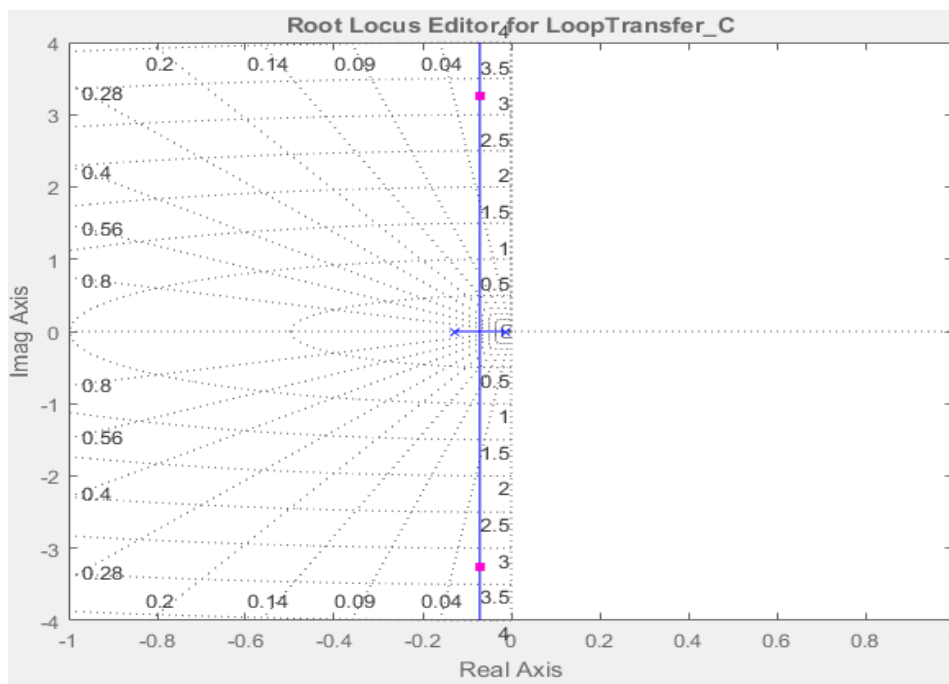
Figura 34 - Resposta do sistema em malha fechada



Fonte: autoria própria.

A resposta do sistema tem um comportamento oscilatório, com um sobre-sinal muito elevado (próximo a 2x o valor de *set point*), proporcionando um tempo de amortecimento alto. Utilizando a função *rltool(G)*, pode-se encontrar a mesma resposta do sistema em malha fechada apresentada na figura 34 e o lugar geométrico das raízes. A figura 35 apresenta o lugar geométrico das raízes, demonstrando os pólos do sistema.

Figura 35 - Lugar geométrico da raízes



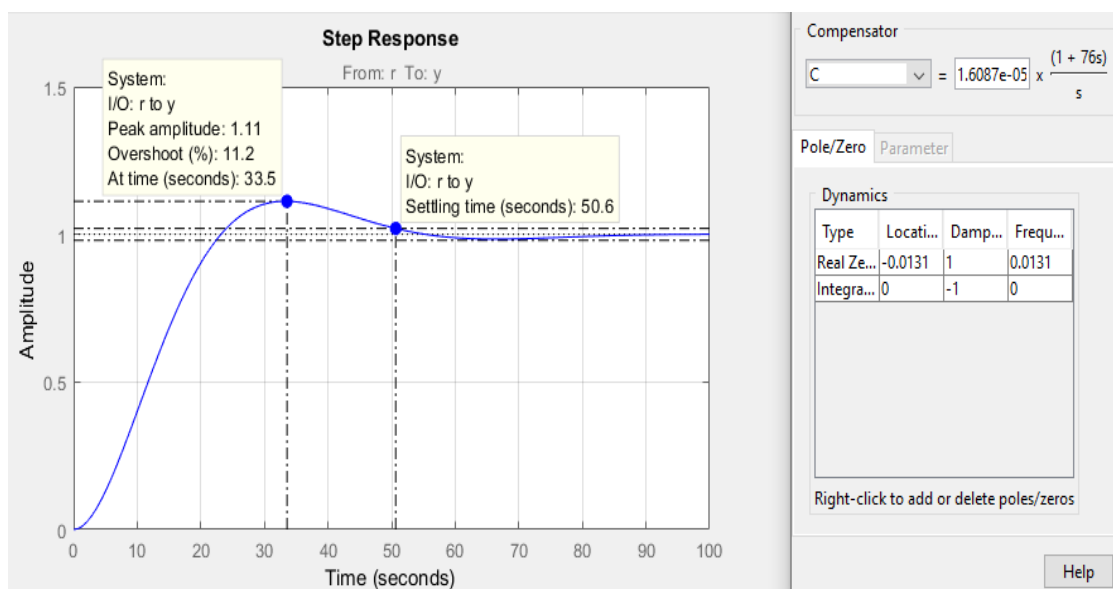
Fonte: autoria própria.

