

## Capítulo II – Estratégias de Controle Inteligente Aplicadas a uma Planta Didática de Nível de Baixo Custo

Prof. MS.c. Bruno Leandro Galvão Costa<sup>7</sup>

### RESUMO

Este artigo ilustra a aplicação de abordagens de controle inteligente em uma planta didática de controle de nível construída com dispositivos de baixo custo, neste caso, utilizando componentes relacionados à plataforma eletrônica Arduino Uno. Detalhes construtivos deste recurso didático de ensino são comentados ao longo do trabalho. As estratégias de controle inteligente consideradas para a análise no sistema proposto baseiam-se nas redes neurais artificiais e lógica fuzzy, sendo seus desempenhos comparados com o de um controlador proporcional-integrativo clássico. Resultados de ensaios experimentais foram obtidos e são ilustrados no decorrer do trabalho visando evidenciar as vantagens, em termos de controle, de utilização das estratégias inteligentes em relação à convencional.

**Palavras-chave:** Arduino Uno. Controle PI. Lógica Fuzzy. Redes Neurais Artificiais. Sistema de Controle de Nível.

### INTELLIGENT CONTROL STRATEGIES APPLIED TO A LOW COST LEVEL CONTROL DIDACTIC PLANT

### ABSTRACT

This paper illustrates the application of intelligent control approaches in a didactic level control plant built with low-cost devices, in this case, using components related to the Arduino Uno electronic platform. Constructive details of this teaching didactic resource are discussed throughout the article. The intelligent control strategies considered for the analysis of the proposed system are based on artificial neural networks and fuzzy logic, and their performances are compared with that of a classical proportional-integrative controller. Results of experimental test were obtained and are illustrated in the course of the work in order to highlight the advantages, in terms of control, of using intelligent strategies in relation to the conventional one.

**Key-words:** Arduino Uno. PI Control. Fuzzy Logic. Artificial Neural Networks. Level Control System.

---

<sup>7</sup> Docente do programa de pós-graduação na Faculdade da Indústria SENAI Londrina. E-mail: [brunolgcosta@gmail.com](mailto:brunolgcosta@gmail.com)

## 1. INTRODUÇÃO

Atualmente, nos ambientes industriais, muitas estratégias de controle têm sido utilizadas, visando melhorar o desempenho dos processos envolvidos, bem como aliar questões relacionadas à maximização de produção e minimização de custos. Dentre estas estratégias de controle, os controladores PI/PID são certamente uma das melhores metodologias aplicadas ao controle de processos, principalmente por conta de sua simplicidade de implementação (KOSÁK, 2014).

O projeto de um controlador PI/PID para um dado sistema consiste na definição de alguns parâmetros que caracterizem a dinâmica do processo em questão. Inicialmente, um método de identificação deve ser empregado para a modelagem matemática do sistema de controle. Sendo assim, na literatura podem ser encontrados vários métodos de identificação tais como os de Broida (HAMITI, 1996), Ziegler/Nichols, Sundaresan / Krishnaswamy e de Smith (COELHO; COELHO, 2004).

Após este procedimento é necessário efetuar a sintonia do controlador, ou seja, ajustar seus respectivos ganhos: proporcional, integrativo e derivativo. Dessa forma os métodos de sintonia podem ser empregados, sendo baseados i) em curvas de resposta ao degrau (ASTROM; HÄGGLUND, 1995), ii) no lugar das raízes (OGATA, 2010) e iii) na resposta em frequência (ANGÉLICO; CAMPANHOL; SILVA, 2014).

Apesar da enorme aplicabilidade dos controladores PI/PID, em algumas situações a obtenção de modelos matemáticos de processos de controle pode ser bastante difícil, devido a inúmeros motivos dentre eles às suas características não-lineares. Devido a isso, muitas estratégias alternativas de controle, e que não exigem o conhecimento do modelo do sistema, têm surgido nas últimas décadas, podendo resultar em desempenhos superiores àqueles dos controladores PI/PID. Dentre elas podem ser destacadas àquelas baseadas em sistemas inteligentes (ou inteligência artificial), neste caso, sistemas fuzzy (GOMIDE; GUDWIN, 1994) e redes neurais artificiais (RNAs) (SILVA, 2016).

A lógica fuzzy, também conhecida como lógica nebulosa ou difusa, é uma metodologia computacional que busca desenvolver raciocínios muito próximos aos dos seres humanos, visando a solução de problemas cujas informações sejam imprecisas ou incertas. As RNAs, por sua vez, são algoritmos que se baseiam no comportamento do cérebro humano, o qual possui neurônios, elementos sensoriais

dispostos em uma vasta rede cuja função é a de propagação de impulsos nervosos. No ambiente computacional as RNAs apresentam capacidade de aprendizado e generalização, realizados por meio de informações que lhes são fornecidas.

Vários trabalhos ilustram a aplicação de sistemas inteligentes em processos de controle. Almeida (2014) descreve a aplicação de topologias de controladores fuzzy para um processo multivariável. Por sua vez, o artigo de Andrade, Agulhari, Goedel (2015) ilustra a implementação de um PI-neural para um sistema de controle de vazão. Já em Bertachi et al. (2013), controladores neurais foram desenvolvidos para o controle de uma malha multivariável de nível e temperatura. No trabalho de Freire Júnior, Vargas, Goedel (2013) um PI-neural foi projetado para o controle de um servomecanismo.

Este artigo busca aplicar estratégias de inteligência artificial em sistemas de controle, neste caso, como controladores baseados nos conceitos da lógica fuzzy e das RNAs, sendo analisadas em uma planta didática de controle de nível desenvolvida com equipamentos de baixo custo e controlada via plataforma Arduino Uno. Tal sistema de controle foi desenvolvido com o objetivo de poder auxiliar ambientes de ensino e aprendizagem acadêmicos, de modo a facilitar o entendimento sobre aspectos relacionados à área de controle. Neste contexto, controladores fuzzy e neural foram implementados e seus comportamentos foram analisados com o de um controlador PI convencional, sendo apresentados no decorrer do trabalho. Testes experimentais foram coletados e são mostrados de modo a evidenciar as principais contribuições em termos de controle de tais estratégias inteligentes.

