

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

VINÍCIUS BALDESSAR

PROJETO DE UM CONTROLADOR FUZZY APLICADO PARA O CONTROLE DE
NÍVEL DE TANQUE

Joinville
2022

VINÍCIUS BALDESSAR

PROJETO DE UM CONTROLADOR FUZZY APLICADO PARA O CONTROLE DE
NÍVEL DE TANQUE

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica no curso de Engenharia Mecatrônica, da Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientador: Prof. Dr. Benjamin Grando Moreira

Joinville
2022

RESUMO

O controle de nível de tanque é necessário para determinados processos na indústria e para solucionar esta necessidade são utilizados controladores nos processos. Este trabalho apresenta o estudo e aplicação da lógica Fuzzy para criação de controladores Fuzzy capazes de controlar sistemas de nível com comportamento não linear. O sistema a ser controlado é um tanque de água simulado que apresenta uma vazão de entrada e uma de saída. A função do controlador é manter próximo ao nível de líquido no tanque de acordo com a referência estipulada. A construção do controlador Fuzzy e simulações do sistema são feitas com ferramentas *open source*. O desempenho do primeiro controlador Fuzzy implementado é comparado com a resposta gerada por controladores clássicos proporcionais através de simulações. Os resultados obtidos e comparações realizadas com trabalhos similares mostram a aplicabilidade de controladores Fuzzy para o controle de nível. As soluções de controladores encontradas seguem o sinal de referência por meio de uma implementação simples, sendo analisado também a resposta dos controladores quando variada a taxa de amostragem do sistema.

Palavras-chave: Controlador Fuzzy. Mamdani. Controle de Nível. Simulação.

ABSTRACT

Tank level control is necessary for certain processes in the industry and to solve this need controllers are used in the processes. This work presents the study and application of Fuzzy logic to create Fuzzy controllers capable of controlling level systems with non-linear behavior. The system to be controlled is a simulated water tank that has an inlet and an outlet flow. The function of the controller is to keep the liquid level in the tank close to the stipulated reference. The construction of the Fuzzy controller and system simulations are done with *open source* tools. The performance of the first Fuzzy controller implemented is compared with the response generated by classic controllers through simulations. The results obtained and comparisons made with similar works show the applicability of Fuzzy controllers for level control. The controller solutions found follow the reference signal through a simple implementation, and the response of the controllers is also analyzed when the system sampling rate is varied.

Keywords: Fuzzy Controller. Mamdani. Level Control. Simulation.

LISTA DE FIGURAS

Figura 1 – Sistema de Inferência Fuzzy	14
Figura 2 – Funções de pertinência	15
Figura 3 – Antecedentes	16
Figura 4 – Pressão nos freios	16
Figura 5 – Regras Fuzzy para energia cinética	19
Figura 6 – Saída <i>crisp</i> do sistema Fuzzy	19
Figura 7 – Tanques acoplados	21
Figura 8 – Conjuntos do controlador Fuzzy com algoritmos genéticos	22
Figura 9 – Variação da abertura da válvula para os controladores implementados	23
Figura 10 – Modelos de tanque usados por Vieira (2017)	24
Figura 11 – Funções de pertinência da variável de entrada referente ao erro . .	24
Figura 12 – Resposta experimental do controlador PI no tanque trapezoidal . . .	26
Figura 13 – Resposta experimental do controlador Fuzzy no tanque trapezoidal	27
Figura 14 – Resposta do controlador Fuzzy às entradas aplicadas no código 2. .	30
Figura 15 – Esquemático do sistema a ser controlado	33
Figura 16 – Controle Fuzzy	34
Figura 17 – Resultado da primeira implementação	37
Figura 18 – Controlador Fuzzy de comportamento gradual	38
Figura 19 – Controlador Fuzzy de comportamento determinístico	39
Figura 20 – Resposta do controlador proporcional	40
Figura 21 – Esquemático do novo modelo de tanque	41
Figura 22 – Controle utilizando apenas o erro atual do sistema	43
Figura 23 – Funções Antecedentes	44
Figura 24 – Função consequente para a vazão no sistema	44
Figura 25 – Resposta controlador com 2 funções antecedentes	45
Figura 26 – Resposta com baixa taxa de amostragem	47

LISTA DE CÓDIGOS

Código 1 – Implementação da pressão do pedal utilizando a Scikit-fuzzy	29
Código 2 – <i>Input</i> e <i>output</i> de valores <i>crisp</i> no controlador Fuzzy	30
Código 3 – Implementação do controle Fuzzy utilizando a Scikit-fuzzy	34
Código 4 – Implementação do controle Fuzzy	35

LISTA DE TABELAS

Tabela 1 – Regras Fuzzy	17
Tabela 2 – Regras Fuzzy para energia cinética	18
Tabela 3 – Regras para o controlador Fuzzy	45
Tabela 4 – Comparação entre controladores Fuzzy apresentados	48

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivo	10
1.1.1	Objetivo Geral	10
1.1.2	Objetivos Específicos	10
1.2	Estrutura do trabalho	10
2	INTRODUÇÃO À SISTEMAS DE CONTROLE	12
2.1	Sistemas de malha aberta e malha fechada	12
3	LÓGICA FUZZY	13
3.1	Fuzzificação	14
3.1.1	Variáveis Linguísticas	14
3.1.2	Funções de pertinência	15
3.2	Base de regras Fuzzy	16
3.3	Inferência Fuzzy	17
3.3.1	Método Mamdani	18
3.4	Defuzzificação	20
3.4.1	Método do centroide	20
4	TRABALHOS SIMILARES	21
4.1	Estrutura do tanque	21
5	FERRAMENTAS PARA DESENVOLVIMENTO	28
5.1	Scikit-Fuzzy	28
5.2	Ambiente de simulação	31
6	SIMULAÇÕES	32
6.1	Controlador Fuzzy	32
6.2	Primeira simulação do tanque	32
6.3	Resultados da simulação	36
6.4	Simulação com controle simplificado	37
6.5	Controlador proporcional	39
6.6	Simulação com vazão de saída e vazão de entrada controlada	40
6.7	Efeitos da amostragem em controladores Fuzzy	46
7	RESULTADOS	48
7.1	Comparações	48
7.2	Resultados das simulações	49

8	CONCLUSÃO	50
8.1	Trabalhos futuros	51
8.1.1	Genetic-fuzzy	51
8.1.2	Aplicações práticas	52
	REFERÊNCIAS	53

1 INTRODUÇÃO

O controle de nível é aplicado em diversas áreas da indústria, inclusive em aplicações cotidianas, como em caixas d'água e tanques de combustíveis. Também é utilizado para garantir o controle quantitativo de elementos, sejam líquidos ou sólidos granulados, bem como a estabilidade e segurança na produção (GHM GROUP, 2018).

Existem vários sensores disponíveis no mercado, desenvolvidos para realizar a medição de nível de diferentes substâncias e em diversos tipos de aplicações (GHM GROUP, 2018). Uma forma simples de se fazer esse controle, que está presente até mesmo nas caixas d'água, é utilizando um sensor do tipo boia. Outros sensores, que possuem a mesma finalidade, são os sensores de nível do tipo hidrostático, capacitivo, ultra sônico e potenciométrico, cada um utilizado de acordo com a viabilidade de cada aplicação.

Segundo Dorf e Bishop (2001), o controle de sistemas requer modelagem e compreensão, até mesmo para processos onde esses requisitos não estão totalmente desenvolvidos, como em processos químicos e sistemas modernos de alta complexidade. Um sistema de controle é uma configuração formada pela conexão de componentes com a finalidade de gerar uma resposta desejada para aquele sistema (DORF; BISHOP, 2001).

Para realizar o controle de nível de processos, uma alternativa à utilização dos controladores clássicos, são os controladores que utilizam a lógica Fuzzy na sua construção. A lógica Fuzzy foi desenvolvida e estruturada por Lofti Asker Zadeh na década de 1960, a partir de princípios idealizados por Jan Lukasiewicz em 1920, que introduziu a ideia de conjuntos com grau de pertinência e conceitos da lógica clássica, desenvolvida por Aristóteles (RIGNEL; CHENCI; LUCAS, 2011).

Controladores Fuzzy aplicados em sistemas e processos que requerem controle, são construídos utilizando a compreensão e a experiência humana. Ao contrário dos controladores clássicos, que necessitam de grande entendimento e conhecimento físico, controladores Fuzzy abordam o problema de forma linguística, o que torna a implementação rápida e efetiva (SIMÕES; SHAW, 2004).

Este trabalho apresenta a construção e a simulação de um controlador Fuzzy, com o propósito de ajustar o nível de líquido em um tanque de acordo com o valor de referência estipulado, para isso é simulada a estrutura de um tanque a fim de permitir seu modelamento e análise. Utilizando-se conceitos da teoria Fuzzy para o desenvolvimento do controlador, é implementado um controlador capaz de responder de forma satisfatória e precisa às variações da referência.

Segundo Simões e Shaw (2004), pode-se projetar controladores Fuzzy para

ter ou apresentar comportamento dedutivo ou indutivo, sendo o dedutivo a habilidade de fazer inferências com base em informações já conhecidas, enquanto os indutivos, tratam da capacidade de aprender através de exemplos particulares gerados a partir de observações do comportamento do processo.

Para realizar o controle de nível em um tanque, foi construído um controlador Fuzzy com comportamento dedutivo, dessa forma, os conjuntos e funções de pertinência gerados posteriormente, assim como a base de regras geradas, foram baseados em estudos prévios e informações já conhecidas sobre esse modelo. Para a confecção desse controlador, cada etapa foi montada e remodelada com o intuito propiciar uma resposta precisa do sistema.

1.1 OBJETIVO

A seguir são apresentados os objetivos gerais e os objetivos específicos do trabalho.

1.1.1 Objetivo Geral

Desenvolver propostas de controladores Fuzzy para controle de nível em tanques.

1.1.2 Objetivos Específicos

- Definir a estrutura de nível de tanque a ser usada para controle;
- Definir o controlador Fuzzy para o controle do nível de tanque;
- Identificar ferramentas para realizar a simulação e controle do sistema;
- Desenvolver controladores para cenários simulados.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte maneira: O capítulo 2 discorre brevemente sobre o que são sistemas de controle e suas principais configurações. O capítulo 3 apresenta a fundamentação teórica sobre a teoria Fuzzy, abordando questões como: conjuntos Fuzzy, funções de pertinência e variáveis linguísticas, este capítulo também mostra de que forma a teoria é aplicada para a confecção de um controlador Fuzzy. No capítulo 4 é feita a análise na forma de construção e nas características das respostas dos controladores confeccionados em aplicações semelhantes. O capítulo 5 aborda as questões relacionadas à ferramenta utilizada para realizar as simulações do controlador Fuzzy, também é apresentado de forma simples e direta um exemplo de construção de controlador Fuzzy fazendo uso desta ferramenta. O capítulo 6 apresenta as simulações do trabalho, nele é descrito a estrutura do tanque a ser

utilizado, o ambiente escolhido para as simulações e o método Fuzzy a ser empregado na construção do controlador. O capítulo 8 descreve as considerações finais realizadas.

2 INTRODUÇÃO À SISTEMAS DE CONTROLE

Sistemas de controle fazem parte do nosso dia a dia e são parte integrante na sociedade moderna, estando presentes desde simples aplicações como máquinas de lavar e geladeiras até as mais complexas como foguetes e veículos autônomos (NISE, 2013). De acordo com Nise (2013), existem inúmeros sistemas de controle no interior do próprio corpo humano, como, por exemplo: a regulação do açúcar no sangue feita pelo pâncreas e a forma em como seguimos visualmente um objeto em movimento para mantê-lo em foco.

Segundo Dorf e Bishop (2001), os fundamentos fornecidos da teoria dos sistemas lineares são a base para a análise de um sistema de controle. Este sistema é considerado como a união de vários componentes responsáveis pela formação de uma configuração, atuando-se sobre os componentes e supondo uma relação de causa e efeito, uma resposta desejada pode ser alcançada deste sistema (DORF; BISHOP, 2001).

2.1 SISTEMAS DE MALHA ABERTA E MALHA FECHADA

A configuração ou arquitetura interna de sistemas de controle podem ser divididas em dois principais tipos, sendo eles: sistemas de malha aberta e sistemas malha fechada (NISE, 2013). De acordo com Dorf e Bishop (2001), a relação entrada-saída representa uma relação de causa e efeito do processo e para sistemas de controle de malha aberta a entrada não é realimentada com a saída do sistema (não possui retroação). Enquanto que, para sistemas em malha fechada, sua resposta de saída é usada para comparar com a referência do sistema (caracterizando a retroação ou realimentação do sistema).

Uma característica distinta dos sistemas de malha aberta é a sua incapacidade de realizar compensações de perturbações que alterem a resposta da saída, podendo essas perturbações estarem presentes em um ou vários pontos no sistema. Dessa maneira, não há qualquer tipo de correção devido às perturbações nesse sistema, eles são simplesmente orientados por seu valor de entrada (NISE, 2013).

Os problemas presentes em sistemas de malha aberta, como a sensibilidade a perturbações e inabilidade de corrigir seus efeitos, podem ser solucionados por sistemas de malha fechada. Nesse tipo de sistema, por meio de um sensor, ocorre a medição do valor de saída e conversão para uma forma de sinal compreendida pelo controlador. Dessa forma, o controlador é capaz de comparar a entrada de referência com a saída da planta, podendo assim compensar os efeitos das perturbações na resposta (NISE, 2013).

3 LÓGICA FUZZY

A lógica booleana, diferentemente da lógica Fuzzy, permite apenas duas respostas possíveis para uma pergunta, verdadeiro ou falso. Esse tipo de abordagem limita a forma que determinado problema pode ser resolvido, uma vez que a complexidade dos problemas só aumenta. Sistemas convencionais precisam enfrentar problemas não-lineares e computacionalmente complexos, em alguns casos, os sistemas necessitam de grande consumo de processamento e em outros simplesmente não tem poder computacional suficiente para realizar a tarefa (COX, 1994).

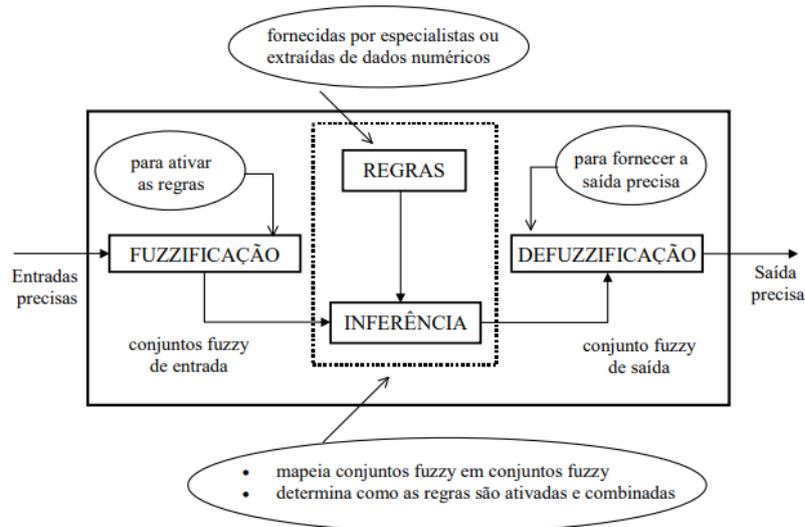
Na lógica Fuzzy podem existir infinitas respostas possíveis contidas no intervalo de 0 a 1, sendo o valor 0 completamente falso e o valor 1 completamente verdadeiro. As outras infinitas possibilidades dentro deste intervalo é o que pode ser caracterizado como incertezas, que adicionam ao sistema uma similaridade a forma de pensar de um ser humano.