5.2.1 Projeto do controlador PI

O controlador PI possui por definição um pólo na origem e um zero em K_i/K_p . Pode-se definir que o zero deste controlador é dinâmico, mudando de posição conforme os valores de K_p e K_i , portanto é possível alocá-lo no lugar geométrico das raízes de forma que se tenha uma região ampla para variação do ganho K .

Utilizando a ferramenta $rltool(G)$ pode-se alocar o pólo e o zero do controlador PI e observar o comportamento da resposta ao degrau do sistema quando variamos o valor do ganho K . Foi definido o valor de *overshoot* entre 7% e 15% e o tempo de acomodação em torno de 30 a 90s. A figura 36 apresenta a resposta do sistema e os valores dos parâmetros do controlador PI.

Figura 36 – Projeto controlador PI com $M_p = 7\text{-}15\%$ e $t_s = 30\text{-}90\text{s}$



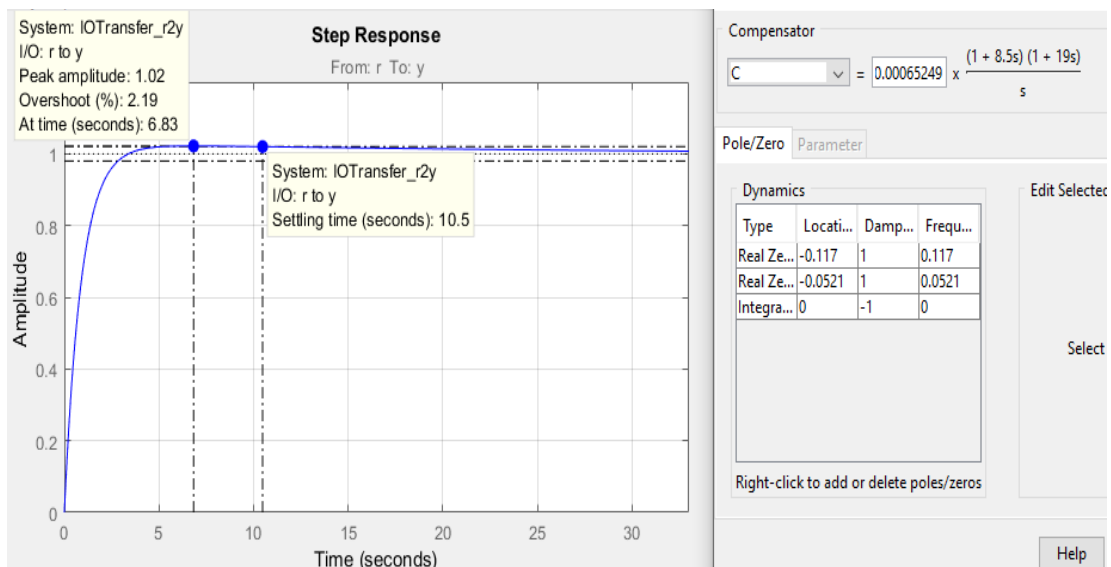
Fonte: autoria própria.

Obteve-se como resultado um valor de *overshoot* igual a 11.2% e um tempo de acomodação igual a 50.6s, o que já se apresenta como uma resposta rápida para o sistema desenvolvido. O zero foi alocado entre os dois pólos reais, mais próximo a origem. Os valores dos parâmetros para este controlador são $K_p = 0.00122$ e $K_i = 1.6e-5$.

5.2.2 Projeto do controlador PID

O projeto do controlador PID consiste em realizar o projeto do controlador PI, com um pólo na origem e um zero em K_i/K_p , e posteriormente alocar o zero do controlador PD, em K_p/K_d , de maneira a diminuir o sobre-sinal e melhorar a resposta transitória do sistema. A figura 37 apresenta a resposta do sistema e os valores dos parâmetros do controlador PID.

Figura 37 - Projeto do controlador PID



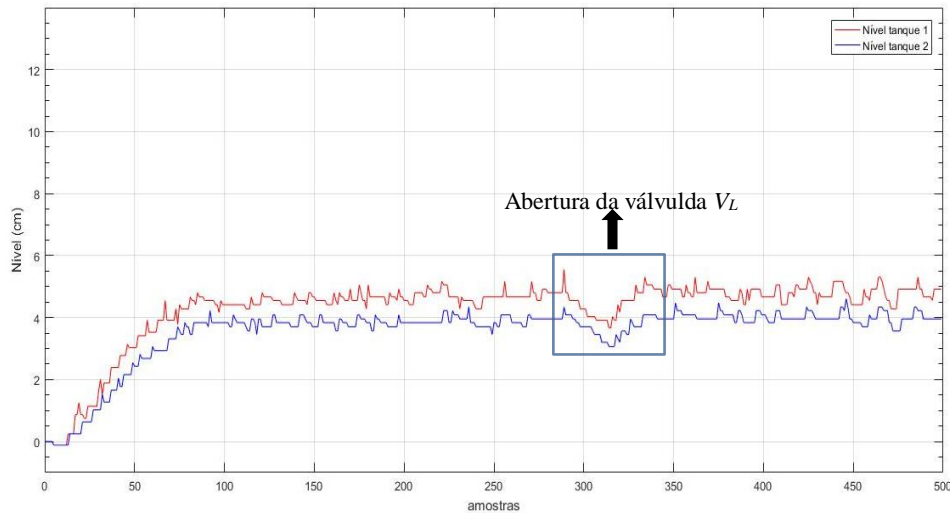
Fonte: autoria própria.