## **2. FUNDAMENTAÇÃO TEÓRICA**

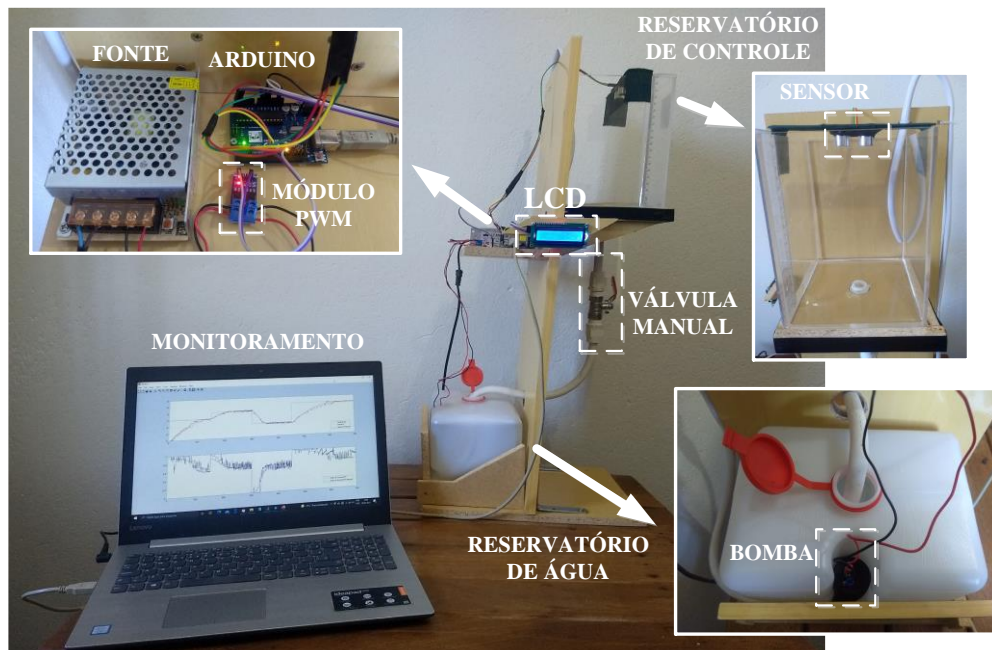
### **2.1 Planta didática de controle de nível**

A Figura 1 ilustra a planta didática de controle de nível desenvolvida neste artigo, constituída principalmente pelos seguintes componentes de fácil acesso:

1. Plataforma de desenvolvimento Arduino;
2. Sensor de nível ultrassônico HC-SR04;
3. Módulo PWM D4184 30A/36V 400W;
4. Fonte de tensão 5A/12V;

5. Display LCD 16x2 (com adaptador I2C);
6. Bomba de água 4A/12V;
7. Válvula manual; e
8. Reservatórios de água e de controle, acoplados com engates de PVC e mangueira para envio/recebimento de água.

Figura 1 – Planta didática de controle de nível



Fonte: Elaborado pelo autor (2021)

Esta planta utiliza a plataforma de desenvolvimento Arduino Uno para a implementação dos controladores que serão analisados ao longo do trabalho, a qual foi adotada no projeto devido ao fato de ser uma plataforma mundialmente difundida e popularizada, tanto nos ambientes acadêmicos quanto nos comerciais e industriais, com grande facilidade de utilização. Dentre as principais características do Arduino Uno podem ser destacadas (MCROBERTS, 2011):

1. Estrutura composta por um microprocessador Atmel AVR, microcontrolador ATmega328P e regulador de tensão linear de 5V (padrão de operação);
2. 6 pinos de entrada analógica (A0 a A5) (10 bits), 14 pinos de entradas/saídas digitais (0 a 13), sendo 6 destes pinos do tipo PWM (8 bits);
3. Corrente CC por pino de entrada/saída igual a 40mA;
4. Clock interno a cristal com 16 MHz;

5. Possibilidade de desenvolvimento de algoritmos baseados em linguagem C utilizando o IDE Arduino, ambiente gratuito e fornecido pelo fabricante;
6. Porta USB para comunicação de dados com computadores.

Para realizar as medições de nível no reservatório de controle, optou-se pelo módulo ultrassônico HC-SR04, um dispositivo de sensoriamento de baixo custo e de fácil utilização. Como dispositivo atuador do sistema de nível optou pela utilização de uma bomba de água 12V, acoplada em um reservatório de água (ambos utilizados em limpadores para-brisas de automóveis), sendo operada nos experimentos com uma tensão máxima de 6V para conservar sua vida útil. Devido ao seu consumo de corrente ser de aproximadamente 4A, optou-se pela utilização do módulo PWM D4184, também de baixo custo, que permite operações até 30A. Acoplado ao módulo PWM está uma fonte de tensão 5A/12V, necessária para a alimentação da bomba, de acordo com o nível PWM enviado pelo Arduino. Um pequeno display LCD é considerado para o monitoramento do sistema. Por fim, uma válvula manual é acoplada ao sistema de nível possibilitando possíveis distúrbios.

Vale a pena destacar que para o monitoramento em tempo real do sistema de controle de nível, optou-se por utilizar o software MATLAB, o qual possibilita a criação de algoritmos de controle bastante sofisticados. Neste caso, utiliza-se a comunicação serial como interface para a troca de dados entre a planta de controle e o software.

## **2.2 Abordagem para o projeto de controladores PI/PID**

### **2.2.1 Identificação de processos**

A identificação de um sistema consiste, basicamente, na determinação de um modelo matemático com base em informações do sistema a ser controlado (COELHO; COELHO, 2004). O objetivo principal desta etapa é obter uma função de transferência (FT) que caracterize da melhor forma possível a dinâmica de um determinado sistema de controle, de modo a auxiliar a etapa de sintonia do controlador a ser empregado, ou seja, ajuste de seus ganhos ou parâmetros.

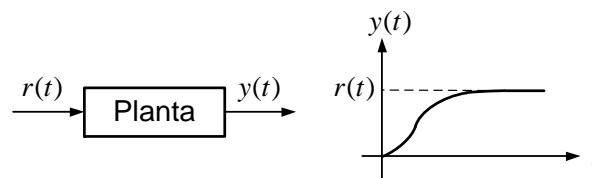
Essa função, em geral, é obtida no domínio do tempo, sendo representada com o auxílio da transformada de Laplace. Alguns processos industriais, tais como sistemas de controle de nível, temperatura, vazão ou pressão, por exemplo, muitas

vezes podem ser representados por FTs de primeira ou de segunda ordem (OGATA, 2010). Neste trabalho, a planta de controle de nível será modelada como um sistema de primeira ordem, cuja FT é dada pela equação (1):

$$(1)$$

onde  $K$  é o ganho em regime permanente,  $L$  o atraso de transporte e  $T$  a constante de tempo do sistema. Para a obtenção desta FT alguns métodos de identificação de sistemas baseados na resposta do processo ao degrau podem ser considerados. Tal resposta, como ilustrado na Figura 2 por  $y(t)$ , assume um formato em “S” o que facilita a identificação do processo por meio de tais métodos, que já fornecem equações características para a determinação dos parâmetros  $K$ ,  $L$  e  $T$  da equação (1).