Por meio da lógica Fuzzy é possível realizar a construção de controladores para sistemas lineares e não lineares. De acordo com Simões e Shaw (2004), por meio de regras individuais, que são característica de um controlador Fuzzy, a formulação de especialistas através de termos linguísticos é capaz de formar estruturas para execução de controles complexos.

Segundo Simões e Shaw (2004), controladores Fuzzy oferecem robustez e confiabilidade ao sistema devido ao processamento individual de cada regra gerada. Um erro de cálculo em um controlador convencional, que precisa de processamento sequencial de equações complexas, torna todo o resultado subsequente questionável, enquanto o erro em uma das regras do controlador Fuzzy pode não deteriorar completamente a resposta.

O sistema de Inferência Fuzzy é dividido em quatro subconjuntos (Figura 1). Cada um dos subconjuntos desempenha uma função de modo que a resposta na saída, assim como na entrada do sistema, sejam valores precisos que podem ser utilizados nas mais variadas aplicações, esses valores são referenciados como valores *crisp*.

Figura 1 – Sistema de Inferência Fuzzy



Fonte: Tanscheit (2004, p. 26).

Segundo Tanscheit (2004), a grande maioria das aplicações apresentarão um valor de entrada gerado a partir da leitura de sensores ou por meio de observações, por isso, primeiro é feito a fuzzificação desses dados utilizando-se os conjuntos Fuzzy. Após isso, são aplicadas as regras Fuzzy, a Inferência Fuzzy e a defuzzificação para encontrar o valor de saída correspondente a essa entrada.

3.1 FUZZIFICAÇÃO

Após a inserção dos dados no controlador, a fuzzificação é a primeira etapa realizada no sistema de inferência Fuzzy. Um dos objetivos dessa etapa é fazer a transformação do valor *crisp* (valor não Fuzzy) para uma característica que não pode ser mensurada de forma precisa, para essa característica é dado o nome de variável linguística.

3.1.1 Variáveis Linguísticas

Variáveis linguísticas podem ser utilizadas para descrever uma relação dos valores de entrada e de saída (SIMÕES; SHAW, 2004). Utilizando-se como exemplo uma variável de entrada referente a estatura de um indivíduo, é possível subdividir de forma simples em 3 subconjuntos, sendo eles: baixa, mediana e alta. Se utilizado a variável de temperatura como entrada, facilmente subdividi-se ela em um número maior de subconjuntos, sendo eles: muito frio, frio, agradável, quente e muito quente.

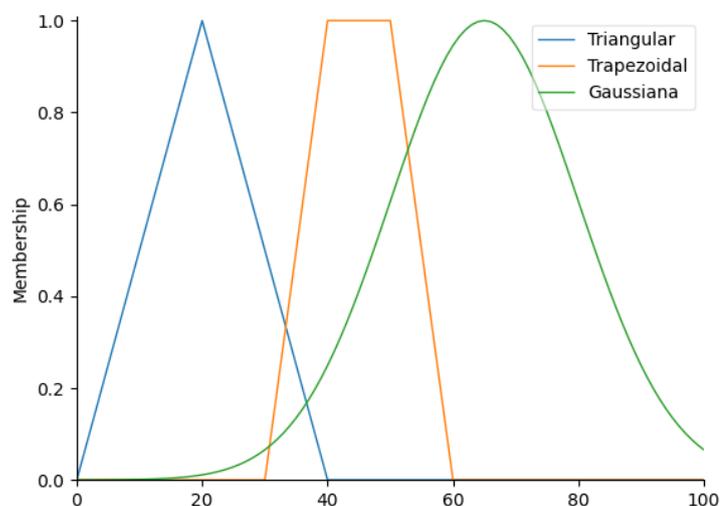
De acordo com a teoria Fuzzy, o valor *crisp* de uma variável pode fazer parte de 1 ou mais conjuntos, cada conjunto sendo representado por apenas 1 variável linguística. Para a união desses conjuntos é dado o nome de conjunto Fuzzy, e ele

engloba todo o intervalo em que o valor da variável em questão pode ser encontrado.

3.1.2 Funções de pertinência

As funções de pertinência delimitam e moldam o grau de pertencimento de um valor *crisp* em um ou mais conjuntos. Segundo Tanscheit (2004), o tamanho do intervalo e a forma das funções de pertinência podem ser criados a partir da compreensão do usuário sobre determinado tema, mas também é possível fazer o uso das funções de pertinência mais comuns, tais como: triangular, trapezoidal e Gaussiana (Figura 2).

Figura 2 – Funções de pertinência

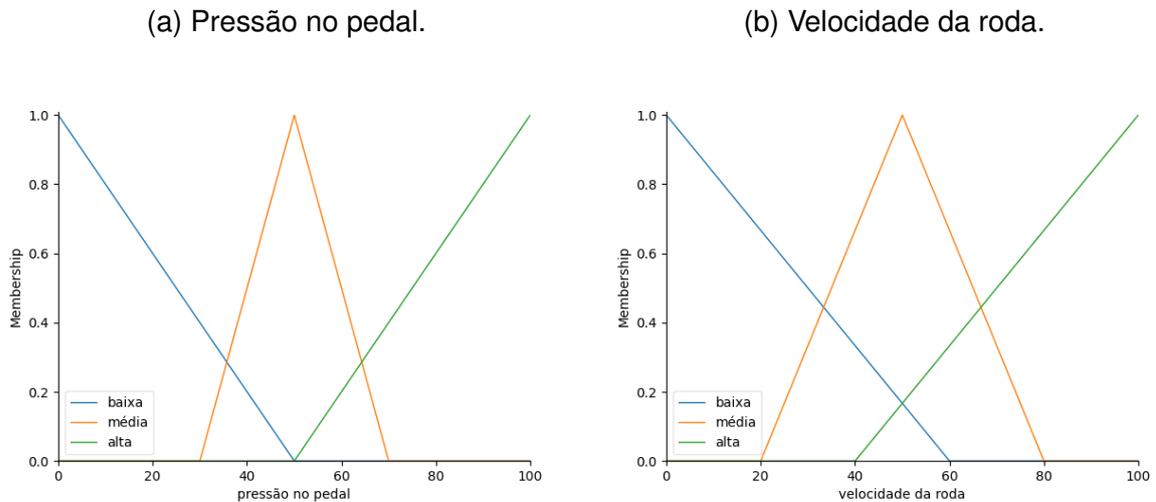


Fonte: elaborado pelo autor (2021).

O eixo Y do gráfico (figura 2) possui intervalo delimitado do valor 0 até o valor 1, esse intervalo representa o grau de pertinência de uma entrada *crisp* à um conjunto. Utilizando o raciocínio característico de um ser humano, o grau de pertinência pode ser interpretado como: o quanto um valor se encaixa em determinado conjunto. A resposta sendo 1 ou 0 significa a pertinência completa ou definitivamente não pertence ao conjunto, respectivamente.

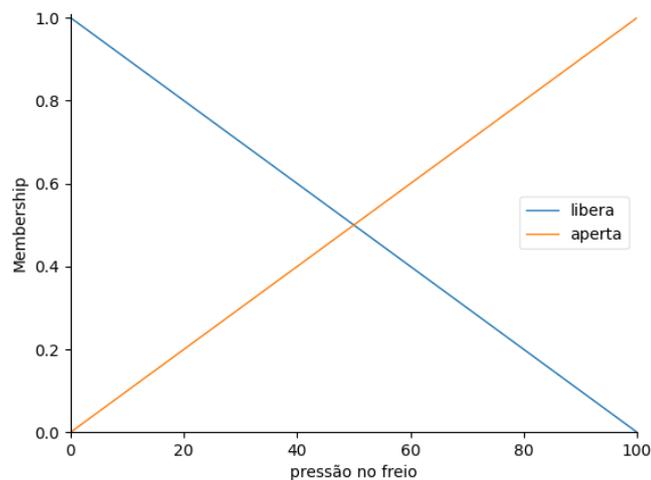
Utilizando como exemplo um sistema de controle Fuzzy responsável pelos freios de um carro quando as estradas ficam escorregadias e as rodas travam, as Figuras 3a e 3b mostram as funções de pertinência dos antecedentes, e a Figura 4 mostra as funções de pertinência do consequente. Outro antecedente presente neste exemplo é a função de pertinência da velocidade do carro, que neste caso é idêntico a Figura 3b.

Figura 3 – Antecedentes



Fonte: adaptado de Coppin (2004).

Figura 4 – Pressão nos freios



Fonte: adaptado de Coppin (2004).

Os gráficos mostram as variáveis linguística utilizadas na descrição dos conjuntos e também qual a função de pertinência que o conjunto apresenta, sendo todas essas funções do tipo triangular. No caso da Figura 4 é possível notar que, excluindo-se as extremidades 0 e 100, todos os valores fazem parte simultaneamente dos 2 únicos grupos existentes.

3.2 BASE DE REGRAS FUZZY

Com o término da etapa de fuzzificação dos valores de entrada é construído uma base de regras. Ela é baseada na experiência de pessoas que conhecem o processo e qual deve ser a resposta esperada do controlador dependendo do valor

de entrada aplicado. A base de regras é montada em uma estrutura de (Se... então...), sendo a primeira parte (Se...) o antecedente, responsável por uma ou mais entradas, e a segunda parte (então...) o conseqüente, responsável pela resposta de saída.

De acordo com Gomide e Gudwin (1994), a base de regras apresenta a vantagem da independência entre cada uma das regras geradas, isso permite o ajuste de uma parte isolada do modelo inicialmente montado. Esse ajuste individual é aplicado quando uma certa parte da resposta de saída não condiz com a resposta esperada, quando isso ocorre é possível alterar apenas a regra que não demonstrou o comportamento desejado, não sendo necessário repensar todo sistema novamente.

Utilizando o exemplo dos freios para um carro podemos montar uma base de regras Fuzzy (Tabela 1) para descrever o comportamento do conseqüente em relação aos antecedentes.

Tabela 1 – Regras Fuzzy

SE	a pressão no pedal de freio for média	ENTÃO	aplicar o freio
SE	a pressão no pedal de freio for alta E a velocidade do carro for alta E a velocidade das rodas for alta	ENTÃO	aplicar o freio
SE	a pressão no pedal de freio for alta E a velocidade do carro for alta E a velocidade das rodas for baixa	ENTÃO	liberar o freio
SE	a pressão no pedal do freio for baixa	ENTÃO	liberar o freio

Fonte: adaptado de Coppin (2004).

3.3 INFERÊNCIA FUZZY

Segundo Tanscheit (2004), a inferência Fuzzy busca modelar e imitar por meio de regras linguísticas a forma de raciocínio humano para tomada de decisões, sendo essas aplicações concentradas em grande parte na área de controle. Com as etapas passadas de fuzzificação e tabela de regras concluídas, são utilizadas essas informações para realizar a aproximação de raciocínio na etapa de inferência.

Para realizar a inferência de sistemas Fuzzy existem vários métodos baseados em regras linguísticas, tais como: método Mamdani, Sugeno e Tsukamoto. Cada um desses métodos apresenta diferenças entre si, que dependendo da aplicação, um pode ser mais apropriado do que o outro. O método Mamdani pode ser destacado por ser altamente intuitivo, assim como o método Sugeno pode ser destacado por sua eficiência computacional.

Segundo Ross (2010), o método Mamdani é o mais comum tanto na literatura como na prática, por esse motivo o desenvolvimento deste trabalho utiliza deste método para a elaboração dos testes e avaliação dos resultados.

3.3.1 Método Mamdani

De acordo com Ross (2010), existem variações do método Mamdani, que são formas diferentes para fazer as agregações dos gráficos de antecedentes e diferentes modos de defuzzificação, como por exemplo o método max-min e o método max-product. A seguir é mostrada como é feita a variação para o método de agregação de Mamdani chamado de inferência max-min, sendo o mesmo utilizado nas simulações.

Para a apresentação do método max-min é utilizado o exemplo desenvolvido por Ross (2010) a fim de esclarecer de forma sucinta o seu funcionamento. Neste caso, para a exemplificação foi usado um sistema com duas variáveis de entrada, que representam os antecedentes, e uma de saída, representando o consequente.

O sistema proposto tem a função de calcular a energia cinética de um corpo. Para o cálculo dessa energia em joules é utilizada a equação 1.

$$k = \frac{mv^2}{2} \quad (1)$$

Onde k é a energia cinética e m e v representam a massa do corpo e a sua velocidade, respectivamente. Assim, formam-se duas funções de entrada, uma dependente do valor de m e outra dependente do valor de v , e uma consequente de saída, representando a energia cinética (k).

Com base em observações desse sistema são deduzidos alguns dos comportamentos esperados dele, como mostra a tabela (Tabela 2).

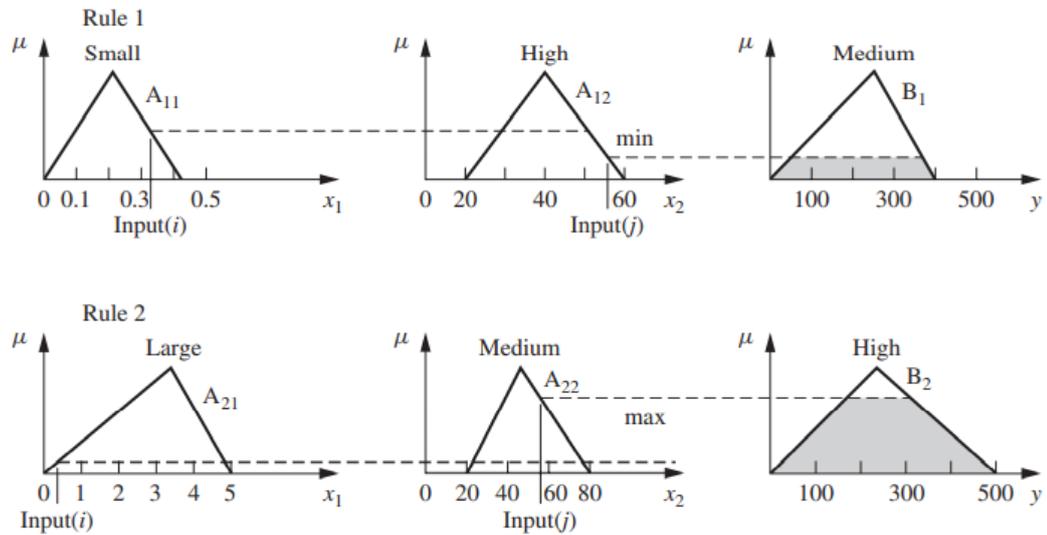
Tabela 2 – Regras Fuzzy para energia cinética

SE	a massa for pequena E a velocidade for alta	ENTÃO	a energia cinética é média
SE	a massa for alta OU a velocidade for alta	ENTÃO	a energia cinética é alta

Fonte: Adaptado de Ross (2010).

Assim, por meio de observações, é estimando uma massa de $0,350g$ e velocidade de $55m/s$ para um determinado corpo, esses dois valores são utilizados como valores *crisp* como entradas do sistema. As funções de pertinência e grau de ativação do sistema com essas entradas são mostradas na figura 5.