Obteve-se um valor de *overshoot* igual a 2.19% e um tempo de amortecimento igual a 10.5s. Como era esperado, o controlador PID possui um sinal de amortecimento pequeno e um valor de sobre-sinal praticamente nulo. Os valores para os parâmetros deste controlador são $K_p = 0.0181$, $K_i = 0.000652$ e $K_d = 0.107$.

5.3 Resultados do sistema de controle

A figura 38 apresenta o resultado do controlador PI na planta desenvolvida para o primeiro teste. Foi definido um valor de *set point* igual a 4cm para o nível do tanque 2. Os valores dos parâmetros foram $K_p = 0.00122$ e $K_i = 1.6e-5$.

Figura 38 - Teste experimental 1 do controlador PI



Fonte: autoria própria.

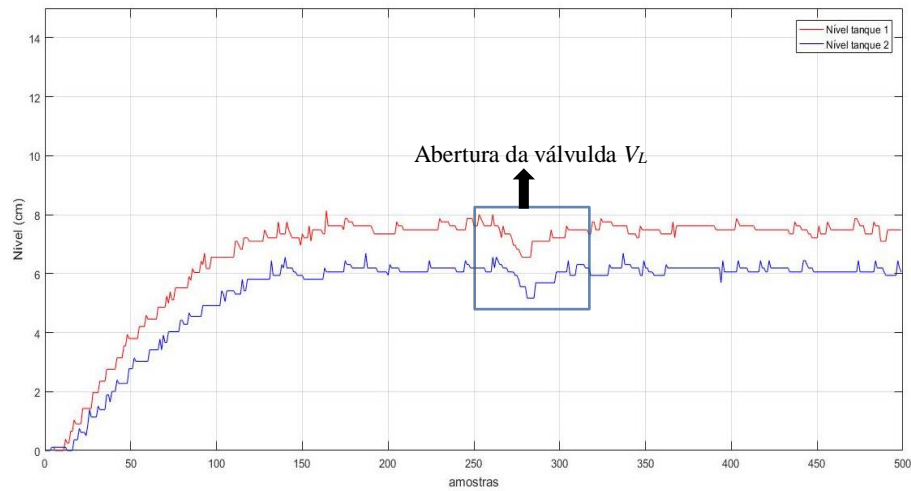
O teste apresentou um bom resultado para o controlador PI. Para um período menor do que 100 amostras, o nível do tanque 2 se encontra muito próximo ao valor de *set point*, mostrando que o controlador tem uma resposta rápida para o sistema desenvolvido. Porém, diferente do que se esperava, a resposta apresentada neste teste não apresentou um sobre-sinal.

Após o nível do tanque 2 se estabilizar, tendo o valor igual ao *set point*, a válvula V_L é aberta, representando um distúrbio ao sistema. O controlador apresenta uma resposta transitória rápida, e sem oscilações, validando os valores dos parâmetros obtidos no projeto do controlador PI.

Para o segundo teste foi definido um valor de *set point* igual a 6cm para o nível do tanque 2. Os valores dos parâmetros foram $K_p = 0.00122$ e $K_i = 1.6e-5$. A figura 39 apresenta o resultado do controlador PI para o segundo teste.

Neste segundo teste, a resposta do do sistema de controle apresentou um pequeno sobre-sinal e, assim como no teste anterior, um tempo de acomodação pequeno, de aproximadamente 100 amostras. No período próximo a 250 amostras, a válvula V_L é aberta e a dinâmica do controlador rapidamente trata o sinal de erro, corrigindo o nível do tanque 2 para o valor de *set point*.

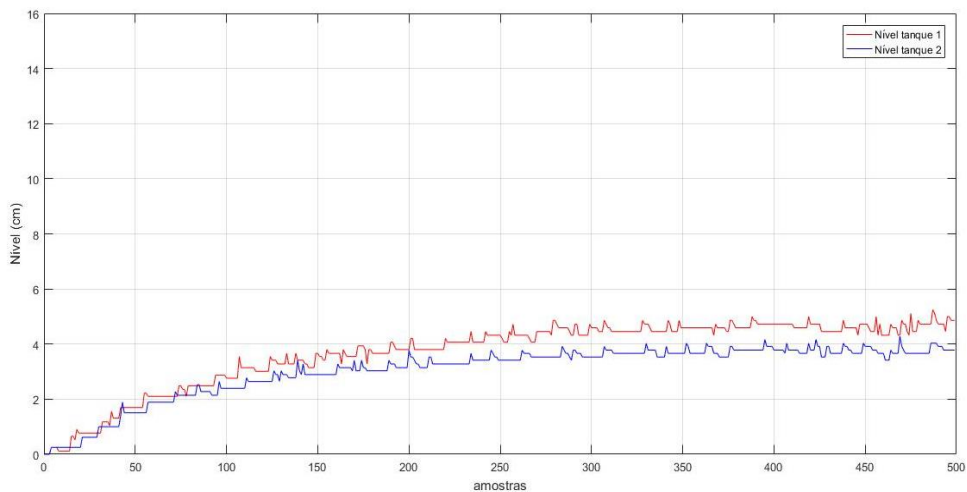
Figura 39 - Teste experimental 2 do controlador PI



Fonte: autoria própria

A figura 40 apresenta o resultado do controlador PID. O valor de *set point* foi definido como no primeiro teste do controlador PI (4cm) para que se pudesse fazer um comparativo dos resultados. Os valores dos parâmetros foram $K_p=0.0181$, $K_i=0.000652$ e $K_d=0.107$.