Figura 2 – Resposta de um processo ao degrau unitário

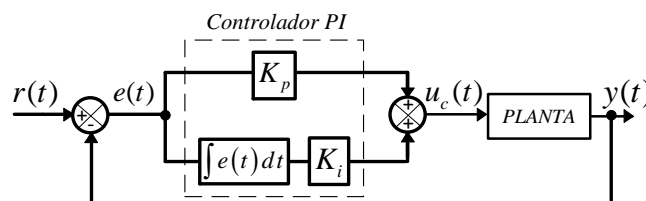


Fonte: Elaborado pelo autor (2021)

### 2.2.2 Sintonia de controladores PI/PID

Após a etapa de identificação do sistema, métodos de sintonia podem ser aplicados para projetar controladores PI/PID. Neste trabalho, considera-se o projeto de um controlador PI para a planta de nível, cuja estrutura é ilustrada na Figura 3, ação de controle  $u_c(t)$  e FT  $C(s)$  dadas, respectivamente, pelas equações (2) e (3):

Figura 3 – Estrutura do controlador PI



Fonte: Elaborado pelo autor (2021)

$$(2)$$

$$(3)$$

onde  $K_p$  e  $K_i$  são os ganhos proporcional e integrativo, respectivamente. Por sua vez, o erro atuante do sistema  $e(t)$  é definido como a diferença entre os sinais de referência  $r(t)$  e da saída do sistema  $y(t)$  (que neste trabalho é o nível de água). A escolha dos parâmetros  $K_p$  e  $K_i$  geralmente atende a requisitos de desempenho, tais como máximo sobressinal, tempos de subida e de acomodação, por exemplo. Uma das técnicas empregadas no ambiente industrial é a tentativa e erro (do inglês *trial-and-error*), baseada em chutes empíricos para os ganhos. No entanto, esta técnica apresenta problemas: necessidade de inúmeros testes, às vezes exaustivos, e gasto de tempo.

Para solucionar este tipo de problema, métodos de sintonia foram desenvolvidos para uma melhor determinação de parâmetros, tais como os métodos i) baseados na resposta ao degrau (COELHO; COELHO, 2004), ii) baseados na análise do lugar das raízes (OGATA, 2010) e iii) baseados na resposta em frequência, sendo este último considerado neste artigo para o ajuste dos ganhos do controlador PI, cuja metodologia de projeto é detalhada em Angélico, Campanhol, Silva (2014). Além do conhecimento da FT do sistema no qual deseja-se controlar, dois parâmetros de projeto devem ser definidos: i) a frequência de cruzamento de ganho ( $\omega_G$ ) e ii) a margem de fase desejada ( $\gamma$ ), estipulados pelo projetista (ou usuário).

### 2.2.3 Discretização do controlador

Para o desenvolvimento de sistemas de controle em simulações computacionais são utilizados modelos matemáticos baseados em equações diferenciais. Contudo, para a implementação física destes sistemas em microcontroladores, principalmente o controlador PI, devem ser consideradas equações de diferenças, dependentes do tempo de amostragem ( $T_s$ ), uma variável importante a ser considerada em implementações de hardware digital.

Tais equações são obtidas a partir de métodos de discretização, equações que relacionam o tempo contínuo ( $s$ ) com o tempo discreto ( $z$ ), sendo geralmente considerados os métodos de aproximação *Backward*, *Forward* e Trapezoidal (ou Tustin), sendo este último o considerado para o presente trabalho, onde a variável de Laplace é definida como  $s=(2/T_s)[z-1]/[z+1]$ . Substituindo esta relação na FT da equação (3), e aplicando os conceitos relacionados à transformada  $z$ , obtém-se a equação de diferenças para a ação de controle do PI, dada pela equação (4):

(4)

em que as variáveis  $k$  e  $k-1$  representam uma amostra atual e uma anterior no sistema em tempo discreto, respectivamente.

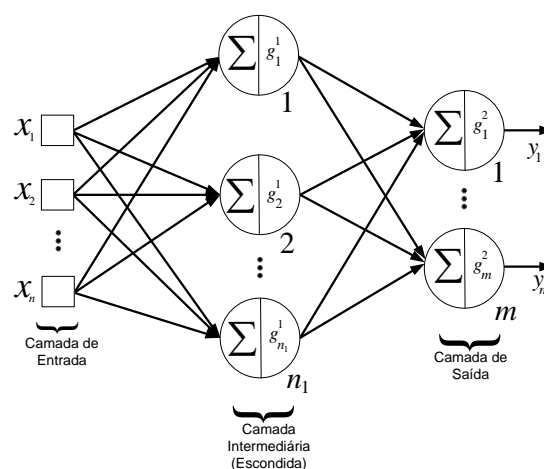
## 2.3 Abordagens baseadas em inteligência artificial

### 2.3.1 Redes neurais artificiais

As RNAs, nas últimas décadas, vêm se tornando uma abordagem amplamente estudada em diversas áreas do conhecimento, devido as suas potencialidades, que as tornam muito atrativas para inúmeras aplicações. Estas redes são modelos computacionais inspirados no sistema nervoso de seres vivos, que possuem capacidade de aquisição e manutenção de conhecimento/aprendizado (SILVA, 2016).

Uma das topologias mais empregadas, e que é considerada neste trabalho, é a Perceptron Multicamadas (PMC), pertencente à arquitetura *feedforward* de camadas múltiplas, muito aplicada em problemas que envolvem aproximação de funções, controle de processos, classificação de padrões, por exemplo (SILVA, 2016). Sua estrutura é caracterizada, por pelo menos uma camada intermediária (ou escondida) de  $n_1$  neurônios, entre as camadas de entrada (com  $x_n$  sinais de entrada) e de saída (com  $m$  neurônios), como ilustra a Figura 4.

Figura 4 – Estrutura de uma rede PMC



Fonte: Elaborado pelo autor (2021)

Basicamente, cada neurônio artificial da rede encontra-se interconectado com suas respectivas entradas por meio dos pesos sinápticos (representados pelas flechas na Figura 4), que são vetores ou matrizes numéricas ajustados por meio dos algoritmos de aprendizagem. Cada neurônio, por sua vez, possui a capacidade de avaliar informações vindas de cada entrada, que são ponderadas a partir dos pesos, e geram uma saída proporcional a estes “estímulos”, com base em funções de ativação (representadas pela variável  $g$  na Figura 4).

O tipo de aprendizagem desta rede é o supervisionado, caracterizado pela disponibilidade de amostras de entrada com suas respectivas saídas. Logo, é desejável que o algoritmo busque garantir que as saídas da rede sejam praticamente idênticas às saídas desejadas. Para a rede PMC o algoritmo geralmente empregado é o *Backpropagation* (BP), cujos detalhes podem ser encontrados em Silva (2016).

### 2.3.2 Lógica fuzzy

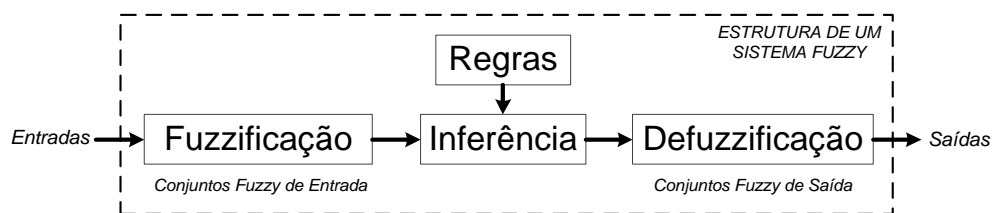
Os conceitos relacionados à lógica fuzzy surgiram da ideia de reprodução do raciocínio de seres humanos, capazes de lidar com problema complexos, a partir de informações imprecisas. Zadeh foi o idealizador dessa lógica, que buscava uma ferramenta matemática para o tratamento de informações vagas e que não fossem extremas (0 ou 1) como ocorre na lógica clássica (GOMIDE; GUDWIN, 1994).