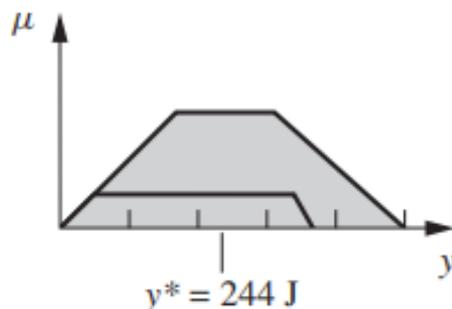
Figura 5 – Regras Fuzzy para energia cinética



Fonte: Ross (2010, p. 151).

Utilizando os valores *crisp* estimados anteriormente nas funções antecedentes e seguindo a lógica apresentada pelas regras descritas na tabela 2, são calculadas as saídas fuzzy para cada uma das regras. De acordo com a primeira regra se a massa de um corpo for pequena e sua velocidade alta, sua energia cinética será média. Por utilizar a relação **E** na regra, o valor de ativação da função consequente é mínimo. Para a segunda regra é utilizada a relação **OU**, dessa forma é selecionado o máximo para a ativação da função consequente.

Unindo-se as áreas das duas funções consequentes forma-se a geometria apresentada na figura 6 e a partir da defuzzificação dessa estrutura é extraído o valor *crisp* (y^*) utilizado como resposta do sistema. Nesse caso, a energia cinética encontrada pela defuzzificação utilizando o método do centroide é de 244 *joules*.

Figura 6 – Saída *crisp* do sistema Fuzzy

Fonte: Ross (2010, p. 151).

3.4 DEFUZZIFICAÇÃO

Devido ao uso do método Mamdani onde as funções de pertinência dos antecedentes resultam em formas geométricas nos consequentes, é necessário fazer o retorno dos resultados Fuzzy dessas estruturas a fim de se obter um valor resultante que seja aplicável a um sistema físico. Desse modo, se faz necessária a defuzzificação quando este método é aplicado.

A defuzzificação é o último estágio em um controlador Fuzzy e de acordo com Ross (2010), seu objetivo é agregar as saídas da inferência Fuzzy geradas pelos conjuntos difusos para obter um valor *crisp*, que será usado na planta de controle. Essa tarefa pode ser feita de diversas formas, como por exemplo:

- Método do centroide;
- Princípio da maior pertinência;
- Método dos pesos médios;
- Média das pertinências máximas.

Existem ainda mais métodos de se obter uma saída *crisp* a partir da resposta Fuzzy, porém é explicado e usado nas simulação o método do centroide, por ser o mais prevalente e fisicamente apelativo entre os métodos de defuzzificação segundo Sugeno (1985).

3.4.1 Método do centroide

Assim como o nome sugere, o método do centroide utiliza do cálculo do centroide de uma geometria como resposta à defuzzificação. Sendo a equação do centroide apresentada como:

$$z^* = \frac{\int \mu(z)z dz}{\int z dz} \quad (2)$$

Devido à necessidade de se realizar esse processo para obtenção de um valor real capaz de ser aplicado em um sistema físico, ocorre um processamento extra no controlador, que quando comparado a outros métodos de defuzzificação e métodos de inferência Fuzzy não é muito eficiente computacionalmente.

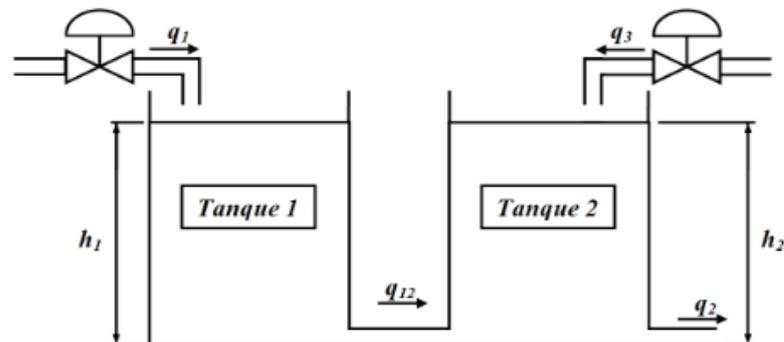
4 TRABALHOS SIMILARES

Para a definição da estrutura do tanque a qual o nível de água é controlado, inicialmente considerou-se estruturas já desenvolvidas em outros projetos semelhantes, sendo esses projetos utilizados para fins de comparação.

4.1 ESTRUTURA DO TANQUE

No trabalho apresentado por Silva et al. (2007), foi desenvolvida e analisada a resposta da lógica Fuzzy associada a algoritmos genéticos para o controle de nível de 2 tanques acoplados (Figura 7). De forma que, a lógica Fuzzy é utilizada para a confecção do controlador Fuzzy e os algoritmos genéticos para a sintonização dos parâmetros do controlador com o objetivo de melhorar a resposta transitória do sistema.

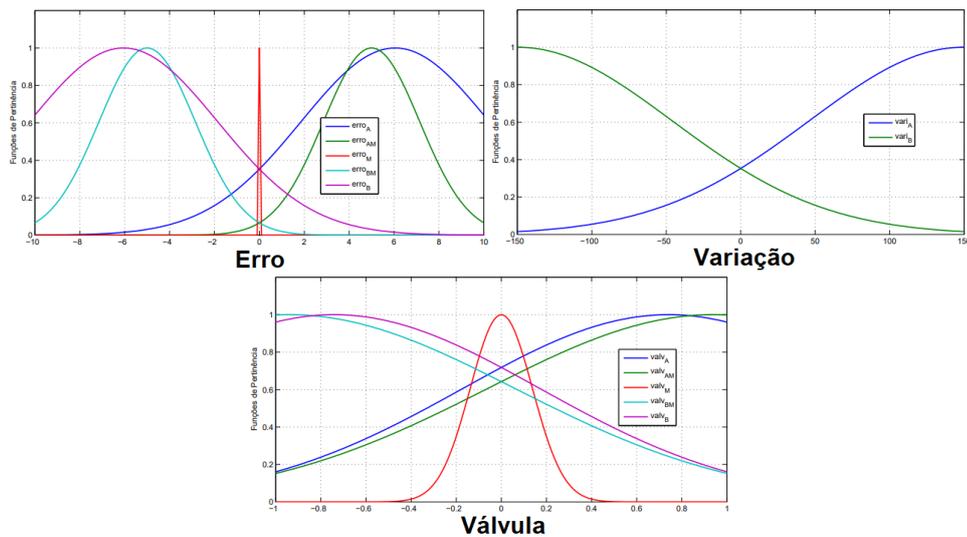
Figura 7 – Tanques acoplados



Fonte: Silva et al. (2007, p. 2).

As simulações foram realizadas utilizando o ambiente SIMULINK do MatLab 7 e o objetivo do controlador é manter a regulagem do nível do tanque 2 evitando o transbordamento do tanque 1. O primeiro controlador Fuzzy foi programado com 2 variáveis de entrada, sendo elas: o *erro*, possuindo 5 funções de pertinência, e a *variação*, possuindo 2 funções de pertinência. A variável de saída do controlador é responsável pelo controle da válvula, sendo esta descrita por 5 funções de pertinência (Figura 8).

Figura 8 – Conjuntos do controlador Fuzzy com algoritmos genéticos



Fonte: Silva et al. (2007, p. 4).

Para o primeiro controlador montado por Silva et al. (2007), as funções de pertinência apresentadas na figura 8 possuem forma Gaussiana e com elas são confeccionadas um total de 8 regras. Essas regras ditam qual deve ser a saída do controlador de acordo com a combinação das variáveis de entrada. Enquanto o algoritmo genético é utilizado para variar a forma dessas funções Gaussianas e o método de defuzzificação utilizado, a fim de encontrar a melhor resposta. Para o primeiro controlador é totalizado 13 parâmetros.

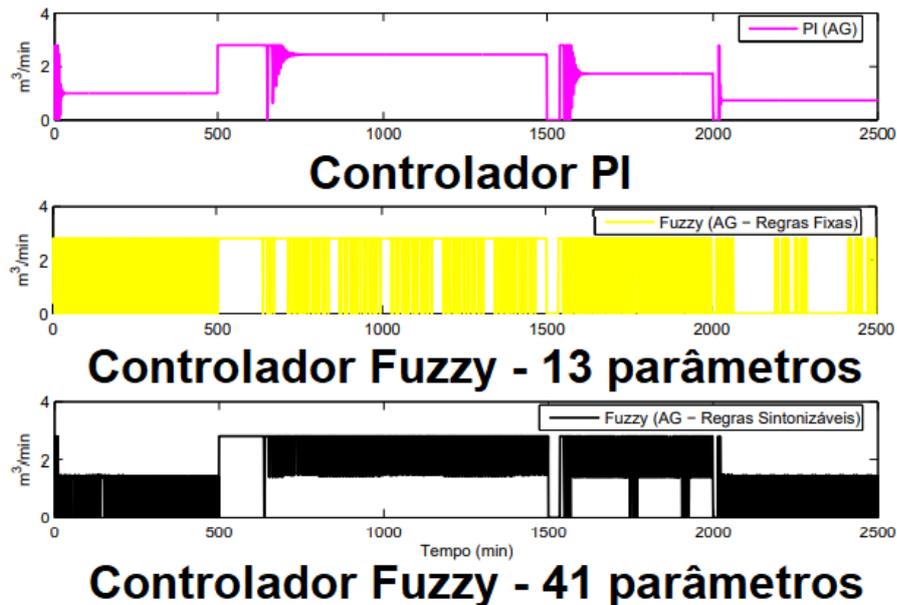
Para o segundo controlador Fuzzy a variável de entrada referente à *variação* passa a utilizar 5 funções de pertinência ao invés de 2 (como no primeiro caso). Agora, o algoritmo genético é responsável por encontrar a melhor Gaussiana para as 15 funções de pertinência (5 funções na entrada do *Erro*, 5 na entrada da *Variação* e 5 na saída da *Válvula*), a melhor relação entre as 25 regras geradas (combinação de todas as funções de pertinência dos antecedentes entre si) e o melhor método de defuzzificação, totalizando 41 parâmetros de sintonização.

Em Silva et al. (2007), também há a implementação de um controlador PI sintonizado utilizando algoritmos genéticos para fins de comparação. A sintonização de todos os controladores é feita de forma a selecionar o indivíduo que apresente o menor somatório do erro, que é cálculo da diferença entre o *set-point* e o nível atual de líquido no tanque 2, dessa forma o controlador escolhido é o que apresenta a resposta de convergência mais rápida.

Para os dois controladores Fuzzy sintonizados por algoritmo genético e para o controlador PI o resultado da simulação segue a referência de entrada e não apresenta sobre-sinal da resposta, porém para o segundo controlador Fuzzy, que utiliza 41 parâmetros de sintonização, a convergência para o valor final ocorre de forma mais rápida quando comparada aos outros controladores.

A convergência para o valor final do controlador PI é um pouco mais lenta em relação ao melhor controlador Fuzzy, porém a variação da resposta é muito menor (Figura 9). As grandes variações na válvula de entrada que os controladores Fuzzy apresentaram, ocorreram devido à construção das funções de pertinência, que são modelados de forma a permitir que ocorra uma mudança súbita no seu comportamento.

Figura 9 – Variação da abertura da válvula para os controladores implementados



Fonte: Silva et al. (2007, p. 4).

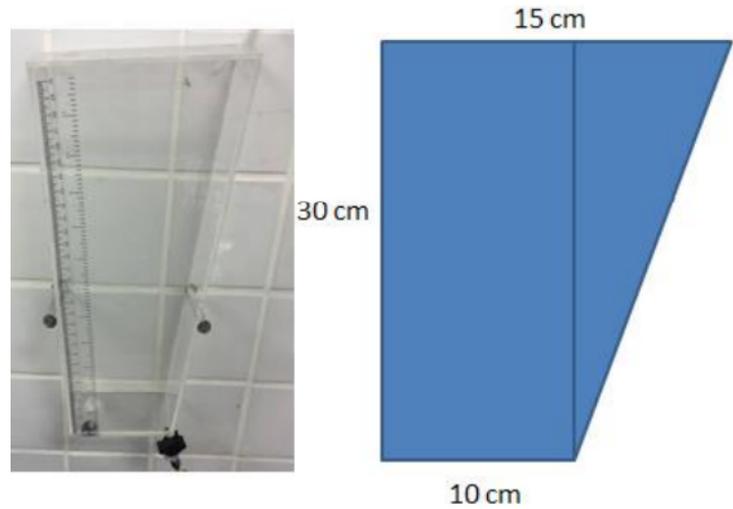
O trabalho desenvolvido por Vieira (2017) implementa um controlador Fuzzy e um controlador PI para o controle de 2 tanques diferentes, um de formato cilíndrico (Figura 10a) e outro com formato trapezoidal (Figura 10b), cuja área transversal aumenta com o nível do tanque. Nesse trabalho os controladores testados são primeiramente simulados utilizando-se a ferramenta Simulink do MatLab, e depois é feito novamente o teste nos tanques reais.

Figura 10 – Modelos de tanque usados por Vieira (2017)

(a) Tanque cilíndrico.



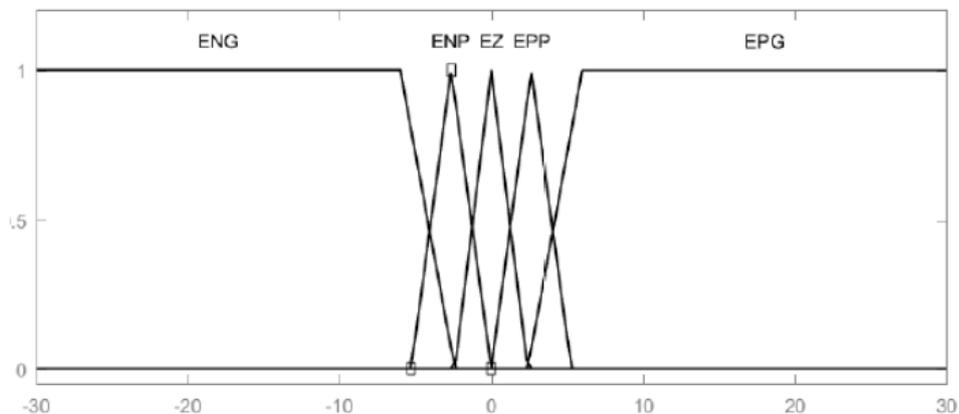
(b) Tanque trapezoidal.



Fonte: Vieira (2017, p. 34, 30).

O controlador Fuzzy programado por Vieira (2017) apresenta 2 variáveis de entrada, são elas: *erro* e *derivada do erro*; e 1 de saída: a *tensão*. A variável de entrada do erro (Figura 11) é dividida em 5 funções de pertinência, tendo seu domínio variando de -30 até 30 , sendo utilizado o formato trapezoidal e triangular para a delimitação de cada função. De forma análoga, a variável da derivada do erro possui 5 funções de pertinência concentradas no domínio de -0.3 até 0.3 , sendo elas trapezoidais e triangulares.

Figura 11 – Funções de pertinência da variável de entrada referente ao erro



Fonte: Vieira (2017, p. 40).