Figura 40 - Teste experimental do controlador PID



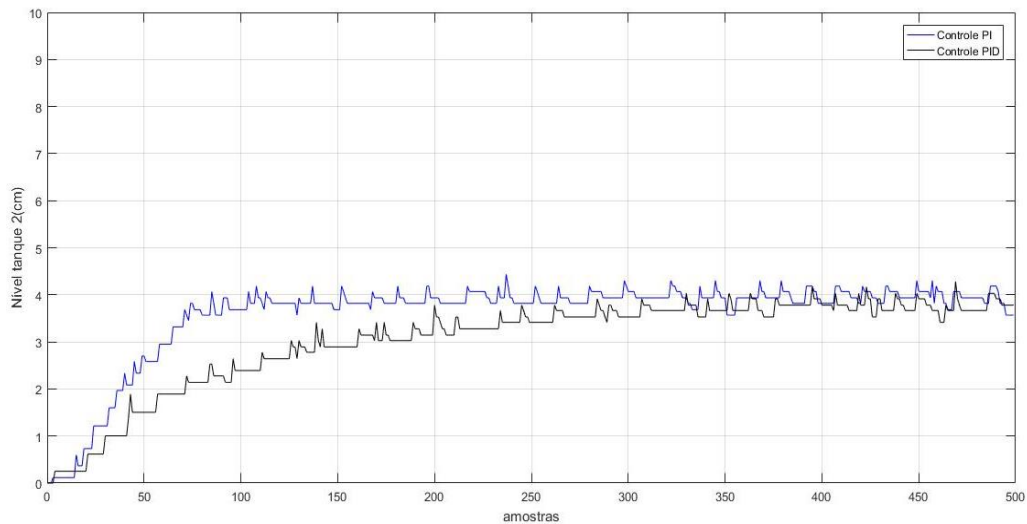
Fonte: autoria própria

O controlador PID apresentou um resultado muito diferente do esperado. Pela simulação do projeto deste controlador, o resultado que se esperava era um controle mais robusto e com uma resposta rápida. O teste experimental apresentado na figura 40 mostra uma resposta muito lenta.

O parâmetro derivativo do controlador PID ajusta a resposta do sistema tornando-a mais suave. Entretanto, o sistema apresenta um sinal de ruído elevado e como o controle PD é sensível a frequências elevadas, ele amplifica o ruído presente no sistema. Desta maneira, a resposta do controlador PID tem um comportamento diferente do esperado, controlando o sistema de forma lenta devido ao processo derivativo que tenta prevê de que maneira o sinal de erro irá se comportar.

A figura 41 apresenta um comparativo entre os sinais de resposta dos controladores PI e PID.

Figura 41 - Comparação entre as respostas dos controladores PI e PID



Fonte: autoria própria

Observamos que o controladore PID possui um atraso muito grande quando comparado com o controlador PI. Portanto, pode-se afirmar que para o sistema desenvolvido, o controlador PI se mostrou mais aplicável, com uma boa resposta transitória.

6 CONCLUSÃO

O trabalho desenvolvido apresentou resultados satisfatórios para a modelagem e construção do sistema de tanques acoplados. O embasamento teórico proporcionou uma base sólida para o desenvolvimento do protótipo, apresentando simulações que nortearam as etapas de aquisição dos sensores e atuadores do projeto. Os resultados apresentados nas simulações para o sistema em malha aberta se aproximaram de forma satisfatória dos resultados obtidos nos testes experimentais. O projeto do controlador foi baseado no método lugar das raízes, uma vez que este conteúdo faz parte da grade curricular do curso de engenharia elétrica da Universidade Federal do Espírito Santo, proporcionando uma aplicação prática dos elementos teóricos desenvolvidos em sala de aula.

Comparando os resultados experimentais obtidos pelos controladores PI e PID, é possível afirmar que o controlador PI tem respostas mais satisfatórias (resposta rápida para o sistema desenvolvido e com sobre-sinal pequeno) que o controlador PID. Mesmo com a presença de uma perturbação proveniente da válvula V_L , o controlador PI manteve o nível do tanque 2 igual ao valor de referência.

O controlador PID apresentou uma resposta lenta devido a ação derivativa do controlador, que tem a característica de corrigir o erro de forma mais suave, porém como o sistema possui sinais ruidosos, esta ação amplifica estes sinais e retarda a dinâmica rápida do controlador. Estes sinais existem devido a diferença entre a faixa de leitura do sensor de pressão e a taxa de amostragem do microcontrolador ATMEGA328. A dinâmica do processo de enchimento do tanque também provoca ruídos nos valores lidos pelos sensores, devido ao comportamento do líquido dentro do tanque.