Com base nessa estrutura lógica torna-se possível o desenvolvimento de diversos tipos de sistemas, denominados “sistemas fuzzy”. A Figura 5 ilustra a estrutura de um sistema fuzzy, o qual pode ser dividido em três etapas principais:

1. *Fuzzificação*: consiste no mapeamento das entradas (provenientes de uma dada aplicação) em conjuntos fuzzy (ou termos linguísticos), definidos para as variáveis linguísticas;
2. *Inferência*: processo que irá relacionar as entradas já condicionadas aos conjuntos fuzzy com as saídas do sistema. Nesta etapa consideram-se as regras linguísticas, conjunto de proposições fuzzy combinadas por meio de operadores lógicos, que irão associar as variáveis linguísticas de entrada com as de saída (definidas com base no conhecimento do usuário);
3. *Defuzzificação*: etapa final que irá transformar as ações fuzzy, definidas na inferência, em ações compatíveis com a aplicação, ou seja, é escolhida a

melhor ação para a decisão tomada no processo de inferência, baseada na utilização de métodos matemáticos.

Figura 5 – Estrutura básica de um sistema fuzzy



Fonte: Elaborado pelo autor (2021)

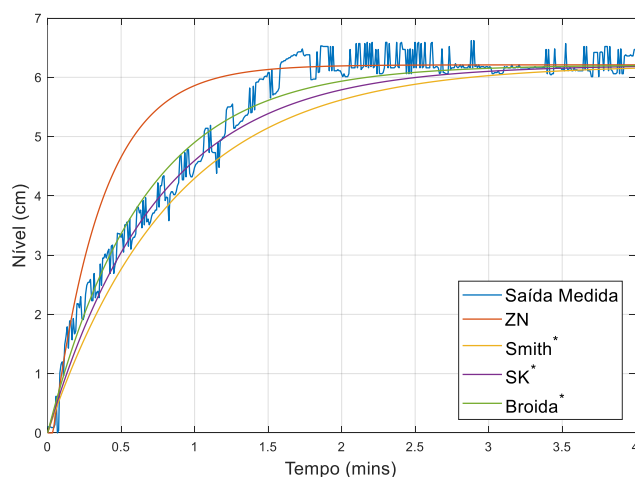
### 3. METODOLOGIA

#### 3.1 Projeto do controlador PI convencional

A primeira etapa a ser considerada para o projeto do controlador PI é a caracterização do modelo matemático do sistema de controle de nível, ou seja, sua  $G(s)$ . Sendo assim, neste artigo, para a obtenção dessa função, considerou-se uma análise baseada na comparação das curvas de resposta ao degrau aproximadas fornecidas pelos métodos de identificação de i) Ziegler/Nichols (ZN), ii) Smith, iii) Sundaresan/Krishnaswamy (SK) e iv) Broida, sendo ilustradas na Figura 6.

Durante esta análise foi percebido que, para os métodos de Smith, SK e Broida, o comportamento de resposta ao degrau do sistema ficou mais próximo do sistema real quando o atraso de transporte ( $L$ ) não foi considerado nas FTs. Na Figura 6 o símbolo (\*) na legenda indica justamente esta constatação, ou seja, respostas ao degrau para as funções  $G(s)$  resultantes sem considerar o atraso de transporte.

Figura 6 – Comparação das respostas ao degrau obtidas para cada método



Fonte: Elaborado pelo autor (2021)

Logo, para a escolha do melhor modelo da planta, considerou-se o índice “Erro Relativo Médio” (ERM), cuja equação é dada abaixo:

$$(5)$$

em que  $n$  é o número total de amostras, sendo  $e(i) = y(i) - \hat{y}(i)$ , onde  $y(i)$  representa a saída real medida enquanto  $\hat{y}(i)$  representa a saída estimada pelo modelo para uma dada  $i$ -ésima amostra. A Tabela 1 mostra os valores do ERM obtidos para cada um dos modelos resultantes.

Tabela 1 – Erro Relativo Médio (ERM) dos modelos aproximados

Modelo	ZN	ZN*	Smith	Smith*	SK	SK*	Broida	Broida*
<i>ERM</i>	0,4536	0,4995	0,4509	0,3854	0,3705	0,2729	0,2550	0,2331

Fonte: Elaborado pelo autor (2021)

Com base nos valores obtidos, é possível concluir que o menor ERM corresponde ao modelo dado pelo método de Broida desconsiderando o atraso de transporte, cuja FT aproximada para a planta de controle de nível é dada por:

$$(6)$$

Uma vez obtida a  $G(s)$  do sistema de controle é possível desenvolver o projeto do controlador. Neste artigo, optou-se pela utilização do método de resposta em frequência para o projeto dos ganhos do PI, cujos parâmetros são a frequência de cruzamento de ganho ( $\omega_G$ ), dada em rad/s, e a margem de fase ( $\gamma$ ), dada em graus.

Sendo assim, cinco controladores PI foram projetados para o sistema em questão, sendo cada um deles testado em protótipo, a fim de analisar a melhor configuração. Vale destacar que para a discretização dos PIs considera-se o método Trapezoidal, sendo adotada uma taxa de amostragem  $T_s = 1$  ms. A Tabela 2 descreve os ganhos  $K_p$  e  $K_i$  obtidos em cada um dos projetos, assim como seus respectivos parâmetros  $\omega_G$  e  $\gamma$ . Para a escolha do melhor PI, considerou-se os índices de desempenho  $IAE$  e  $IAU$ , descritos em (7) e (8), nos quais são realizados os cálculos das integrais dos módulos do erro  $e(t)$  e da ação de controle  $u_c(t)$ , respectivamente. Os valores de  $IAE$  e  $IAU$  para cada um dos controladores também são mostrados na Tabela 2. Por sua vez, a Figura 7 ilustra o comportamento da variável de saída e da  $u_c(t)$  do sistema para cada controlador (vale destacar que tais curvas foram obtidas considerando um cenário de degraus comentado em detalhes na seção 4).