Combinando-se os conjuntos das variáveis de entrada formam-se 25 regras

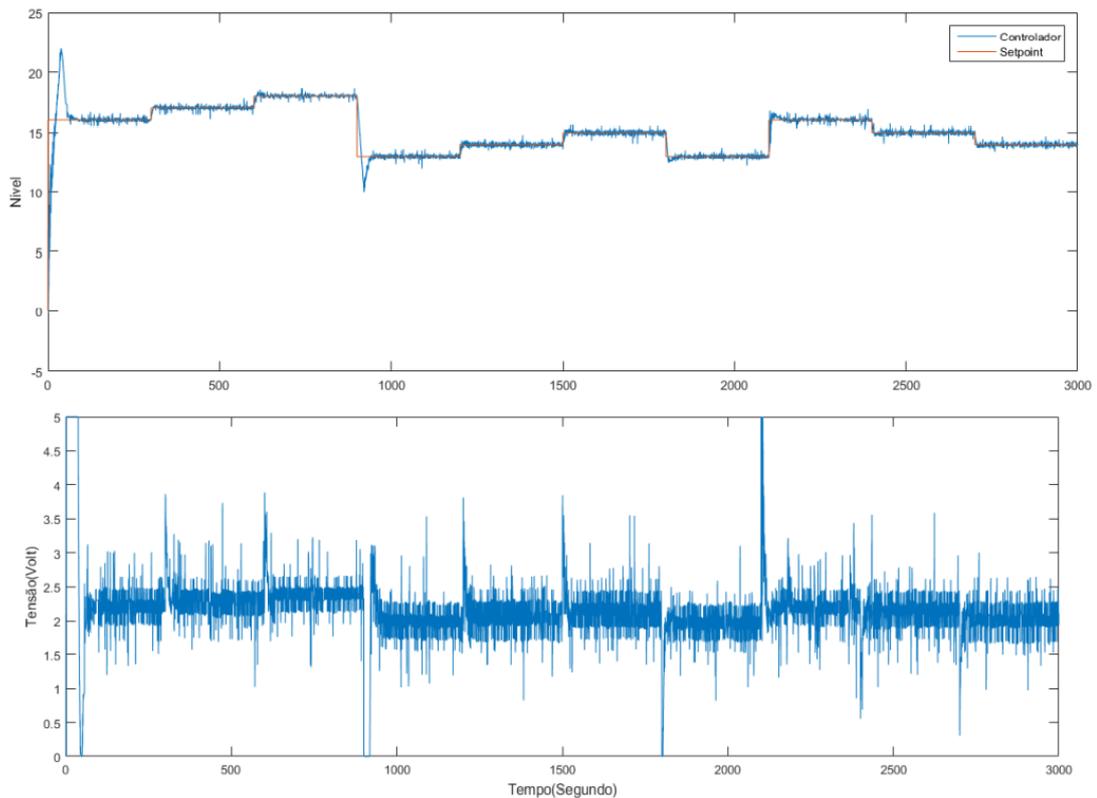
para esse sistema, que atuam na variável de saída do controlador Fuzzy e tem a finalidade de manter o nível do tanque na altura desejável. Para a variável de saída do controlador também foram definidas 5 funções de pertinência, que controlam a tensão de alimentação do motor elétrico que bombeia a água, aumentando ou diminuindo a entrada de líquido no tanque.

O controlador PI desenvolvido por Vieira (2017) é utilizado para comparar os resultados em relação ao controlador Fuzzy implementado. Esse controlador foi programado para apresentar característica agressiva, de forma a priorizar o tempo de acomodação da resposta, o que gera, conseqüentemente, variações bruscas de tensão sobre o motor controlado. Por outro lado, o controlador Fuzzy é implementado para gerar uma resposta mais suave, aumentando assim o tempo de acomodação.

O resultado dos controladores pela ferramenta de simulação Simulink foram muito similares entre si, eles seguem a referência de entrada e tem baixo tempo de acomodação. Por ser mais agressivo, o controlador PI demonstrou maior velocidade de convergência da resposta, enquanto que o controlador Fuzzy, como já esperado de sua implementação, apresenta um tempo de acomodação da resposta levemente maior.

Nos testes práticos realizados nos 2 tanques e com os 2 controladores (Fuzzy e PI) além do ruído natural de sistemas reais, o sinal do sensor responsável por medir a altura de água no tanque é passado ao controlador sem qualquer tipo de tratamento, desafiando ainda mais os controladores. Com os resultados obtidos nos experimentos dos 2 tanques é possível identificar de forma nítida as peculiaridades de cada controlador.

Figura 12 – Resposta experimental do controlador PI no tanque trapezoidal

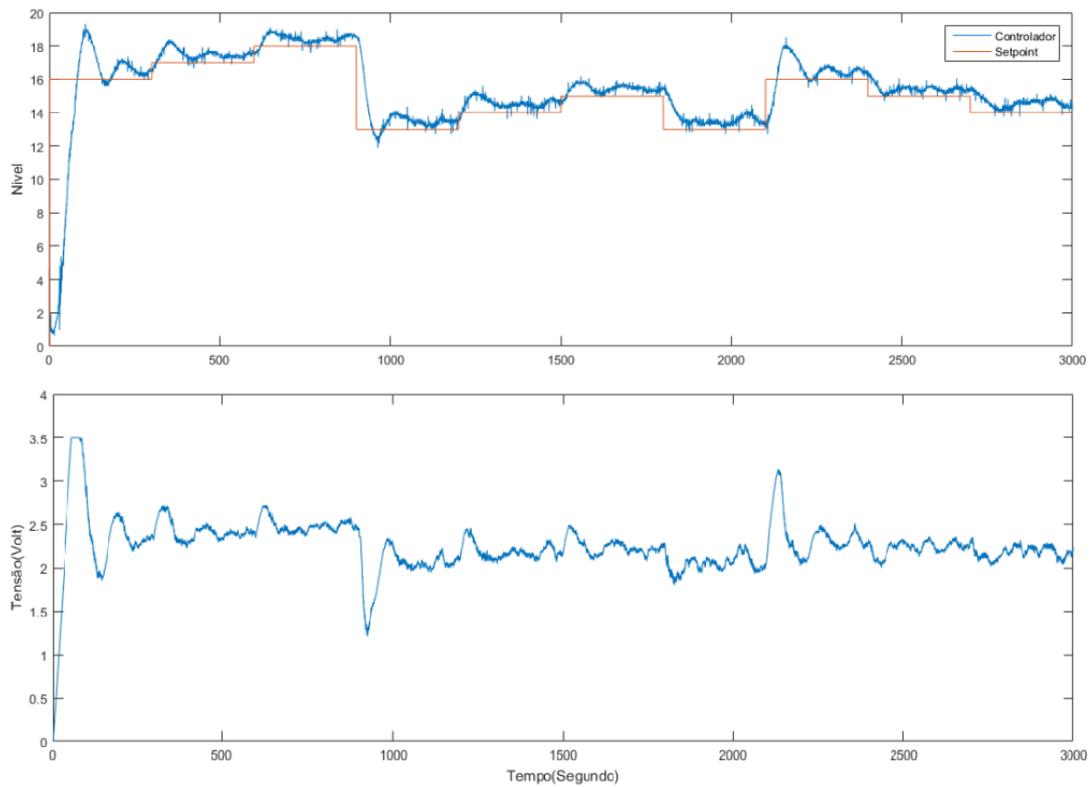


Fonte: Vieira (2017, p. 65).

O resultado do teste aplicando-se o controlador PI para controlar o nível de água no tanque trapezoidal é apresentado na Figura 12, o primeiro gráfico mostra como o controlador segue com pouco erro a referência, por outro lado, para que ocorra essa precisão no controle há grande variação de tensão no motor elétrico.

O mesmo experimento é realizado com o controlador Fuzzy, cuja resposta é apresentada pela Figura 13. A resposta não segue de forma precisa a referência como ocorria com o controlador PI, porém a variação da tensão no motor é muito menor.

Figura 13 – Resposta experimental do controlador Fuzzy no tanque trapezoidal



Fonte: Vieira (2017, p. 67).

Dessa forma, para fins didáticos, primeiramente é proposto a simulação de um tanque simples, apenas com uma vazão de entrada e sem saída. Com base nele é estudado o comportamento, o efeito do formato das funções de pertinência e as principais características apresentadas por um controlador Fuzzy.

Após o término do estudo e da implementação para o primeiro tanque, a segunda estrutura proposta é a de um tanque com vazão de entrada controlada e vazão de saída. Neste caso, além da análise do seu comportamento, também é analisado a velocidade de convergência da resposta e os efeitos da amostragem.

5 FERRAMENTAS PARA DESENVOLVIMENTO

Para o desenvolvimento deste trabalho é utilizada a linguagem de programação Python. Para isso buscou-se bibliotecas que disponibilizavam a montagem e o uso de ferramentas para a confecção de controladores Fuzzy. Algumas das bibliotecas encontradas foram a *pyfuzzylite*¹, *fuzzylab*², *simpful*³ e *scikit-fuzzy*⁴.

Algumas dessas bibliotecas ainda estão em processo de desenvolvimento ou oferecem poucos recursos para a montagem e visualização do controlador. Por esse motivo, foi selecionada para o desenvolvimento dos testes e simulações a biblioteca *scikit-fuzzy*, que fornece diversas funções e ferramentas para auxiliar a geração do controlador Fuzzy. Ela é aplicada para agilizar o desenvolvimento do objetivo geral do trabalho e facilitar a confecção e sintonia do controlador.

Foi utilizado um total de quatro pacotes para a realização das simulações, cada um desempenhando papel importante no código, sendo eles:

- *matplotlib*: biblioteca utilizada para *plotagem* de gráficos e visualização da resposta ao longo do tempo;
- *scipy*: disponibiliza a função *odeint*, que resolve sistemas de equações diferenciais ordinárias;
- *numpy*: ferramenta que auxilia na manipulação de listas e vetores;
- *scikit-fuzzy*: biblioteca utilizada para a programação e visualização dos resultados referentes a lógica fuzzy do controlador.

5.1 SCIKIT-FUZZY

As simulações e resultados apresentados neste trabalho são gerados utilizando a biblioteca Scikit-fuzzy, que trabalha em conjunto com a biblioteca numpy para auxiliar o usuário com a criação e manipulação de vetores. O objetivo deste pacote é oferecer de forma simples e com poucas linhas de código maneiras de criar sistemas Fuzzy, além de auxiliar com a visualização dos sistemas montados.

Utilizando o exemplo do sistema de controle Fuzzy para os freios de um carro, são mostradas as etapas para a construção de parte do modelo de controle. Com a aplicação da biblioteca *scikit-Fuzzy* é apresentada no código 1 a construção das funções de pertinência associadas as suas respectivas variáveis linguísticas, essas são usadas para formar os conjuntos de entrada do sistema.

¹ <https://pypi.org/project/pyfuzzylite/>

² <https://pypi.org/project/fuzzylab/>

³ <https://pypi.org/project/simpful/>

⁴ <https://pypi.org/project/scikit-fuzzy/>

Código 1 – Implementação da pressão do pedal utilizando a Scikit-fuzzy

```

1 import numpy as np
2 import skfuzzy as fuzz
3 from skfuzzy import control as ctrl
4 import matplotlib.pyplot as plt
5
6 pressao = ctrl.Antecedent(np.arange(0, 101, 1), 'pressao no pedal')
7
8 freio = ctrl.Consequent(np.arange(0, 100.1, .1), 'pressao no freio')
9
10 pressao['baixa'] = fuzz.trimf(pressao.universe, [0, 0, 50])
11 pressao['media'] = fuzz.trimf(pressao.universe, [30,50,70])
12 pressao['alta'] = fuzz.trimf(pressao.universe, [50, 100, 100])
13
14 freio['libera'] = fuzz.trimf(freio.universe, [0,0,100])
15 freio['aperta'] = fuzz.trimf(freio.universe, [0,100,100])
16
17 regra1 = ctrl.Rule(pressao['media'], freio['aperta'])
18
19 controle = ctrl.ControlSystem([regra1])
20 engine = ctrl.ControlSystemSimulation(controle)

```

Utilizando da pressão aplicada no pedal do carro como exemplo, ela é dividida em um universo de 100 valores inteiros, pertencentes ao intervalo fechado de 0 até 100. A pressão aplicada no pedal é uma das entradas fornecidas para o controlador, isso significa que esse dado é um antecedente, por isso ela utiliza da função *Antecedent* para sua construção. O mesmo ocorre para a função do freio, que é a saída desse sistema.

Com o antecedente da *pressão no pedal* criado, é gerado a partir dele as funções de pertinência de acordo com a experiência e conhecimento de operadores humanos, responsáveis por descrever qual deve ser o seu comportamento. São atribuídas variáveis linguísticas para cada conjunto e a posição e forma que melhor descreve tal conjunto. Neste caso é utilizada o método *trimf* da biblioteca para gerar uma curva de formato triangular.

Finalizada a construção dos antecedentes e consequentes (Figuras 3a e 4) de acordo com a experiência de especialistas da área, é feita a base de regras que será responsável pela forma como os antecedentes influenciarão os consequentes. A demonstração de como é feita a programação de uma regra no código 1 utilizando da biblioteca *Scikit-fuzzy* é apresentada na linha dezessete, onde é observado como é utilizada as variáveis linguísticas para criar o tipo de relação que a saída do controlador terá com a entrada de dados.

Código 2 – *Input* e *output* de valores *crisp* no controlador Fuzzy

```

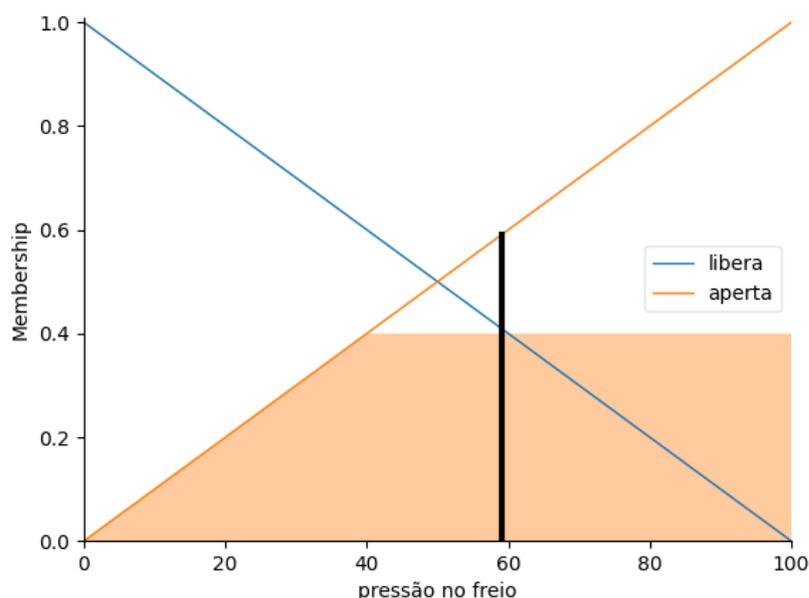
1 engine.input['pressao no pedal'] = 70
2 engine.input['velocidade da roda'] = 80
3 engine.input['velocidade do carro'] = 70
4
5 engine.compute()
6
7 output = engine.output['pressao no freio']
8 freio.view(sim=engine)

```

Para exemplificar a utilização do controlador é necessário alimentá-lo com valores de entrada fictícios, como demonstrado entre as linhas um a três do código 2. Normalmente esses valores são inseridos no controlador por meio de sensores presentes na planta, com o objetivo de monitorar o seu comportamento enquanto o controlador atua. Utilizando dos valores fictícios como exemplo, a resposta obtida à essa entrada é de 59,166.