Desta maneira, podemos afirmar que algumas considerações deverão ser feitas em trabalhos futuros para que se obtenha uma melhor resposta do controlador PID.

- Desenvolvimento de filtros digitais para os sinais dos sensores N_1 e N_2 ;
- Desenvolvimento de um sistema multivariável, proporcionando o controle automático das válvulas V_{12} e V_2 ;

- Desenvolvimento e comparação de outros métodos de sintonia para o sistema de controle;
- Desenvolvimento de funções na tela supervisorio que proporcionem ao usuário a escolha de qual variável será manipulada e qual variável se deseja controlar.
- Substituição do microcontrolador ATMEGA328 por um processador com maior taxa de resolução.

Estes ajustes poderão proporcionar uma melhor experiência ao usuário e garantir uma aplicação mais ampla dos conteúdos estudados em sala de aula, além de produzir resultados mais satisfatórios para o sistema de controle, em especial na aplicação do controlador PID.

O projeto desenvolvido ficará a disposição dos alunos e dos professores que desejarem utilizá-lo para estudos de sistemas não-lineares, sistemas de controle, sintonia de controladores e identificação e correção de falhas em processos industriais. Assim, o projeto terá uma relevante contribuição para o ensino e pesquisa na área de automação e controle.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDREA, C. Q. Método do Lugar das Raízes. Universidade Tecnológica Federal do Paraná. Curitiba, 2011. Disponível em: <http://paginapessoal.utfpr.edu.br/camaral/laboraorio-de-sistemas-de-controle/18_1%20-20%20Lugar%20das%20Raizes%20UTFPR.pdf>. Acessado em 20 jun. 2019.

ARDUINO, G. **Arduino Uno Rev3**. 2019. Disponível em:<<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acessado em 02 jan. 2019.

BLANKE, M.; KINNAERT, M.; LUNZE, J.; STAROSWIECKI, M. **Diagnosis and fault-tolerant control**. 2. ed. Verlag Berlin Heidelberg: Springer, 2006.

GOSMANN, H. L. **Um Sistema multivariável de tanques acoplados para avaliação de técnicas de controle**. 2002. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Tecnologia, Universidade de Brasília, Distrito Federal, 2002.

MARGON, G. M. **Especificação de planta piloto para estudo e pesquisa em modelagem e projeto de controladores**. 2010. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Colegiado do Curso de Engenharia Elétrica, Vitória, 2010.

MATIAS, J. **Teoria de controle PID**. Mecatrônica Atual, n. 3, abr. 2002. Disponível em: <http://coral.ufsm.br/beltrame/arquivos/disciplinas/medio_automacao_industrial/Artigo_Teoria_controle_PID.pdf>. Acessado em 29 mai. 2018.

MATHWORKS, T. **MATLAB Documentation**. 2019. Disponível em: <<https://www.mathworks.com/help/matlab/>>. Acessado em 12 fev. 2019.

OGATA, K. **Engenharia de controle moderno**. 4. ed. São Paulo: Pearson Prentice Hall, 2003

PORTA, L.D. **Sensor de fluxo de água para arduino 1-30L/min.** abr. 2016. Usinainfo – eletrônica e robótica. Disponível em:< <https://blog.usinainfo.com.br/sensor-de-fluxo-de-agua-para-arduino-1-30-lmin/>>. Acessado em 29 jan. 2019.

RAMOS, A. P.; WENSE, G. L. B. **Sistema didático de nível de líquidos.** 2008. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade de Brasília. Faculdade de Tecnologia, Distrito Federal, 2008.

REODIQUE, A.; Shultz, W. Noise Considerations for Integrated Pressure Sensors. **Sensor and Systems Applications Engineering**, rev 2, mai. 2005. Disponível em: <https://www.nxp.com/files-static/sensors/doc/app_note/AN1646.pdf>. Acessado em 30 jan. 2019.

RYAN, J. **Sensores diferenciais de pressão Freescale/NXP.** abr. 2013. Disponível em:< <https://www.automalabs.com.br/sensores-de-pressao-freescale-mpx5010-e-mpx5050/>>. Acessado em 22 ago 2018.

SERAPHIM, A. L. **Modelagem e controle de nível em um sistema de dois tanques acoplados.** 2008. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade São Francisco. 2008.

SILVA, R. A. **Desenvolvimento de módulo de controle de nível.** 2014. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) - Universidade Tecnológica Federal do Paraná. 2014.

THOMSEN, A. **Motor DC com Driver Ponte H L298N.** Mar. 2013. Disponível em:< <https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-l298n/>>. Acessado em 28 jan. 2019.

ZHAN, W. **Robustness Analysis of Pulse Width Modulation Control of Motor Speed.** World Congress of Engineering and Computer Science. 2007. San Francisco, oct. 2007.