(7)

(8)

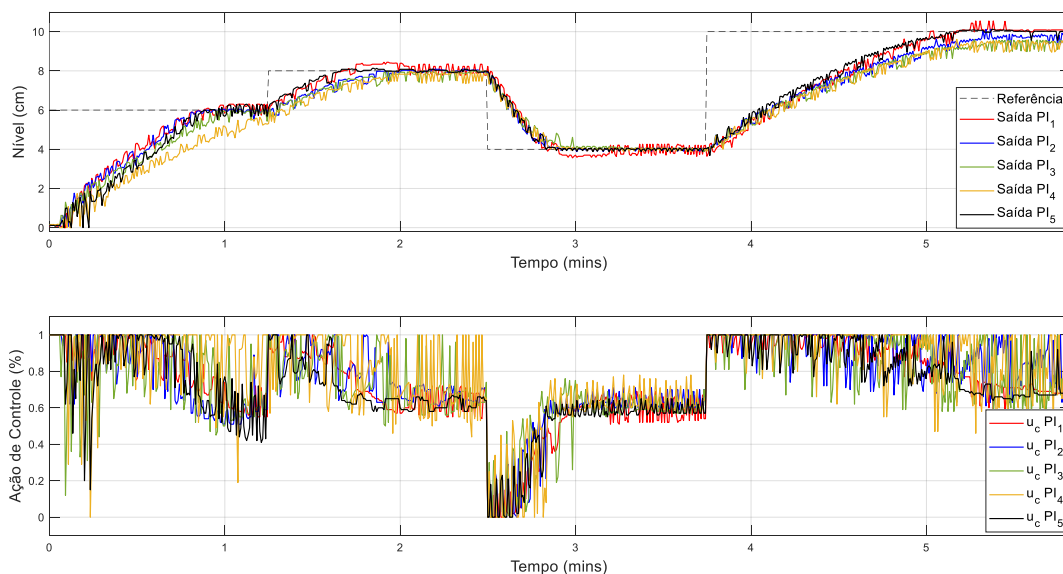
Tabela 2 – Resultados obtidos para os controladores PI

Controlador	$\omega_G$ (rad/s)	$\gamma$ (°)	$K_p$	$K_i$	IAE	IAU
PI <sub>1</sub>	5	50	0,29237	2,27771	7,7842	4,3348
PI <sub>2</sub>	10	50	0,68826	7,87726	7,9292	4,4876
PI <sub>3</sub>	10	60	0,81459	6,56244	8,8910	4,3860
PI <sub>4</sub>	15	50	1,08414	16,79865	9,9303	4,6446
PI <sub>5</sub>	8	50	0,52990	5,23882	7,4583	4,1903

Fonte: Elaborado pelo autor (2021)

É possível observar nesta tabela que o controlador “PI<sub>5</sub>” é o que resultou em menores valores de IAE e IAU, consequência de um melhor desempenho dentre os controladores analisados. Portanto, este será o controlador PI considerado para o desenvolvimento do projeto do PI-neural, assim como para a análise com os controladores inteligentes.

Figura 7 – Comparação entre os controladores PI projetados



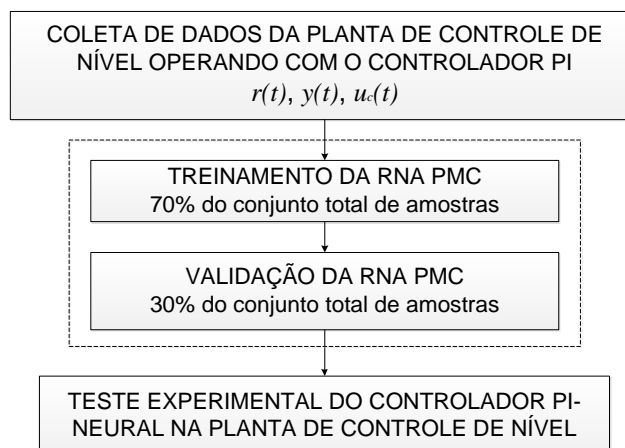
Fonte: Elaborado pelo autor (2021)

### 3.2 Projeto do controlador PI-neural

A Figura 8 ilustra a metodologia considerada para a implementação do controlador PI-neural neste artigo, composta por 4 etapas principais:

1. *Coleta dos Dados da Planta:* obtenção do conjunto total de amostras que serão consideradas para o treinamento e validação da RNA, obtidas a partir do sistema de controle operando com o controlador PI (escolhido na seção 3.1). Neste trabalho, considerou-se amostras da referência ( $r(t)$ ), da saída ( $y(t)$ ) e da ação de controle ( $u_c(t)$ ) do sistema;
2. *Treinamento da RNA PMC:* etapa de aprendizagem da rede no qual 70% das amostras são consideradas;
3. *Validação da RNA PMC:* finalizado o treinamento, 30% do conjunto total de amostras, que não foram consideradas na etapa de aprendizagem, são utilizadas para a etapa de validação, visando verificar se a RNA retorna valores de saída próximos aos valores desejados;
4. *Teste do Controlador PI-neural na Planta:* uma vez encerrada a etapa de validação, os pesos sinápticos ajustados podem ser utilizados para a implementação em hardware do PI-neural visando avaliá-lo;

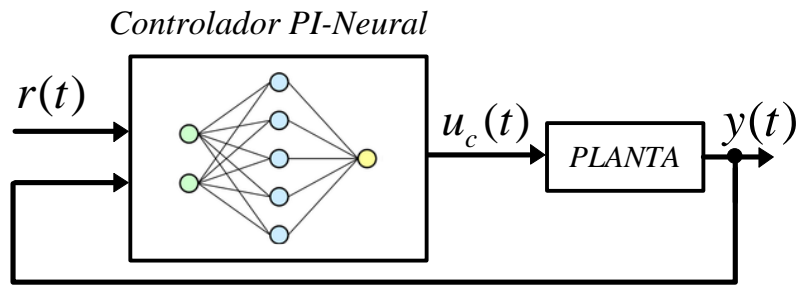
Figura 8 – Metodologia para a implementação do controlador PI-neural



Fonte: Elaborado pelo autor (2021)

Vale a pena destacar que a coleta dos dados, assim como as etapas de treinamento e validação foram feitas utilizando algoritmos desenvolvidos no MATLAB via linhas de código e sem utilização de *toolboxes*. O controlador PI-neural da planta de nível implementado neste trabalho é constituído por três camadas, conforme mostrado na Figura 9: i) *camada de entrada*, com dois neurônios correspondentes aos valores de referência ( $r(t)$ ) e de saída do sistema ( $y(t)$ ), ii) *camada escondida*, composta por cinco neurônios e iii) *camada de saída*, com um único neurônio, referente à ação de controle ( $u_c(t)$ ). Os demais parâmetros considerados para o desenvolvimento do controlador são apresentados na Tabela 3.