O valor de 59,166 é encontrada por meio da defuzzificação, etapa necessária quando utilizado o método Mamdani, fazendo uso do método do centroide aplicado por padrão pela biblioteca *Scikit-fuzzy*. A resposta gerada pelo controlador (figura 14) é fornecida pela própria biblioteca utilizando-se o método *view* como exemplificado na linha oito do código 2.

Figura 14 – Resposta do controlador Fuzzy às entradas aplicadas no código 2.



Fonte: elaborado pelo autor (2022).

5.2 AMBIENTE DE SIMULAÇÃO

Diversos dos trabalhos analisados referentes a controladores Fuzzy e suas respectivas simulações foram desenvolvidos utilizando-se a ferramenta *Simulink*⁵. Não foi realizado um mapeamento sistemático para se saber qual ferramenta de simulação era a mais utilizada para este tipo de projeto, mas sim uma observação empírica.

Existem ferramentas alternativas de simulação que podem ser usadas para substituir o uso do *Simulink*, como por exemplo o *CIF 3*⁶. *CIF 3* ou *Compositional Interchange Format*, que é uma ferramenta *open source* que suporta o processo de desenvolvimento de controladores auxiliando também com as simulações, visualização e validação dos resultados.

Outra ferramenta alternativa ao *Simulink* que pode ser utilizada para a simulação de controle de nível de tanque é o uso de bibliotecas já disponíveis em Python. Neste caso, toda a simulação, assim como construção do controlador Fuzzy é implementado em um único código, facilitando a manipulação dos parâmetros usados e o tempo necessário para se fazer a aplicação.

⁵ <https://www.mathworks.com/products/simulink.html>

⁶ <https://cstweb.wtb.tue.nl/cif/trunk-r9682/>

6 SIMULAÇÕES

6.1 CONTROLADOR FUZZY

O método aplicado na construção do controlador Fuzzy é chamado método Mamdani. De forma mais precisa, é utilizado uma das variantes deste método, também conhecida como inferência *max-min*. O método Mamdani e a inferência *max-min* foi apresentada e exemplificada no capítulo 3.

Neste capítulo são implementados, por meio do uso das bibliotecas listadas no capítulo 5, os controlares responsáveis por realizar o controle de entrada de líquido no tanque. São expostas as formas como esse processo ocorre e a resposta resultante de cada projeto.

As simulações exibidas, começando pela aplicação mais simples, que é um tanque apenas com vazão de entrada, tem o propósito de fundamentar o funcionamento de controladores Fuzzy e como a mudança em suas funções de pertinência geram impactos na resposta. Um controlador proporcional é utilizado para comparação e são analisadas as semelhanças das respostas.

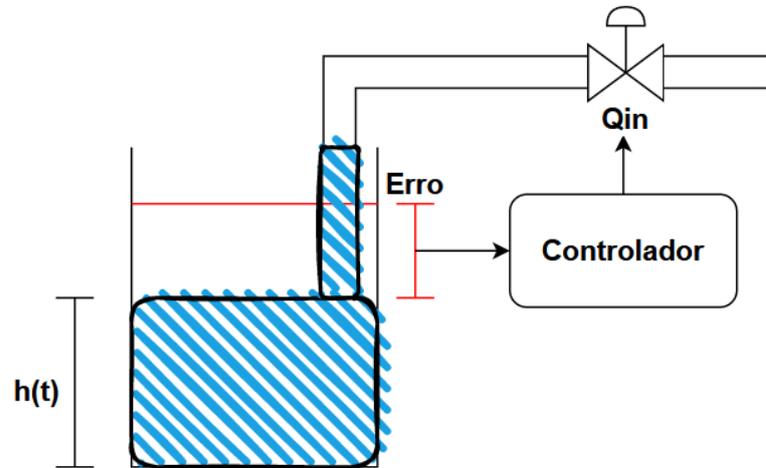
O próximo sistema a ser controlado possui vazão de saída livre e vazão de entrada controlada. Para esse sistema é primeiramente utilizado o mesmo controlador confeccionado para os sistemas com apenas vazão de entrada e analisada sua resposta. Após isso, é implementado outro controlador Fuzzy capaz responder de forma mais precisa à nova natureza da planta.

Por fim, são analisadas as respostas variando a taxa de amostragem da simulação, sendo discutidos os efeitos da subamostragem para controladores Fuzzy.

6.2 PRIMEIRA SIMULAÇÃO DO TANQUE

O código desenvolvido simula e controla o nível de água em um tanque com apenas uma vazão de entrada e sem vazão de saída para o líquido (Figura 15). Seu controle é feito utilizando um controlador Fuzzy, cuja responsabilidade é determinar o grau, em porcentagem, da abertura da torneira para entrada do líquido (Q_{in}). A válvula varia de completamente aberta (100%) até completamente fechada (0%).

Figura 15 – Esquemático do sistema a ser controlado



Fonte: elaborado pelo autor (2021).

O modelo do tanque da figura 15 é mostrada na equação 3.

$$\frac{dh}{dt} = \frac{Q_{valve}}{\rho A} \quad (3)$$

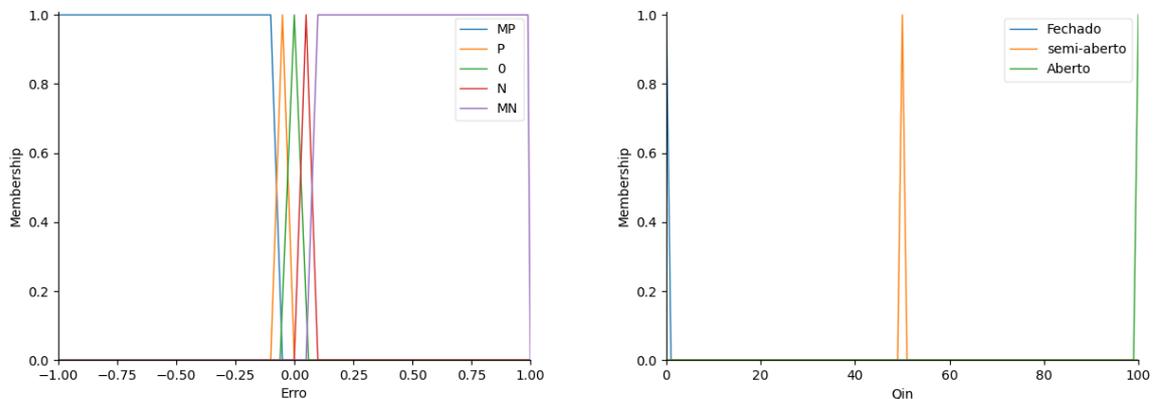
A equação 3 é interpretada como: a variação do nível de líquido no tanque depende da vazão máxima (Q) multiplicado pela porcentagem da válvula aberta ($valve$) dividido pela densidade do líquido (ρ) e a área da secção transversal do tanque (A).

Para a simulação são determinadas as condições iniciais do tanque e algumas variáveis relacionadas ao seu cálculo, tais como: a vazão máxima de entrada, o nível de líquido desejado, a densidade do líquido e a área transversal do tanque. O controlador Fuzzy implementado apresenta apenas uma função antecedente (Figura 16a), sendo o erro atual no sistema, e uma função consequente (Figura 16b), que é a porcentagem de abertura da válvula.

Figura 16 – Controle Fuzzy

(a) Função do erro.

(b) Porcentagem da válvula aberta.



Fonte: elaborado pelo autor (2021).

O código 3 apresenta a forma como as funções de pertinência da figura 16 são implementadas com a utilização da biblioteca *Scikit-fuzzy* e como é criada suas regras e montado o controlador.

Código 3 – Implementação do controle Fuzzy utilizando a Scikit-fuzzy

```

1 ## -----FUZZY CONTROL-----
2 ## Controla a vazao de entrada com base no erro de nivel
3 erro = ctrl.Antecedent(np.arange(-1, 1.01, 0.01), 'Erro')
4 vazao = ctrl.Consequent(np.arange(0, 101, 1), 'Qin')
5 #MN = muito negativo
6 erro['MN'] = fuzz.trapmf(erro.universe, [-1, -1, -0.10, -0.05])
7 erro['N'] = fuzz.trimf(erro.universe, [-0.10, -0.05, 0])
8 erro['O'] = fuzz.trimf(erro.universe, [-0.06, 0, 0.06])
9 erro['P'] = fuzz.trimf(erro.universe, [0, 0.05, 0.10])
10 erro['MP'] = fuzz.trapmf(erro.universe, [0.05, 0.10, 1, 1])
11 #MP = muito positivo
12 vazao['Fechado'] = fuzz.trimf(vazao.universe, [0, 0, 1])
13 vazao['semi-aberto'] = fuzz.trimf(vazao.universe, [49, 50, 51])
14 vazao['Aberto'] = fuzz.trimf(vazao.universe, [99, 100, 100])
15
16 regra1 = ctrl.Rule(erro['MP'], vazao['Aberto'])
17 regra2 = ctrl.Rule(erro['P'], vazao['semi-aberto'])
18 regra3 = ctrl.Rule(erro['O'] | erro['N'] | erro['MN'], vazao['Fechado',
19 ])
20 controle_vazao = ctrl.ControlSystem([regra1, regra2, regra3])
21 engine = ctrl.ControlSystemSimulation(controle_vazao)

```

Para a programação do controlador Fuzzy é necessário determinar o tamanho

do universo para o sistema em estudo. Assim, primeiramente é preciso saber quais são as dimensões do tanque simulado. Para essa simulação foi suposto um tanque com área de $0.01m^2$ (10cm por 10cm) e altura de 1m, a vazão máxima de entrada de líquido é de $2Kg/s$, ou, considerando a densidade da água igual a $1kg/L$, $2L/s$.

Com as dimensões estabelecidas, o tamanho do universo para esse sistema pode ser programado, assim são desenvolvidas as linhas dois e três no código 3. O universo do *erro* (linha dois) varia de -1 à 1 com espaçamento de 0.01 entre cada ponto, isso ocorre pois em um tanque de 1 metro de altura, apesar do sistema possuir apenas vazão de entrada, o erro total em relação à referencia pode ser tanto 1 metro positivo como 1 metro negativo. Já o universo da *vazão* (linha três) está descrito em porcentagem, podendo variar de 0% à 100% progredindo de 1% em 1% .

O passo seguinte é a montagem das funções de pertinência tanto do antecedente como do consequente. No código 3 nas linhas seis até dez são modelados o formato e a localização de cada função no antecedente do *erro*, assim como a variável linguística aplicada para cada curva. Enquanto que nas linhas doze à catorze, assim como ocorre para o antecedente, há o modelamento das funções no consequente. As variáveis linguísticas utilizadas para a programação do antecedente são siglas que resumem os seguintes significados a seguir.

- Muito Negativo (MN);
- Negativo (N);
- Zero (0);
- Positivo (P);
- Muito Positivo (MP).

Com as funções prontas são desenvolvidos três regras conforme trecho de código das linhas dezesseis à dezoito, a primeira é de manter a válvula completamente aberta enquanto o erro for muito negativo (*MN*). A segunda regra torna a válvula semi-aberta quando o erro passa a ser apenas negativo (*N*), e a última regra fecha completamente a vazão de entrada quando o erro é zero (*0*) ou passa a ser positivo (*P*).

O controlador Fuzzy recebe o erro atual do sistema (Figura 15), como implementado na linha dois (código 4), e o nível de abertura da válvula recebe o valor *crisp* gerado pelo controlador de acordo com a linha 6 do código 4. A seguir, a função *odeint* é responsável por resolver a equação diferencial do sistema, os argumentos passados para essa função são: o modelo do tanque (equação 3), o nível inicial de água presente no tanque, o tempo entre amostras e argumentos adicionais opcionais.

Código 4 – Implementação do controle Fuzzy

```
1 #Calcula o erro e insere no Antecedente
2 engine.input['Erro'] = SP - Level0
```

```

3 engine.compute()
4
5 # valve recebe saída crisp do controlador
6 valve = engine.output['Qin']
7 if(valve<1):
8     valve = 0
9 if(valve>99):
10    valve = 100
11
12 valve = valve/100 #porcentagem da valvula aberta
13
14 y = odeint(tank,Level0,[0,0.1],args=(Q, valve))

```

6.3 RESULTADOS DA SIMULAÇÃO

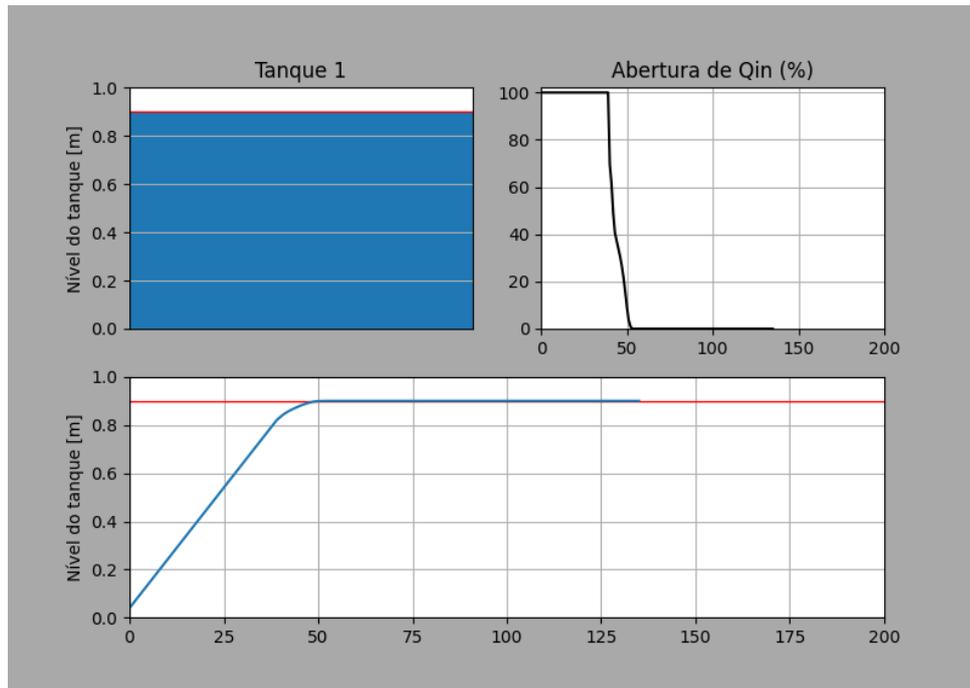
O resultado da primeira simulação é apresentado na Figura 17, sendo a imagem dividida em três gráficos. O gráfico superior esquerdo representa o tanque no momento atual, o gráfico superior direito é a abertura da válvula de entrada no decorrer do tempo e o gráfico inferior é a evolução de nível de líquido no tanque. O objetivo desta simulação é controlar a subida no nível de água até o valor de 90 centímetros, sinalizado pela linha vermelha horizontal.

Utilizando das dimensões propostas anteriormente, o tanque apresenta um volume total de $0.01m^3$ (10 litros de água), e considerando a vazão máxima de entrada de $2L/s$, é necessário 5 segundos para o seu completo enchimento. Pela Figura 17 nota-se o crescimento de forma linear até que o erro final seja de aproximadamente $0.08m$, após isso há uma atenuação na velocidade de convergência, isso ocorre devido à forma das funções antecedentes (Figura 16a) e a tabela de regras implementada no controlador.