APÊNDICE A – ALGORITMO DE AQUISIÇÃO DE DADOS - ARDUINO

```
#define pwm 6
```

```
int velocidade = 100;
```

```
const float altmax1 = 119.0; // valor lido pelo sensor para o tanque1 completamente cheio  
(10cm)
```

```
const float altmin1 = 37.0; // valor lido pelo sensor para o tanque1 completamente vazio
```

```
const float altmax2 = 123.0; // valor lido pelo sensor para o tanque2 completamente cheio  
(10cm)
```

```
const float altmin2 = 40.0; // valor lido pelo sensor para o tanque2 completamente vazio
```

```
float vazao; //Variável para armazenar o valor em L/min
```

```
volatile int contaPulso; //Variável para armazenar a quantidade de pulsos
```

```
int s1=0;
```

```
int s1_1=0;
```

```
int s2=0;
```

```
int s2_2=0;
```

```
int k=1;
```

```
int valor_pwm = 0;
```

```
void incpulso ()
```

```
{
```

```
    contaPulso++; //Incrementa a variável de contagem dos pulsos
```

```
}
```

```
void setup()
```

```
{
```

```
pinMode(pwm, OUTPUT); // define porta 9 como saída
pinMode(3, INPUT); // define pino 3 como entrada
attachInterrupt(digitalPinToInterrupt(3),  incpulso,  RISING); //Configura o pino
3(Interrupção 1) para trabalhar como interrupção
Serial.begin(9600); //inicializa o monitor serial
}

void loop()
{
  contaPulso = 0; //Zera a variável para contar os giros por segundos
  sei();
  delay (velocidade); //Aguarda 0,1 segundo
  cli();
  vazao = contaPulso / 7.5; //Converte para L/min

  //////////// filtro para sensor tanque 1//////////
  s1 = s1_1;
  s1_1 = analogRead(A0);

  if(s1_1 - s1<=2)
  {
    if(s1_1 - s1 > 0 )
    {
      s1_1 = s1;
    }
    s1=0;
  }
  //Serial.print(s1_1);
  //Serial.print(" ");

  //////////// filtro para sensor tanque 2//////////
  s2 = s2_2;
  s2_2 = analogRead(A1);
```

```
if(s2_2 - s2<=2)
{
  if(s2_2 - s2 > 0 )
  {
    s2_2 = s2;
  }
  s2=0;
}
//Serial.println(s2_2);
// Serial.print(" ");

if (Serial.available() && k==1)
{
  valor_pwm = Serial.parseInt();

  k = 0;
}

if (k==0)
{
  float nivel1 = map(s1_1,altmin1,altmax1,0,518.4); // converte o valor lido pelo sensor 1 para
o valor em mL de água (nivel em 10cm)
  float nivel2 = map(s2_2,altmin2,altmax2,0,518.4); // converte o valor lido pelo sensor 2 para
o valor em mL de água (nivel em 10cm))

  delay (velocidade);

  imprimir_status(nivel1,nivel2,vazao);
}
}
```

```
// Função para imprimir o nível do tanque 1 e acionar a bomba
void imprimir_status(float nivel1,float nivel2, float vazao){
  Serial.println(nivel1,2);
  Serial.println(nivel2,2);
  Serial.println(vazao);

  //////////// limita o sinal pwm para o valor máximo de água no tanque 1 ////////////
  if (s1_1 >= altmax1)
  {
    analogWrite(pwm,0);
  }
  else
  {
    analogWrite(pwm, valor_pwm);
  }

  k=1;
}
```

APÊNDICE B – ALGORITMO DE CONTROLE E INTERFACE - MATLAB

```

function varargout = Tanques_Acoplados(varargin)
% Tanques_Acoplados MATLAB code for Tanques_Acoplados.fig
%   Tanques_Acoplados, by itself, creates a new Tanques_Acoplados or
raises the existing
%   singleton*.
%
%   H = Tanques_Acoplados returns the handle to a new Tanques_Acoplados
or the handle to
%   the existing singleton*.
%
%   Tanques_Acoplados('CALLBACK',hObject,eventData,handles,...) calls
the local
%   function named CALLBACK in Tanques_Acoplados.M with the given input
arguments.
%
%   Tanques_Acoplados('Property','Value',...) creates a new
Tanques_Acoplados or raises the
%   existing singleton*. Starting from the left, property value pairs
are
%   applied to the GUI before Tanques_Acoplados_OpeningFcn gets called.
An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to Tanques_Acoplados_OpeningFcn via
varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Tanques_Acoplados

% Last Modified by GUIDE v2.5 05-Jun-2019 23:31:48

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Tanques_Acoplados_OpeningFcn, ...
                  'gui_OutputFcn',  @Tanques_Acoplados_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