Figura 9 – Estrutura do controlador PI-neural



Fonte: Elaborado pelo autor (2021)

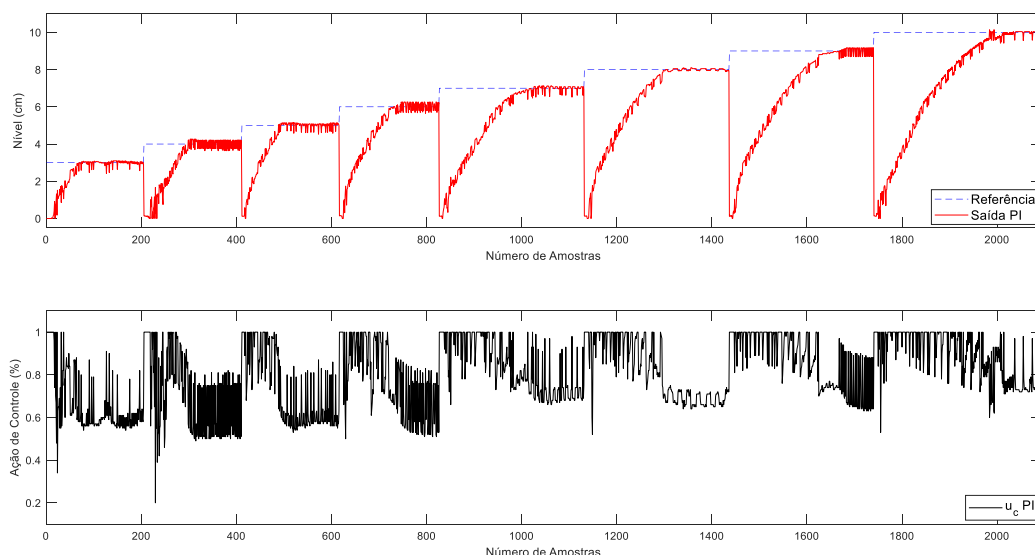
A Figura 10 ilustra o conjunto coletado de amostras composto por valores de referência, saída e ação de controle do protótipo do sistema de nível operando com o controlador PI projetado na seção 3.1. Tais amostras foram obtidas variando-se o valor de referência de 3 cm a 10 cm, com intervalos de 1 cm.

Tabela 3 – Parâmetros da RNA implementada

Neurônios na Camada de Entrada		$x_n = 2$ (referência e saída)	
Neurônios na Camada Escondida		$n_l = 5$	
Neurônios na Camada de Saída		$m = 1$ (ação de controle)	
Função de Ativação dos Neurônios		Tangente Hiperbólica	
Taxa de Aprendizagem do Algoritmo BP		0,001	
Erro requerido		$1.10^{-8}$	
Número Total de Amostras	Treinamento (70%)	2094	1466
	Validação (30%)		628
Época de Convergência		8568	

Fonte: Elaborado pelo autor (2021)

Figura 10 – Conjunto de amostras coletadas da planta de controle de nível



Fonte: Elaborado pelo autor (2021)

As Tabelas 4 e 5 descrevem os valores dos pesos sinápticos dos neurônios das camadas escondida e de saída obtidos ao final da etapa de treinamento da RNA.

A Figura 11 ilustra o resultado do processo de validação da RNA, no qual é possível observar que a rede é capaz de generalizar de forma correta e aproximada ao comportamento do sistema.

Tabela 4 – Pesos sinápticos dos neurônios da camada escondida

	Limiar ( $\theta$ )	Referência ( $r(t)$ )	Saída ( $y(t)$ )
<b>N1</b>	0,6864	0,1139	1,0349
<b>N2</b>	0,6994	0,9739	-0,0011
<b>N3</b>	0,8202	0,5187	0,4461
<b>N4</b>	0,9360	0,4521	0,6384
<b>N5</b>	-0,9479	3,2381	-3,8754

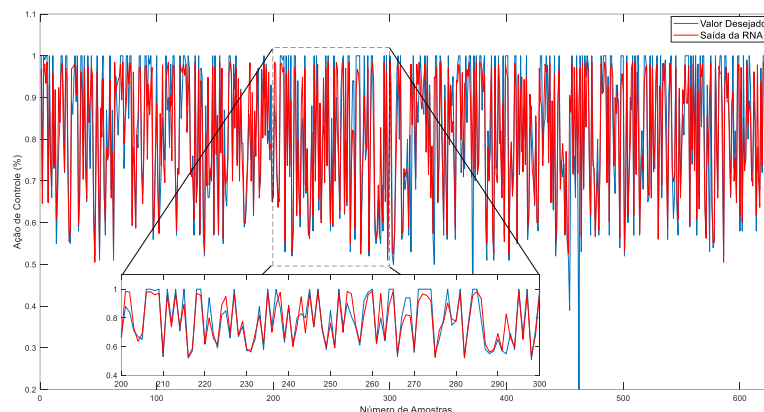
Fonte: Elaborado pelo autor (2021)

Tabela 5 – Pesos sinápticos do neurônio da camada de saída

	Limiar ( $\theta$ )	N1	N2	N3	N4	N5
<b>N1</b>	0,2612	0,1581	0,7334	0,6548	0,2102	2,6793

Fonte: Elaborado pelo autor (2021)

Figura 11 – Resultado do processo de validação da RNA PMC



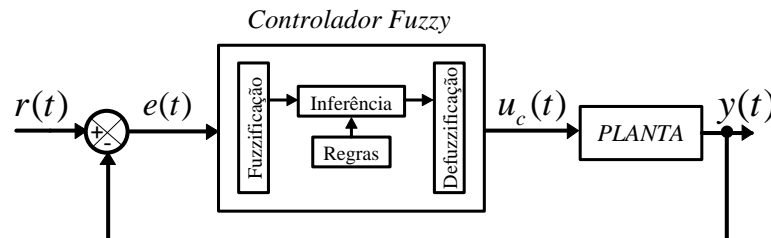
Fonte: Elaborado pelo autor (2021)

### 3.3 Projeto do controlador fuzzy

A Figura 12 ilustra a estrutura do controlador fuzzy para o controle da planta didática de nível. Como é possível notar, as variáveis linguísticas de entrada e de saída do controlador são o erro ( $e(t)$ ) e a ação de controle ( $u_c(t)$ ) do processo, respectivamente. Esse controlador é basicamente um sistema fuzzy convencional com os estágios de i) fuzzificação, etapa que mapeia o sinal do erro nos conjuntos fuzzy, ii) inferência, etapa caracterizada pelo processamento lógico do sistema, realizando a invocação das regras linguísticas e geração dos resultados, e iii) defuzzificação, conversão do resultado do processo de inferência em uma ação de

controle a ser enviada para a planta. Vale a pena destacar que para este trabalho considerou-se o controlador fuzzy com estrutura do tipo Mamdani, sendo implementado utilizando o *Fuzzy Logic Toolbox* do software MATLAB.