Pela tabela de regras, quando o erro é muito negativo a válvula de entrada está totalmente aberta, resultando em vazão de entrada de $2L/s$. Ao se aproximar do valor final desejado de $0.9m$, a entrada *crisp* do controlador diminui o peso na função erro *muito negativo* e gradualmente aumenta de peso na função de erro *negativo*, essa transição gera uma mudança de comportamento no controlador. Agora, outra regra é ativada da tabela de regras, que é: se o erro é *negativo* a válvula é *semi-aberta*.

A mudança gradual do erro gera uma entrada no controlador que pode atingir uma ou mais funções de pertinência dos antecedentes. No caso do controle de nível do tanque, essa variação lenta aumenta o peso nas funções que limitam a vazão de líquido na válvula. Dessa forma, ocorre a diminuição gradual do erro e da vazão de entrada, até o ponto em que o erro final seja zero e a regra "Se o erro é igual ou maior que zero, então a válvula é fechada." seja ativada.

Figura 17 – Resultado da primeira implementação



Fonte: elaborado pelo autor (2021).

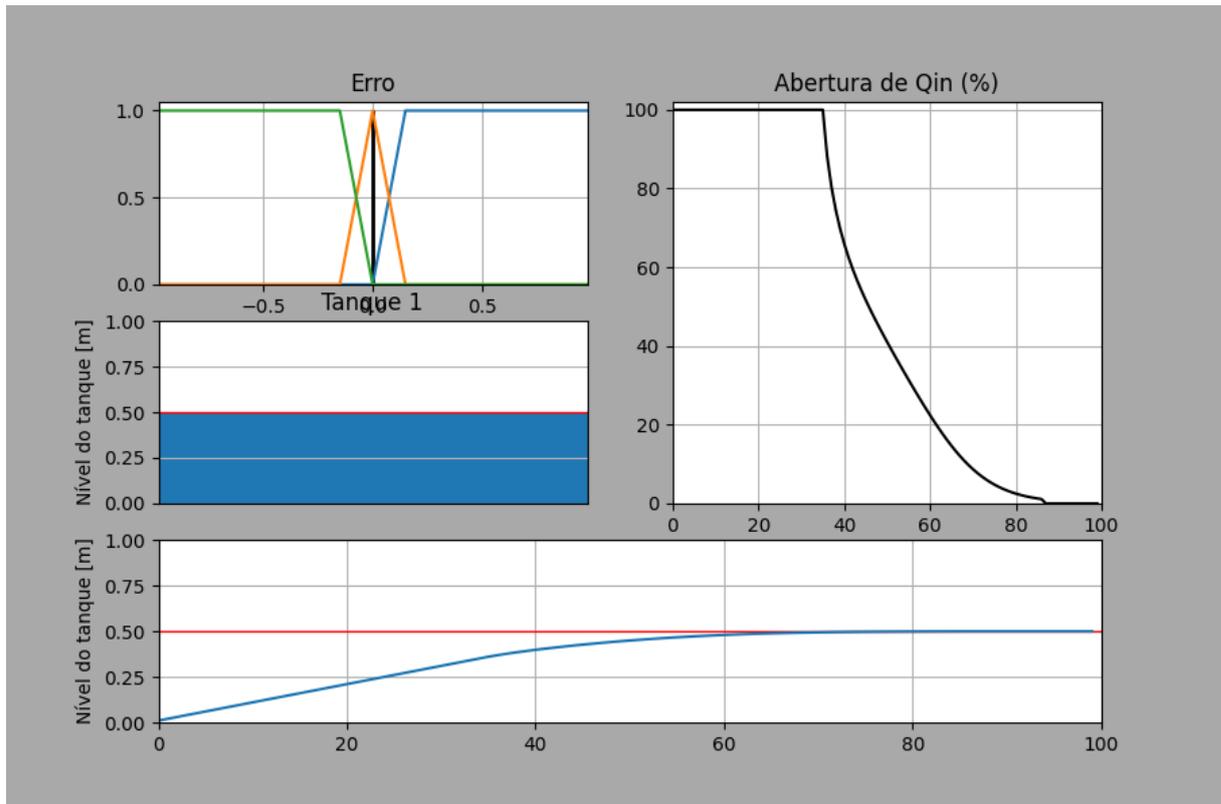
6.4 SIMULAÇÃO COM CONTROLE SIMPLIFICADO

Nesta secção são apresentadas duas simulações com os mesmos valores de vazão de entrada e volume do tanque, porém as funções antecedentes e consequentes são alteradas. Nos dois casos é feito o ajuste das funções antecedentes para apenas três funções de pertinência, sendo elas: *negativa*, *zero* e *positiva*; e no caso dos consequentes o número de funções de pertinência cai de três para duas, sendo expressos apenas pelas funções *aberto* e *fechado*.

O objetivo desse teste é analisar o comportamento da curva quando as funções são reduzidas ao mínimo necessário para se fazer o controle e de que maneira o formato das funções influencia na resposta.

No primeiro caso (Figura 18) a função de pertinência de erro *zero* é um triângulo com valores variando de -0.15 até 0.15 . A resposta para esse sistema é uma curva suave quando o nível de água se aproxima do valor final. Isso ocorre, pois a função de pertinência de erro zero impacta mais cedo a ação do controlador e tem uma variação menos abrupta durante a diminuição do erro.

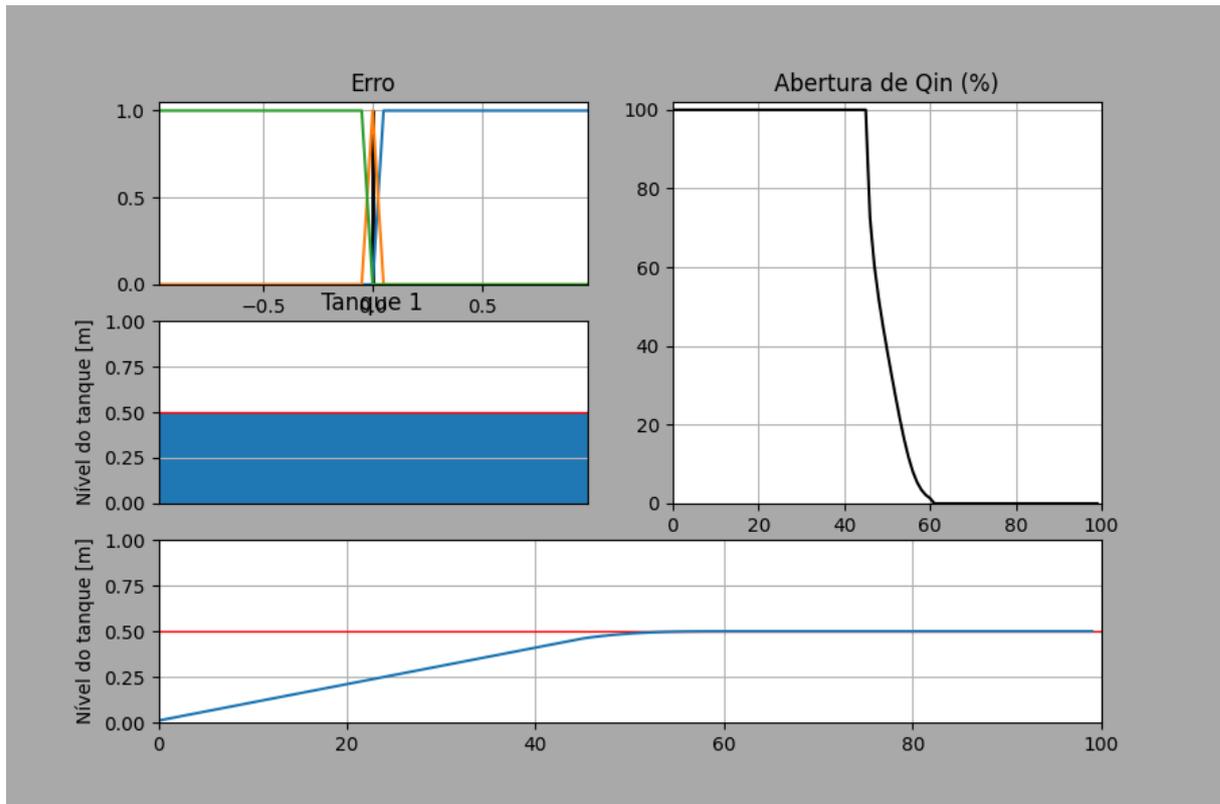
Figura 18 – Controlador Fuzzy de comportamento gradual



Fonte: elaborado pelo autor (2022).

No segundo caso (Figura 19), a função de pertinência em formato triangular quando o erro é zero varia de -0.05 até 0.05 , isso gera uma mudança na velocidade da resposta do controlador, pois demora mais até essa função ser ativada e seu efeito é mais abrupto. Assim como colocado por Ross (2010), na utilização do método Mamdani a sobreposição de funções de pertinência nos antecedentes tem efeito decisivo na suavidade da resposta do sistema. Devido a essa alteração, o resultado é uma convergência mais rápida e menos suave para o valor final.

Figura 19 – Controlador Fuzzy de comportamento determinístico



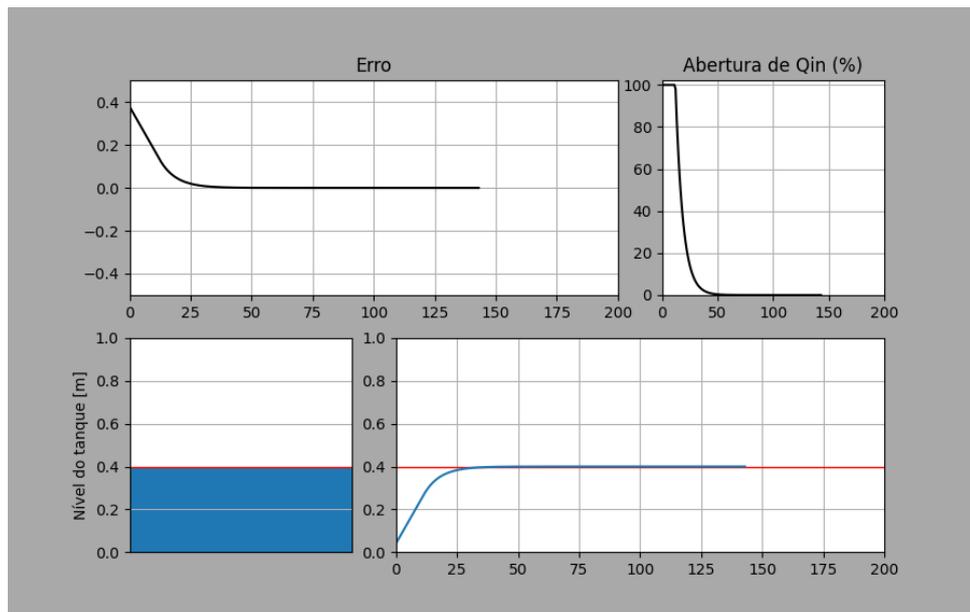
Fonte: elaborado pelo autor (2022).

6.5 CONTROLADOR PROPORCIONAL

Para o controle de nível de tanque simples com apenas uma vazão de entrada e sem saída de líquido (Figura 15), um controlador PID também pode ser implementado. Devido à simplicidade do sistema de estudo a utilização de um controlador proporcional é suficiente para responder de forma satisfatória quanto a velocidade de convergência e erro final próximo ao valor final estipulado.

O sistema de primeira ordem a ser controlado requer apenas a adição de um ganho proporcional elevado (controlador P) para que a convergência seja rápida. No exemplo da figura 20 é implementado um controlador com ganho proporcional elevado ($K_c = 700$), com esse valor a curva se assemelha a resposta do controlador Fuzzy da figura 18, cuja função triangular de erro zero é um pouco mais larga na base quando comparado a da figura 19.

Figura 20 – Resposta do controlador proporcional



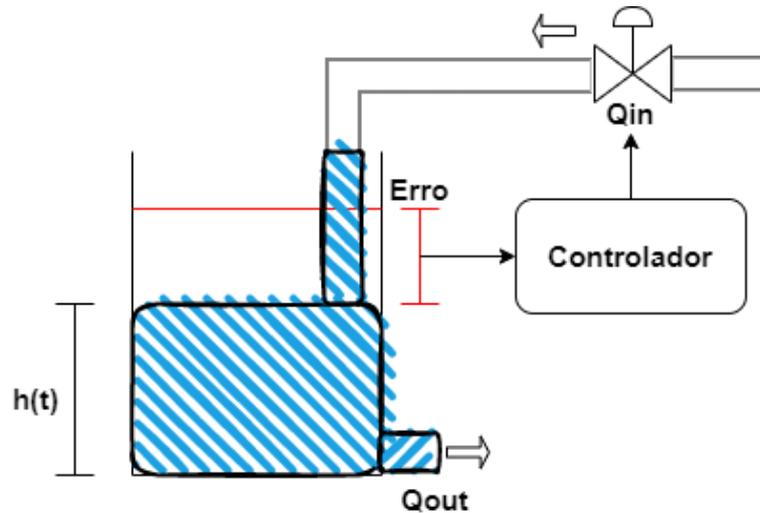
Fonte: elaborado pelo autor (2021).

Com o aumento do ganho proporcional do controlador a resposta do sistema será cada vez mais rápida e mais abrupta durante a convergência. Algo semelhante ocorre quando a função triangular de erro zero do controlador Fuzzy tem sua base reduzida, neste caso a resposta é menos nebulosa e o controlador tem uma ação mais determinística, gerando, dessa forma, convergência rápida ao valor final.

6.6 SIMULAÇÃO COM VAZÃO DE SAÍDA E VAZÃO DE ENTRADA CONTROLADA

Para esse problema a estrutura do tanque muda, agora há uma vazão de saída e outra de entrada de água (Figura 21). Dessa forma, os cálculos a serem realizados também mudam. Para esse novo sistema é considerado uma vazão de entrada Q_{in} e outra de saída Q_{out} , sendo Q_{out} dependente da altura de líquido presente no tanque. Para o cálculo da nova equação diferencial do tanque é utilizado a equação de Bernoulli (equação 4).

Figura 21 – Esquemático do novo modelo de tanque



Fonte: elaborado pelo autor (2022).

$$P_1 + \frac{\rho v_1^2}{2} + \rho g h_1 = P_2 + \frac{\rho v_2^2}{2} + \rho g h_2 \quad (4)$$

Na equação 4 P é a pressão atmosférica e é a mesma tanto na entrada quanto na saída do tanque, ρ é a densidade do líquido, v é a sua velocidade, h a diferença de altura dos pontos em comparação e g a constante gravitacional. Realizando a simplificação para o caso do tanque é obtido a equação 5.

$$\rho g h_1 = \frac{\rho v_{out}^2}{2} + \rho g h_2 \quad (5)$$

Assim, isolando-se v_{out} da equação 5 obtêm-se a equação 6.

$$v_{out} = \sqrt{2g\Delta h} \quad (6)$$

Com v_{out} determinado, a vazão de saída Q_{out} é encontrada substituindo v_{out} pela expressão: $\frac{Q_{out}}{A}$, obtendo-se assim as equações 7 e 8.

$$\frac{Q_{out}}{A} = \sqrt{2g\Delta h} \quad (7)$$

$$Q_{out} = A\sqrt{2g\Delta h} \quad (8)$$

Com o objetivo de encontrar a função que relaciona a altura de água no tanque e sua vazão de entrada e saída, parte-se da equação 9, onde: a variação de altura de

líquido no tanque (h') pode ser relacionada com a vazão (ΔQ) e a área transversal do tanque (A_1).

$$h' = \frac{\Delta Q}{A_1} \quad (9)$$

$$h' = \frac{Q_{in} - Q_{out}}{A_1} \quad (10)$$

Substituindo-se a equação 8 na equação 10, tem-se a relação final para o sistema onde existe uma vazão de entrada Q_{in} e uma vazão de saída Q_{out} dependente da altura de líquido do tanque.

$$\frac{dh}{dt} = \frac{Q_{in} - A_{out}\sqrt{2g\Delta h}}{A_1} \quad (11)$$

O controle no nível do tanque é aplicado adicionando-se uma válvula na vazão de entrada, dessa forma a entrada de líquido no tanque é liberada ou restringida de acordo com o controlador. Assim, a equação 12 é utilizada nas simulações.

$$\frac{dh}{dt} = \frac{Q_{invalve} - A_{out}\sqrt{2g\Delta h}}{A_1} \quad (12)$$

Com essa nova estrutura do tanque, quanto mais alto o nível de água, maior será a vazão de saída, por esse motivo é necessário um controlador Fuzzy que leve esse novo detalhe em consideração. A resposta esperada do controlador é alcançar a referência, independente da vazão na saída.