```

```
% End initialization code - DO NOT EDIT

% --- Executes just before Tanques_Acoplados is made visible.
function Tanques_Acoplados_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Tanques_Acoplados (see VARARGIN)

% Choose default command line output for Tanques_Acoplados
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Tanques_Acoplados wait for user response (see UIRESUME)
% uiwait(handles.figure1);

%% INICIA COM '0' CADA LACUNA DO SUPERVISÓRIO
set(handles.k, 'String', '0');
set(handles.ki, 'String', '0');
set(handles.kd, 'String', '0');
set(handles.amostras, 'String', '0');
set(handles.sp, 'String', '0');
set(handles.degrau, 'String', '0');
%% INICIALIZAÇÃO DO GRAFICO " RESPOSTA DO SISTEMA "
axes(handles.resp_sistema);

% --- Outputs from this function are returned to the command line.
function varargout = Tanques_Acoplados_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton1

%% VARIÁVEIS DO SISTEMA
global n_amostras s_point tensao kp ki kd nivel_t1 nivel_t2 alt_t1 alt_t2
iniciar_ensaio = get(handles.togglebutton1, 'Value');
pwm = '0';
```



```

valvula='0';%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
den = 0;
nivel2=0.0;
nivell1 = 0.0;
vazao=0.0;
alt1_t1= 0.0;
alt1_t2= 0.0;
saida = 0;
nivel_t1 = zeros(1,n_amostras);
nivel_t2 = zeros(1,n_amostras);
alt_t1 = zeros(1,n_amostras);
alt_t2 = zeros(1,n_amostras);
output = zeros(1,n_amostras);
erro_k = 0;
erro_k1 = 0;
erro_k2 = 0;
parc1 = 0;
parc2 = 0;
parc3 = 0;
T_samp = 0.25; % TEMPO DE AMOSTRAGEM 0.25

%% ROTINA PRINCIPAL DE CONTROLE
if iniciar_ensaio==1
    set(handles.togglebutton1,'BackgroundColor','Green');

    handle(handles.resp_sistema);

    % INICIALIZAÇÃO DA COMUNICAÇÃO SERIAL
    delete(instrfind({'Port'},{'COM3'}));
    %CRIA CONEXÃO COM A SERIAL COM3
    s = serial('COM3','BaudRate',9600);
    warning('off','MATLAB:serial:fscanf:unsuccessfulRead');
    %ABRE PORTA
    fopen(s);
    %fprintf(s,pwm);
    %fprintf(s,s_point);

    for k=1 : n_amostras

        a = get(handles.ma,'Value');

        if (get(s, 'BytesAvailable') == 0)
        else
        if (get(s, 'BytesAvailable') >0.00)
            nivell1 = fscanf(s,'%f');
            nivel2 = fscanf(s,'%f');
            vazao = fscanf(s,'%f');
        end
        end

        %%%% conversão para tanque 1 %%%%
        alt1_t1 = (nivell1/(0.072*0.072))/10000; %converte para o valor em altura
do nível
        alt_t1(k)= alt1_t1;
        nivel_t1(k) = nivell1;

        %%%% conversão para tanque 2 %%%%
        alt1_t2 = (nivel2/(0.072*0.072))/10000; %converte para o valor em altura
do nível

```

```

alt_t2(k)= alt1_t2;
nivel_t2(k) = nivel2;

if a==1 % SELECIONA ENSAIO EM MALHA ABERTA
den = 12; %trocar para 12
saida = tensao;

output(k)= saida;

else
if a==0 % SELECIONA ENSAIO EM MALHA FECHADA
den = 12; %trocar para 20

% erro_k = s_point - nivel_t1(k); % CÁLCULO DO ERRO
erro_k = s_point - alt_t2(k)

% AÇÃO DE CONTROLE PID

parc1 = kp*(erro_k - erro_k1); % AÇÃO PROPORCIONAL
parc2 = ((ki*T_samp)/2)*(erro_k + erro_k1); % AÇÃO INTEGRAL
parc3 = (kd/T_samp)*(erro_k - 2*erro_k1 + erro_k2); % AÇÃO DERIVATIVA

saida = saida + parc1 + parc2 + parc3;

erro_k2 = erro_k1;
erro_k1 = erro_k;

if saida>12
saida = 12;
else
if saida<=0

saida = 0;

end
end

end
end

pwm =(saida*255)/den;
pwm = round(pwm);
pwm = num2str(pwm);

%imprimir a saída pwm
set(handles.text9,'string',pwm);
%imprimir o valor do nível do tanque 1 em mL
set(handles.text11,'string',nivel1);
%imprimir o valor do nivel do tanque 1 em cm
set(handles.text13,'string',alt1_t1);

%imprimir o sinal de vazão lido pelo sensor
set(handles.text21,'string',vazao);
%imprimir o valor do nível do tanque 2 em mL
set(handles.text17,'string',nivel2);
%imprimir o valor do nivel do tanque 2 em cm