Figura 12 – Estrutura do controlador Fuzzy

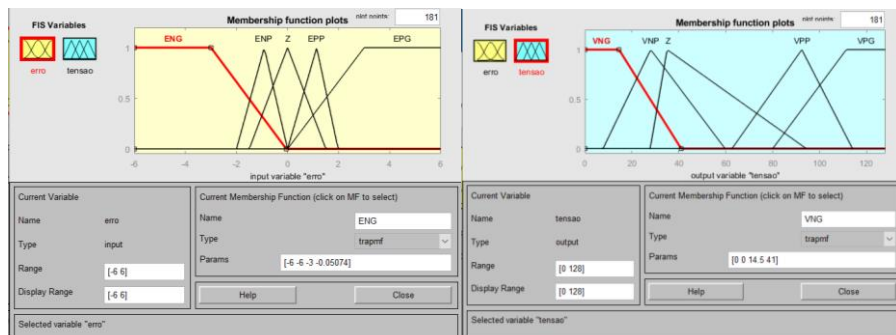


Fonte: Elaborado pelo autor (2021)

A Figura 13 ilustra as configurações definidas para as variáveis linguísticas de entrada, denominada “erro”, e de saída, denominada “tensão”. Para cada uma delas foram definidos cinco termos linguísticos de mapeamento,  $T(erro)$  e  $T(tensão)$ , e os seguintes universos de discurso,  $X(erro)$  e  $X(tensão)$ , sendo descritos a seguir:

1. *Variável “erro”*: ENG (erro negativo grande), ENP (erro negativo pequeno), Z (zero), EPP (erro positivo pequeno) e EPG (erro positivo grande):  $T(erro) = \{ENG, ENP, Z, EPP, EPG\}$ ,  $X(erro) = -6$  a  $6$ ;
2. *Variável “tensão”*: VNG (tensão negativa grande), VNP (tensão negativa pequena), Z (zero), VPP (tensão positiva pequena) e VPG (tensão positiva grande):  $T(tensão) = \{VNG, VNP, Z, VPP, VPG\}$ ,  $X(tensão) = 0$  a  $128$ ;

Figura 13 – Variáveis linguísticas de entrada (à esquerda) e de saída (à direita)



Fonte: Elaborado pelo autor (2021)

A Tabela 6 descreve os métodos considerados para os conectivos lógicos e operadores do controlador fuzzy. Na Figura 14 são descritas as regras linguísticas para a inferência, definidas com base em aspectos fundamentais de controle:

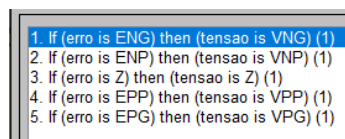
1. Se o erro de controle (positivo/negativo) é grande (ou seja, valor de saída muito distante do sinal de referência), muita ação de controle será necessária (ou seja, a tensão enviada para a bomba deverá ser alta);
2. Se o erro de controle (positivo/negativo) é pequeno (ou seja, valor de saída muito próximo do sinal de referência), pouca ação de controle será necessária (ou seja, a tensão enviada para a bomba deverá ser baixa);
3. Por fim, caso o erro de controle seja zero (indicando que o valor de saída é idêntico ao sinal de referência), nenhuma ação deverá ser realizada;

Tabela 6 – Parâmetros para o controlador fuzzy implementado

Parâmetros	Método
Norma-t e operador de implicação	Min
Conorma-t e operador de agregação	Max
Método de Defuzzificação	Centróide

Fonte: Elaborado pelo autor (2021)

Figura 14 – Regras linguísticas adotadas para o controlador fuzzy



```

1. If (erro is ENG) then (tensao is VNG) (1)
2. If (erro is ENP) then (tensao is VNP) (1)
3. If (erro is Z) then (tensao is Z) (1)
4. If (erro is EPP) then (tensao is VPP) (1)
5. If (erro is EPG) then (tensao is VPG) (1)

```

Fonte: Elaborado pelo autor (2021)

Como mencionado anteriormente, toda a estrutura do controlador fuzzy foi implementada considerando o *toolbox* do MATLAB. Portanto, para a integração deste com a planta de controle de nível, considerou-se um algoritmo de controle baseado em comunicação serial entre Arduino e MATLAB. Neste algoritmo, os valores do erro de controle, processados no Arduino, são enviados via serial para o MATLAB, sendo interpretados pelo controlador fuzzy desenvolvido, que gera sinais de atuação enviados novamente via serial para o Arduino, continuando assim a execução do algoritmo de controle no sistema microprocessado.

#### 4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

A seguir são ilustrados os resultados experimentais obtidos para a planta de controle de nível operando com cada um dos controladores projetados, segundo procedimentos apontados nas seções anteriores, neste caso, com o: i) controlador PI, ii) controlador PI-neural e iii) controlador fuzzy. Basicamente, para realizar o estudo comparativo entre os controladores, vários degraus de referência foram considerados, sendo adotado o seguinte cenário experimental: i)  $0 < t < 1,25$  mins: referência de

nível fixa em 6 cm; ii)  $1,25 < t < 2,5$  mins: referência de nível fixa em 8 cm; iii)  $2,5 < t < 3,75$  mins: referência de nível fixa em 4 cm; iv)  $3,75 < t < 5,8$  mins: referência de nível fixa em 10 cm;

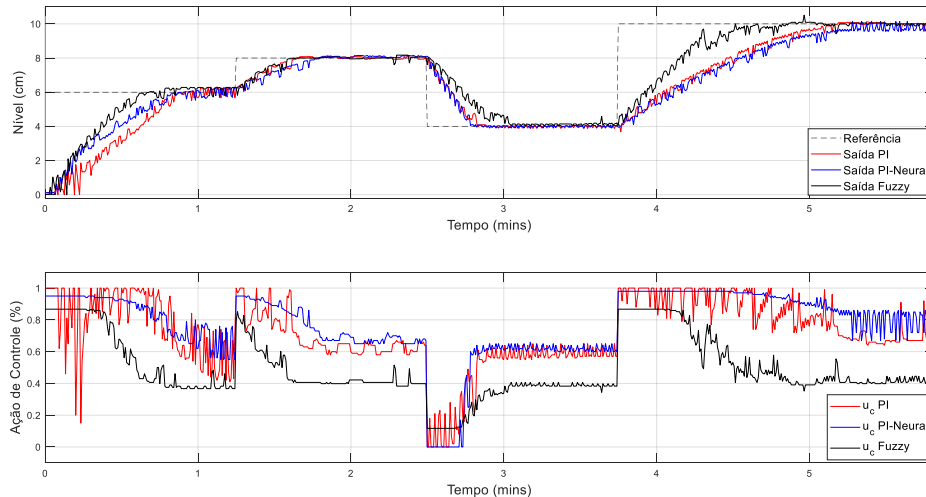
A Figura 15 ilustra os comportamentos do sistema de controle operando considerando os três controladores em estudo. Além disso, para avaliar o desempenho dos controladores, novamente optou-se pela utilização dos índices *IAE* e *IAU*, definidos em (7) e (8), respectivamente. Os valores obtidos considerando estes índices para os controladores projetados são descritos na Tabela 7.