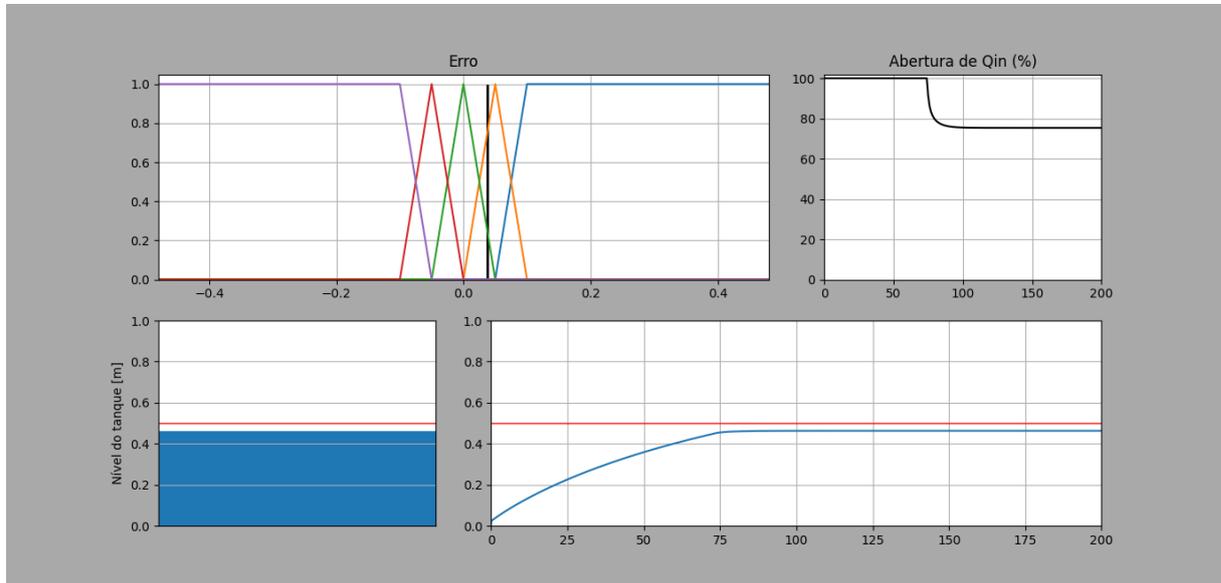
Porém, é inicialmente aplicado o controlador simples utilizado no modelo de tanque sem vazão de saída. Essa aplicação tem o intuito de analisar sua resposta frente a adição de uma nova variável no sistema, que é a variação na vazão de saída. Para as novas simulações é considerado um tanque de área de $1m^2$, altura de máxima de $1m$, vazão máxima $Q_{in} = 100L/s$ e a abertura de saída (Q_{out}) é um orifício quadrado posicionado no fundo do tanque de dimensões $10cm$ por $10cm$.

Nas simulações é usado um novo gráfico que apresenta a posição do erro atual e quais as funções de pertinência ele está ativando no momento. Na figura 22 esse novo gráfico é observado no canto superior esquerdo, enquanto os outros gráficos, assim como na primeira estrutura de simulação, apresentam a abertura da válvula (Q_{in}), o nível atual do tanque e sua evolução ao longo do tempo.

Aplicando-se o controlador para esse novo sistema, como mostrado na figura 22, nota-se que a resposta se aproxima da referência, porém nunca a alcança. A

conclusão é de que falta uma importante informação para o controlador atingir seu objetivo.

Figura 22 – Controle utilizando apenas o erro atual do sistema



Fonte: elaborado pelo autor (2022).

Ao contrário do modelo passado que utilizava apenas uma função antecedente, para o novo modelo é necessário duas funções antecedentes, uma para o erro atual, sendo calculado pela diferença entre a referência e o nível do tanque, e outra para a velocidade do erro, sendo calculado pela diferença entre o erro atual e o erro mensurado na leitura anterior do controlador.

Adicionando-se uma nova função antecedente é gerado novas relações no controlador. Isso significa que sua complexidade aumenta, pois novas regras precisam ser escritas a fim de gerar novos comportamentos esperados.

Com a adição de uma nova entrada para o sistema, é necessário relacionar as duas funções antecedentes entre si para determinar uma resposta na função consequente, que resultará na decisão de abrir ou fechar a válvula. As duas novas entradas e suas funções de pertinência podem ser observadas nas figuras 23a e 23b, enquanto que a função consequente é observada na figura 24.

Assim como para o controlador simples implementado para o tanque com apenas vazão de entrada, a legenda das funções antecedentes nas figuras 23a e 23b é interpretada da seguinte forma:

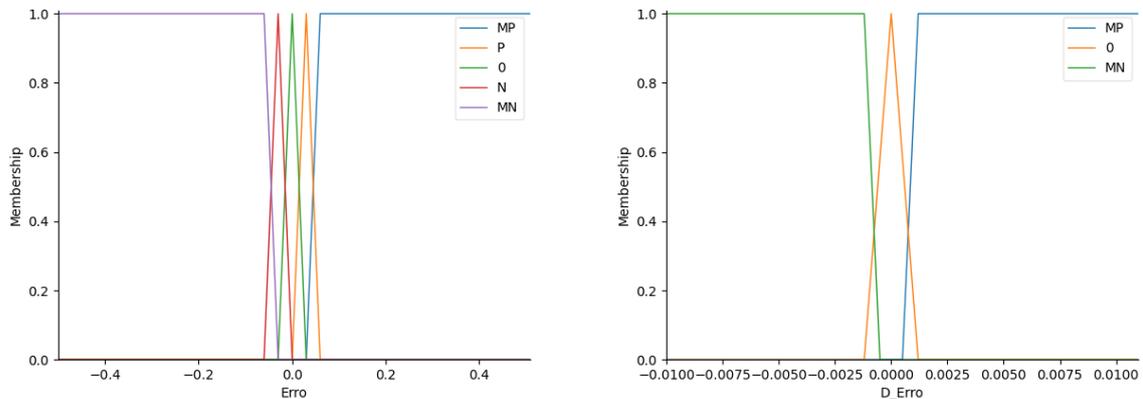
- Muito Negativo (MN);
- Negativo (N);
- Zero (0);
- Positivo (P);
- Muito Positivo (MP).

Enquanto que para a resposta de saída (figura 24) a legenda representa a própria vazão de entrada de líquido no sistema, variando de 0 até 100%.

Figura 23 – Funções Antecedentes

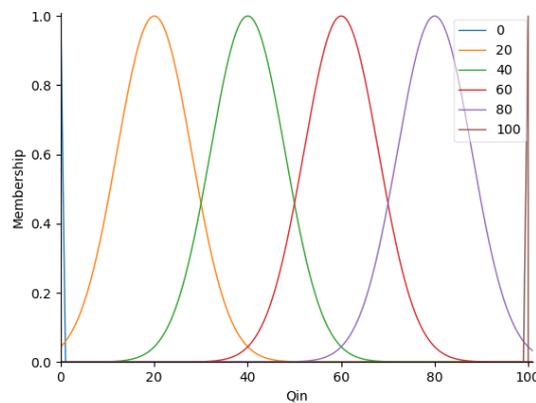
(a) Erro atual.

(b) Velocidade do erro.



Fonte: elaborado pelo autor (2022).

Figura 24 – Função consequente para a vazão no sistema



Fonte: elaborado pelo autor (2022).

Como exemplo de nova regra utilizando das figuras 23 e 24 determina-se que: se o erro for muito positivo e a velocidade do erro for muito positiva, então a válvula mantém-se completamente aberta. Assim, é preciso relacionar as entradas com as saídas do controlador, de forma que, para cada caso real ele se comporte como implementado.

Gerando-se as regras para o novo controlador é possível montar uma tabela de regras, como observado na tabela 3.

Tabela 3 – Regras para o controlador Fuzzy

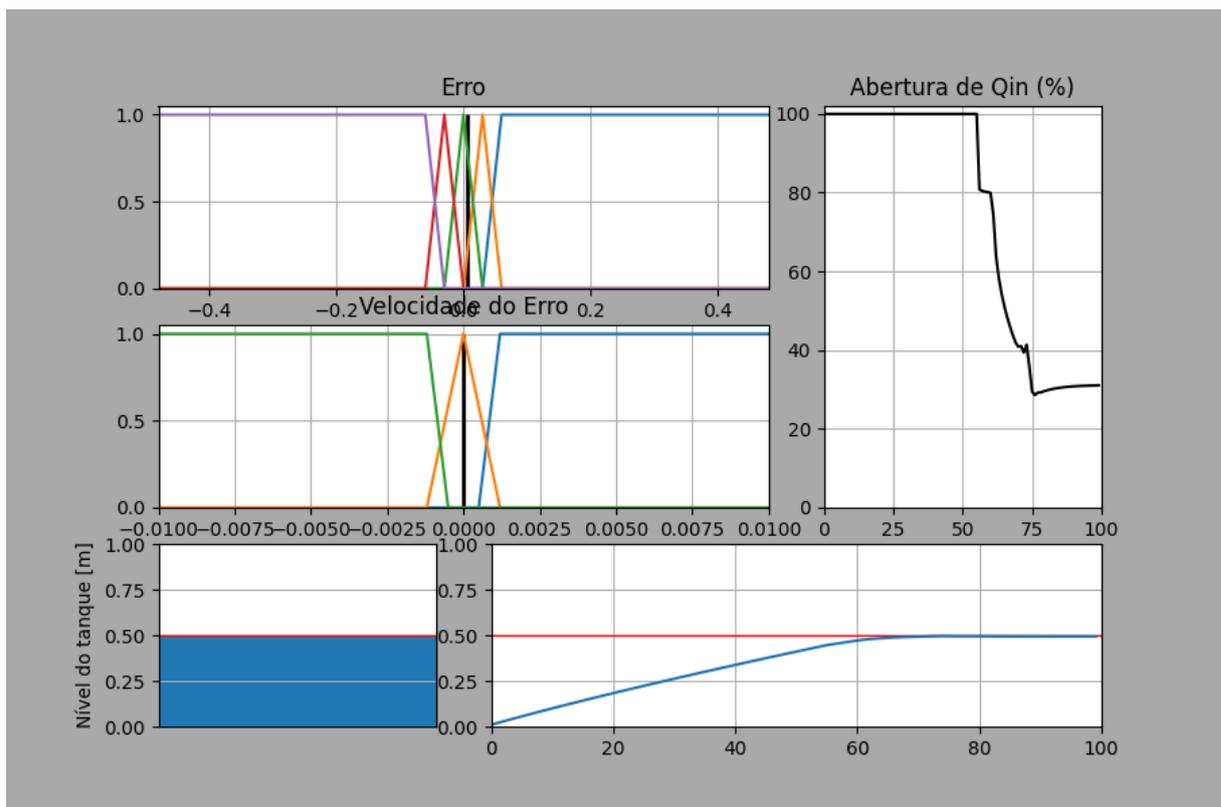
	MN	N	0	P	MP
MP	0	0	20	80	100
0	0	0	20	60	100
MN	0	0	40	60	100

Fonte: elaborado pelo autor (2021).

Na tabela 3 observa-se o comportamento do controlador em cada situação, nesse caso, a linha superior da tabela descreve as funções antecedentes do erro, enquanto a linha vertical à esquerda da tabela mostra as funções referentes à velocidade do erro. A combinação desses campos resulta na resposta esperada do controlador para cada situação.

Aplicado o novo controlador com as funções antecedentes e consequentes apresentadas nas figuras 23 e 24, e com as regras apresentadas na tabela 3, obtém-se a nova resposta da sua atuação sobre a planta.

Figura 25 – Resposta controlador com 2 funções antecedentes



Fonte: elaborado pelo autor (2022).

Como observado na figura 25, com a adição de mais regras e mais uma variável de entrada para o controlador, o seu comportamento agora se torna mais adaptado a nova planta. Sua resposta converge de forma assintótica para o valor de referência até que seja encontrado um equilíbrio entre a vazão de entrada e a de saída.

Os dois controladores Fuzzy programados tem a função de manter o nível de líquido no tanque no valor de referência, porém a primeira implementação não é capaz de atingir este objetivo, uma vez que sua simplicidade desconsidera um ponto fundamental na nova planta. Esse novo ponto, referente à velocidade variável na vazão de saída, é considerado e adicionado à segunda implementação. Com isso, sua complexidade aumenta e a convergência da resposta se torna possível.

6.7 EFEITOS DA AMOSTRAGEM EM CONTROLADORES FUZZY

Assim como para os controladores digitais a amostragem desempenha papel fundamental na programação e na resposta de um controlador Fuzzy. Para o caso de aplicações Fuzzy, seus efeitos influenciam diretamente a resposta de saída agindo nas regras em atuação no momento, como é explicado e exemplificado nesta seção.

No resultado apresentado pela simulação da figura 25 é utilizado uma amostragem de 0.1 segundos. Isso significa que a cada 0.1 segundos, o sensor presente na planta envia a informação de qual é o nível atual de líquido no tanque. De grosso modo, desconsiderando a vazão de saída e supondo Q_{in} máximo ($100L/s$), cada releitura feita pelo sensor pode representar uma variação máxima de $1cm$ no tanque.