```

```

set(handles.text19, 'string', alt1_t2);

fprintf(s, pwm);

plot(handles.resp_sistema, [1:k], [alt_t1([1:k])], 'r--');
hold on
plot(handles.resp_sistema, [1:k], [alt_t2([1:k])], 'b--');
legend('Nivel tanque 1', 'Nivel tanque 2');
title('SISTEMA DE TANQUES ACOPLADOS');
grid on;
axis([0 n_amostras 0 15]);

pause(0.25); %0.25

if get(handles.togglebutton1, 'Value')==0
    iniciar_ensaio = 0;
    set(handles.togglebutton1, 'BackgroundColor', 'Red');
    msgbox('ENSAIO CANCELADO');
    break
end

end
fprintf(s, '0'); % seta a bomba DC com 0 Volts
delete(instrfind({'Port'}, {'COM3'})); % Deleta porta COM3 e finaliza a
comunicação
clear s;
inicia_ensaio = 0;
set(handles.togglebutton1, 'BackgroundColor', 'Red');
uisave({'alt_t1', 'output'}); %SALVA DADOS DO ENSAIO
end
guidata(hObject, handles)

% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over
togglebutton1.
function togglebutton1_ButtonDownFcn(hObject, eventdata, handles)
% hObject handle to togglebutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes during object creation, after setting all properties.

function togglebutton1_CreateFcn(hObject, eventdata, handles)
% hObject handle to togglebutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

function k_Callback(hObject, eventdata, handles)
% hObject handle to k (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of k as text
% str2double(get(hObject, 'String')) returns contents of k as a
double
global kp
kp_1 = get(handles.k, 'String');
kp = str2num(kp_1);

```

```

if kp<0
    msgbox('NÃO É POSSÍVEL UTILIZAR NÚMEROS NEGATIVOS. DEFINA UM NOVO
NÚMERO');
    set(handles.k,'String',num2str(0));
    kp = 0;
end
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function k_CreateFcn(hObject, eventdata, handles)
% hObject    handle to k (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ki_Callback(hObject, eventdata, handles)
% hObject    handle to ki (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ki as text
%         str2double(get(hObject,'String')) returns contents of ki as a
double
global ki
ki_1 = get(handles.ki,'String');
ki = str2num(ki_1);
if ki<0
    msgbox('NÃO É POSSÍVEL UTILIZAR NÚMEROS NEGATIVOS. DEFINA UM NOVO
NÚMERO');
    set(handles.k,'String',num2str(0));
    ki = 0;
end
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function ki_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ki (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function kd_Callback(hObject, eventdata, handles)
% hObject    handle to kd (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of kd as text
% str2double(get(hObject,'String')) returns contents of kd as a
double
global kd
kd_1 = get(handles.kd,'String');
kd = str2num(kd_1);
if kd<0
    msgbox('NÃO É POSSÍVEL UTILIZAR NÚMEROS NEGATIVOS. DEFINA UM NOVO
NÚMERO');
    set(handles.k,'String',num2str(0));
    kd = 0;
end
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function kd_CreateFcn(hObject, eventdata, handles)
% hObject handle to kd (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function amostras_Callback(hObject, eventdata, handles)
% hObject handle to amostras (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amostras as text
% str2double(get(hObject,'String')) returns contents of amostras as
a double
global n_amostras
amostra_1 = get(handles.amostras,'String');
n_amostras = str2num(amostra_1);
if n_amostras<100
    msgbox('UTILIZAR NÚMERO DE AMOSTRAS >= 1000');
    set(handles.amostras,'String',num2str(0));
    n_amostras = 0;
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function amostras_CreateFcn(hObject, eventdata, handles)
% hObject handle to amostras (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function sp_Callback(hObject, eventdata, handles)
% hObject     handle to sp (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of sp as text
%     str2double(get(hObject,'String')) returns contents of sp as a
double
global s_point
sp_1 = get(handles.sp,'String');
s_point = str2num(sp_1);
if s_point<0
    msgbox('NÃO É POSSÍVEL UTILIZAR NÚMEROS NEGATIVOS. DEFINA UM NOVO
NÚMERO. ');
    set(handles.sp,'String',num2str(0));
    s_point = 0;
end
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function sp_CreateFcn(hObject, eventdata, handles)
% hObject     handle to sp (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in mf.
function mf_Callback(hObject, eventdata, handles)
% hObject     handle to mf (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of mf
set(handles.mf,'Value',1);
set(handles.ma,'Value',0);
set(handles.ma,'Enable','off');
set(handles.degrau,'Enable','off');
guidata(hObject,handles)

% --- Executes on button press in ma.
function ma_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to ma (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of ma
set(handles.mf,'Value',0);
set(handles.ma,'Value',1);
set(handles.mf,'Enable','off');
set(handles.k,'Enable','off');
set(handles.ki,'Enable','off');
set(handles.kd,'Enable','off');
set(handles.sp,'Enable','off');
guidata(hObject,handles)

function degrau_Callback(hObject, eventdata, handles)
% hObject    handle to degrau (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of degrau as text
%        str2double(get(hObject,'String')) returns contents of degrau as a
double
global tensao
degrau_1 = get(handles.degrau,'String');
tensao = str2num(degrau_1);
if tensao<0
    msgbox('NÃO É POSSÍVEL UTILIZAR TENSÃO MENOR A 0V, DEFINA UM NOVO
VALOR. ');
    set(handles.degrau,'String',num2str(0));
    tensao = 0;
else
    if tensao>12
        msgbox('NÃO É POSSÍVEL UTILIZAR TENSÃO MAIOR A 12V, DEFINA UM NOVO
VALOR');
        set(handles.degrau,'String',num2str(0));
        tensao = 0;
    end
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function degrau_CreateFcn(hObject, eventdata, handles)
% hObject    handle to degrau (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');z
end

```