É possível observar nos resultados experimentais, em um primeiro momento, que os três controladores respondem de maneira adequada e efetiva aos degraus de referência impostos, com uma rápida estabilização e sem grandes sobressinais. Analisando minuciosamente o comportamento dos sinais de saída é possível notar que as respostas do sistema operando com os controladores PI e PI-neural são muito similares, com o PI operando mais rápido em quase todos os períodos, exceto entre  $0 < t < 1$  min no qual o PI-neural foi mais ágil. Contudo, o sistema operando com o controlador fuzzy obteve, principalmente nos momentos onde houveram degraus de subida, uma resposta mais rápida que os controladores PI e PI-neural.

Além disso, a partir dos resultados obtidos para a ação de controle, é possível notar uma grande característica relacionada aos controladores inteligentes que, neste caso, é o fato de reduzir consideravelmente oscilações de tensão em praticamente todo o cenário de teste, com destaque novamente para o sistema operando com o controlador fuzzy, cujo comportamento da ação de controle é bem menos oscilatório, o que significa menos esforços na atuação da bomba de água.

Por sua vez, os valores retornados pelos índices de desempenho, descritos na Tabela 7, comprovam uma melhor eficiência de controle do sistema operando com o controlador fuzzy, visto que ambos *IAE* (relacionado ao erro de controle) e *IAU* (relacionado à ação de controle) resultaram em menores valores quando comparados aos controladores PI e PI-neural.

Figura 15 – Comportamentos da planta de controle de nível com os controladores projetados



Fonte: Elaborado pelo autor (2021)

Tabela 7 – Comparação de desempenho entre os controladores projetados

<b>Controlador</b>	<b>IAE</b>	<b>IAU</b>
PI	7,4583	4,1903
PI-Neural	7,4373	4,4860
Fuzzy	5,4566	2,8586

Fonte: Elaborado pelo autor (2021)

Um outro aspecto a ser ressaltado com o presente trabalho é com relação aos custos de implementação experimental da planta de controle de nível, os quais estão descritos na Tabela 8. É possível observar que o gasto total do protótipo foi de 435,31 reais, valor consideravelmente baixo quando comparado aos kits disponíveis no mercado, justamente pelo fato de ser composto por dispositivos de baixo custo, além de serem bastante acessíveis.

Tabela 8 – Custos da planta didática de controle

<b>Componentes</b>	<b>Valor</b>
Arduino Uno	R\$ 70,20
Sensor Ultrassônico HC-SR04	R\$ 16,90
Módulo PWM D4184 30A/36V	R\$ 10,90
Fonte Chaveada 60W 12V 5A	R\$ 49,21
Bomba de Água 12 V	R\$ 35,00
Display LCD 16x2 + Módulo I2C	R\$ 22,90
Tanque Acrílico	R\$ 70,00
Reservatório de Água	R\$ 64,90
Cabos 22AWG + Jumpers	R\$ 5,30
Engates Flexíveis de PVC + Mangueira	R\$ 15,00
Válvula Esfera 1/2" Manual	R\$ 15,00
Réguas Plásticas	R\$ 10,00

Placas de Madeira em MDF	R\$ 40,00
Suportes Metálicos + Parafusos	R\$ 10,00
<b>TOTAL</b>	<b>R\$ 435,31</b>

Fonte: Elaborado pelo autor (2021)

## 5. CONSIDERAÇÕES FINAIS

Este trabalho descreveu uma aplicação de controladores inteligentes baseados em lógica fuzzy e redes neurais artificiais em uma planta didática de controle de nível de baixo custo operando com a plataforma Arduino Uno. Todos os procedimentos de projeto de cada controlador foram comentados. Para o projeto do controlador PI foi realizada a modelagem matemática da planta, a partir do método de identificação de Broida, seguido da sintonia pelo método de resposta em frequência. Já o controlador PI-neural, foi projetado considerando amostras compostas por vários pontos de operação do sistema operando com o controlador PI projetado. Por sua vez, o controlador fuzzy também foi implementado segundo aspectos fundamentais relacionados à teoria de controle. Em seguida, os desempenhos dos três controladores foram comparados entre si, sendo notadas vantagens de operação dos controladores inteligentes principalmente com relação ao nível de oscilação de tensão no atuador do sistema. Após uma análise comparativa baseada em observação dos comportamentos dos sinais de saída e da ação de controle do sistema, assim como nos valores numéricos obtidos por índices de desempenho, pode-se notar que os controladores efetuaram um adequado controle para o sistema, com destaque para o controlador fuzzy, que obteve um desempenho superior.

## REFERÊNCIAS

ALMEIDA, J. P. L. S. **Técnicas fuzzy aplicadas ao controle descentralizado**. Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. Cornélio Procópio, 2014.

ANDRADE, L. H. S.; AGULHARI, C. M.; GOEDEL, A. Controle PI-neural aplicado a um sistema de vazão. In. XII Simpósio Brasileiro De Automação Inteligente, 2015. UFRN. **Anais...** Natal, 2015. p-468-473.

ANGÉLICO, B. A.; CAMPANHOL, L. B. G.; SILVA, S. A. O. Proportional-integral/proportional-integral-derivative tuning procedure of a single-phase shunt active power filter using Bode diagram. **IET Power Electronics**. v.7, n.10, p.2647–2659, ago. 2014.

ASTROM, K. J.; HÄGGLUND, T. **PID controllers: theory, design, and tuning**. 2. ed. Research Triangle Park: Instrument Society of America, 1995.

BERTACHI, A. H.; SILVA, L. R. B.; SUMAR, R. R.; ANGÉLICO, B. A.; GOEDEL, A. Controle de um processo multivariável em uma planta didática industrial utilizando redes neurais. In. XI Simpósio Brasileiro De Automação Inteligente, UFC. **Anais...** Fortaleza, 2013. p-1-6.

COELHO, A. A. R.; COELHO, L. S. **Identificação de sistemas dinâmicos lineares**. Florianópolis: UFSC, 2004.

FREIRE JÚNIOR, V. A.; VARGAS, A. N.; GOEDEL, A. Uma abordagem PI-neural aplicado ao controle de um servo-mecanismo. In. XI SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (SBAI), 2013. UFC. **Anais...** Fortaleza, 2013. p-1-6.

GOMIDE, F. A. C.; GUDWIN, R. R. Modelagem, Controle, Sistemas e Lógica Fuzzy. **SBA Controle & Automação**. v.4, n.3, p. 97-115, 1994.

HAMITI, K. Position control of a pneumatic actuator under the influence of stiction. **Control Engineering Practice**. v.4, n.8, p.1079–1088, 1996.

MCCROBERTS, M. **Arduino básico**. São Paulo: Novatec Editora, 2011.

OGATA, K. **Engenharia de controle moderno**. 5. ed. São Paulo: Pearson Prentice Hall, 2010.

SILVA, I. N. **Redes neurais artificiais: para engenharia e ciências aplicadas**. São Paulo: Artliber, 2016.

KOZÁK, S. State-of-the-art in control engineering. **Journal of Electrical Systems and Information Technology**. v.1, n.1, p.1-9, 2014.