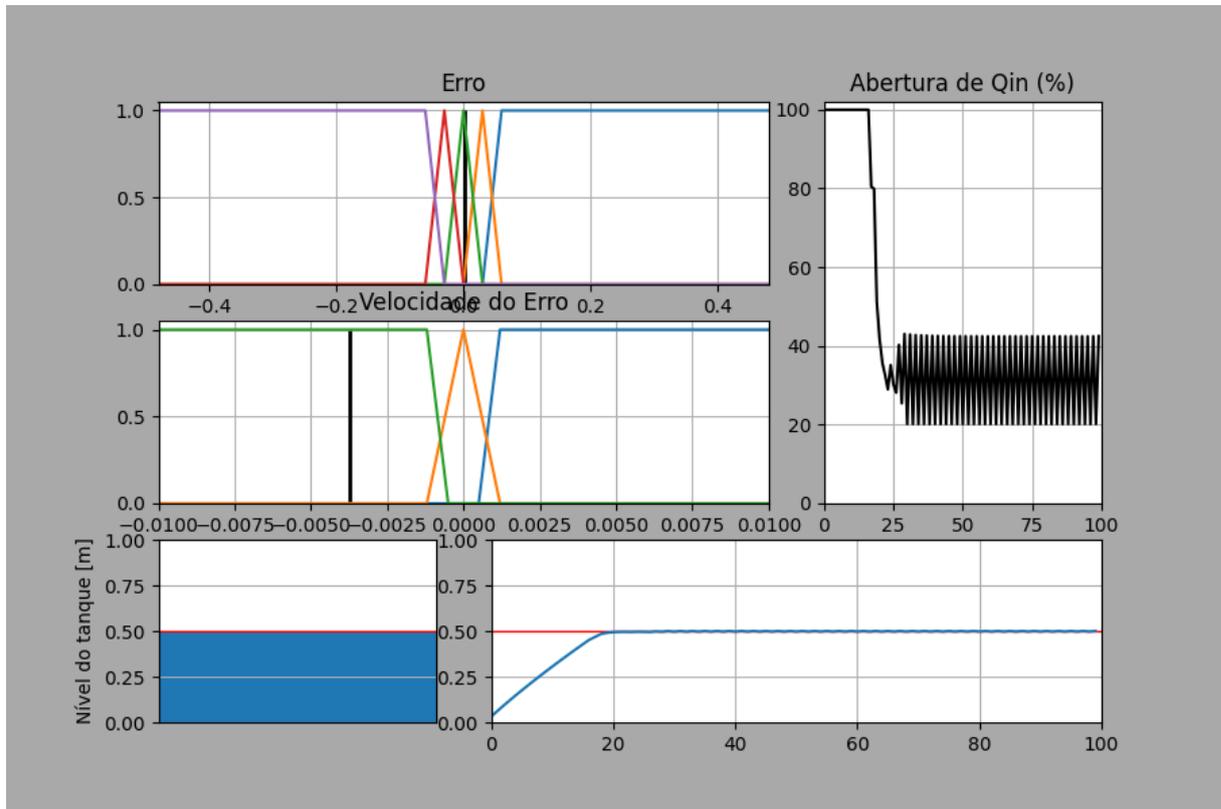
Considerando as dimensões do erro (figura 23a), onde a função de erro zero varia de $-3cm$ à $3cm$, a amostragem utilizada para essa simulação pode ser considerada baixa. Para obtenção de uma resposta mais suave do controlador seria necessário aumentar a taxa de amostragem para o sistema, de forma que não ocorresse grandes "saltos" entre as regras formuladas.

Sendo a variação máxima de altura de $1cm$ entre amostras e as funções de pertinência com precisão de $3cm$, em um intervalo de apenas três amostras o controlador pode mudar completamente sua resposta, causando efeitos negativos e instabilidades. Assim, é justificável o aumento da taxa de amostragem até o ponto em que seu valor seja pelo menos dez vezes menor do que a precisão requerida pelo sistema.

Utilizando este caso como exemplo, se a precisão do sistema é de $3cm$ isso significa que a amostragem mínima recomendada deve ser capaz de perceber mudanças de 0.3 em 0.3 centímetros ao invés de $1cm$. Podendo o sistema apresentar uma variação máxima de $10cm$ em $1s$, pois é um tanque de formato cúbico com capacidade de 1000 litros e vazão máxima de $100L/s$, a amostragem mínima recomendada seria de $0.03s$.

Com o uso de menores taxas de amostragem as variações no erro se tornam cada vez maiores, isso causa os "saltos" de regras de um controlador Fuzzy. A figura 26 utiliza das mesmas funções e dimensão de tanque da figura 25, porém sua taxa de amostragem é reduzida pela metade, passando de 0.1 para 0.333 segundos.

Figura 26 – Resposta com baixa taxa de amostragem



Fonte: elaborado pelo autor (2022).

Devido a subamostragem do sistema, o controlador Fuzzy varia sua atuação constantemente, pois a cada leitura no nível do tanque seu comportamento é alterado em consequência de uma regra completamente diferente ativada. Esse comportamento também constitui uma desvantagem para o sistema, pois o desgaste mecânico sofrido pela válvula é consideravelmente maior.

7 RESULTADOS

7.1 COMPARAÇÕES

Comparando-se os controladores Fuzzy de Silva et al. (2007), implementado com o auxílio de algoritmos genéticos, e com o controlador prático de Vieira (2017), é observado pela tabela 4 a semelhança e diferença entre as implementações.

Tabela 4 – Comparação entre controladores Fuzzy apresentados

	Silva et al. (2007)	Vieira (2017)	Este trabalho
Ferramenta de Simulação	Matlab/Simulink	Matlab/Simulink	Python
Número de regras geradas	8	25	9
Número de Funções Antecedentes	2	2	2

Fonte: elaborado pelo autor (2021).

A semelhança entre os trabalhos está na utilização de duas funções antecedentes para a construção do controlador Fuzzy, pois para o controle de uma planta com essas características se faz necessário no mínimo dois valores de entrada, um correspondente ao erro atual, e outro correspondente a derivado do erro.

Um dos pontos em que os trabalhos divergem é no número de regras geradas. Este parâmetro está ligado a complexidade do controlador, um grande número de regras implicam em um grande número de funções de pertinência nos antecedentes, pois em grande maioria dos casos é necessário fazer a combinação linear entre todas as funções presentes nas n entradas do sistema.

Outro ponto que diferencia os trabalhos é a ferramenta utilizada para simulação dos controladores. Nos trabalhos similares foi utilizado para a simulação o Matlab/Simulink, que disponibiliza para os usuários pacotes de programas para implementação de controladores, aplicação de lógica Fuzzy e simulação dos resultados obtidos.

Nesse trabalho, por sua vez, é utilizado para a simulação e implementação do controlador Fuzzy a linguagem de programação Python, sendo utilizado pacotes para o cálculo de funções diferenciais, cálculo de sistemas Fuzzy e para plotar o desenvolvimento da resposta ao longo do tempo. Todos os pacotes utilizados são disponibilizados online de forma gratuita pelo gerenciador de pacotes do Python.

7.2 RESULTADOS DAS SIMULAÇÕES

Os resultados das simulações dos controladores Fuzzy tanto para o tanque com apenas entrada de líquido como para a planta com entrada e saída alcançaram as propostas inicialmente elaboradas. As respostas, por meio das funções de pertinência e regras escritas, convergem de forma assintótica para o valor de referência estabelecido.

Em todos os testes e resultados obtidos foi utilizada a linguagem de programação Python. Com o uso do pacote *matplotlib* é possível monitorar amostra a amostra o desenvolvimento da resposta e a mudança no comportamento da resposta, permitindo a visualização de cada regra ativada no decorrer de sua atuação, o que também auxiliou na etapa de sintonização dos parâmetros do controlador.

Como mostrado para o caso do controlador responsável pela planta com apenas vazão de entrada, também foi explorado e exemplificado, com um caso simples, os comportamentos que um controlador Fuzzy pode apresentar. Eles podem ser mais agressivos ao se modelar funções de pertinência com rápidas variações entre si, causando mudanças maiores sobre a abertura da válvula. E também suaves, quando suas funções de pertinência possuem mudança gradual a uma taxa de amostragem suficiente.

8 CONCLUSÃO

Controladores Fuzzy são de compreensão simples, uma vez que a lógica Fuzzy é percebida de forma mais intuitiva pelas pessoas. Ao contrário da lógica Fuzzy, o entendimento de controladores PID requer um conhecimento avançado em física, matemática e a própria teoria de controle.

Utilizando-se da lógica nebulosa, pessoas com pouco ou nenhum conhecimento em equacionar o comportamento de sistemas reais são capazes de imaginar como um controlador deve atuar sobre uma planta através das variáveis linguísticas. E podem até mesmo ser capazes de desenvolver um controlador para uma aplicação simples sem a necessidade de um grande conhecimento sobre a teoria de controle.

Para os dois controladores programados a resposta apresentada foi condizente com sua implementação, é possível desenvolver tanto um controlador PID como um Fuzzy que apresentem comportamento abrupto à mudança de referência. Sendo o oposto também verdadeiro, seu comportamento pode ser suave, sendo ele dependente apenas de sua implementação.

Uma das vantagens abordadas sobre o controlador Fuzzy é a capacidade de correção de seu comportamento em faixas específicas da resposta. Se o comportamento durante os testes não for o desejável em determinada área da curva de resposta, ainda é possível corrigir o controlador fazendo alterações apenas nas regras e funções de pertinência responsáveis pela resposta naquela faixa, sem a necessidade de recalcular todo o controlador.

Outro problema solucionado por lógica Fuzzy é sua aplicação em tarefas de tomada de decisão, onde as variáveis atuantes sobre o sistema não são termos exatos. Essas variáveis são conhecidas por operadores humanos treinados, que sabem qual ação tomar nos mais diversos cenários apresentados pela planta. Assim, uma vantagem apresentada por controladores Fuzzy é sua capacidade de capturar as estratégias de controle desses operadores (SIMÕES; SHAW, 2004).

Um ponto negativo que pode ser abordado é o tempo gasto para fazer a sintonia do controlador Fuzzy. A realização desta etapa é feita por meio de tentativa e erro através de simulações consecutivas. Apesar de ser inicialmente obtido um controlador que se aproxime dos valores de referência, o ajuste fino dos parâmetros é a parte mais custosa do projeto. Isso ocorre devido à necessidade de reajuste na forma e/ou tamanho das funções de pertinência dos antecedentes e consequentes até um nível de precisão satisfatório para a aplicação.

Existem formas automatizadas de se fazer a sintonia de um controlador Fuzzy sem a necessidade de um operador humano avaliar e corrigir a resposta ponto a

ponto. Essas formas são extensões de outras aplicações de inteligência artificial e/ou aprendizado de máquina para a lógica Fuzzy, como por exemplo: *Genetic-Fuzzy* e *Neuro-Fuzzy*.

Na sintonização, a escrita das regras também fazem parte do somatório de tempo gasto. Nessa etapa, para pessoas que desconhecem os fenômenos atuantes sobre a planta ou para sistemas com grande número de variáveis o tempo despendido na busca por regras corretas e na escrita das n regras necessárias para determinada aplicação é elevado.

Apesar disso, como já apontado anteriormente, a programação de comportamentos isolados da resposta ainda é possível, uma vez que esses mesmos comportamentos podem ser manipulados de forma singular no controlador.

Assim como os problemas ocorridos pela subamostragem de um sistema, os problemas gerados por ruídos na malha de controle são similares. Apesar desse problema não ser abordado nesse trabalho, controladores Fuzzy para aplicações práticas precisam ser implementados de forma a serem mais robustos a ruídos. Devido aos picos de leitura realizados pelos sensores há a necessidade de funções de pertinência e amostragem bem calibradas, além de formas extras de se tratar esse tipo de interferência.

8.1 TRABALHOS FUTUROS

8.1.1 Genetic-fuzzy

A sintonização do controlador Fuzzy é essencial para apresentar uma boa resposta em uma aplicação, contudo dependendo da complexidade do sistema e da necessidade de se criar mais antecedentes e funções de pertinência esse processo se torna demorado. Por causa dessa exigência é utilizada outras formas de sintonização do controlador, como, por exemplo, aliada a algoritmos genéticos.

Como visto em (SILVA et al., 2007) a aplicação de algoritmos genéticos na sintonização do controlador Fuzzy tem como objetivo melhorar sua resposta, tornando-a mais rápida e precisa. O processo de ajuste fino desempenhado por algoritmos genéticos ocorre de forma natural, uma vez que, nesse método, é criada uma população de controladores e os que demonstram os melhores resultados são selecionados para gerações futuras.

Assim, para desenvolvimento futuro, é possível aliar a construção de um controlador Fuzzy com algoritmos genéticos com o intuito aprimorar a qualidade da resposta e terceirizar o tempo exaurido pela sintonização manual para o processador.

8.1.2 Aplicações práticas

Assim como realizado em (VIEIRA, 2017), a aplicação prática do controlador Fuzzy traz novos desafios para sua sintonização. Após o término das simulações e validação da resposta esse controlador é colocado para atuar no ambiente real, onde existem imprecisões e perturbações decorrentes dos equipamentos utilizados e do ambiente em que o controlador está inserido.

Um dos desafios encontrados por tal aplicação é o ruído, que atua constantemente e prejudica a tomada de decisão dos controladores Fuzzy. Sendo gerado, como por exemplo, na leitura de sensores sua natureza é imprevisível e seu impacto é causado quando ocorre a realimentação na malha fechada de controle.

Devido à ocorrência de picos na leitura de sensores e realimentação desses valores em malha fechada a sintonização Fuzzy precisa ser mais robusta. Ela deve considerar esses novos desafios e ser programada de forma a não ativar outra regra de comportamento oposto por causa da leitura de um pico advindo de um sensor.

Assim, é proposto como trabalho futuro meios para diminuir o impacto causado por ruídos em controladores Fuzzy, como por exemplo com a adição de filtros na sua malha fechada. Aliado aos filtros, a programação do controlador Fuzzy com maior robustez durante a tomada de decisão também mitigam os efeitos negativos dos ruídos.

REFERÊNCIAS

COPPIN, B. **Artificial intelligence illuminated**. Boston: Jones And Barlett Publishers, 2004.

COX, E. **The fuzzy systems handbook: a practitioner's guide to building, using, and maintaining fuzzy systems**. New York: AP Professional, 1994.

DORF, R. C.; BISHOP, R. H. **Sistemas de controle modernos**. 8. ed. Rio de Janeiro: LTC, 2001. 680 p.

GHM GROUP. **6 exemplos de aplicações do sensor de nível industrial**. 2018. Disponível em: <https://www.ghm-group.de/pt/noticia/publicacoes/6-exemplos-de-aplicacoes-do-sensor-de-nivel-industrial/>. Acesso em: 05 abr. 2021.

GOMIDE, F. A. C.; GUDWIN, R. R. Modelagem, controle, sistemas e lógica fuzzy. **SBA controle & Automação**, v. 4, n. 3, p. 97–115, 1994.

NISE, N. S. **Engenharia de sistemas de controle**. 6. ed. Rio de Janeiro: LTC, 2013. 1285 p.

RIGNEL, G. S. D.; CHENCI, P. G.; LUCAS, A. C. Uma introdução a lógica fuzzy. **Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica**, v. 1, n. 1, p. 17–28, 2011.

ROSS, T. J. **Fuzzy logic with engineering applications**. 3. ed. Hoboken, NJ: John Wiley & Sons, 2010. 600 p.

SILVA, P. R. A. et al. Controle de nível em tanques acoplados usando sistemas inteligentes. In: **I SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA COMPUTACIONAL**. Florianópolis, SC: SBRN, 2007. p. 1–6. Disponível em: http://abrimcom.org.br/eventos/sbic_2007/60100023-2/. Acesso em: 22 abr. 2021.

SIMÕES, G. M.; SHAW, S. I. **Controle e modelagem Fuzzy**. 2. ed. São Paulo: Blucher, 2004.

SUGENO, M. An introductory survey of fuzzy control. **Informatics and Computer Science Intelligent Systems Applications**, v. 36, p. 59–83, 1985.

TANSCHKEIT, R. **Sistemas fuzzy**. 2004. Apostila. Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro. Disponível em: <http://www.inf.ufsc.br/~mauro.roisenberg/ine5377/leituras/ICA-Sistemas\Fuzzy.pdf>. Acesso em: 22 abr. 2021.

VIEIRA, F. B. **ESTUDO COMPARATIVO ENTRE CONTROLADORES FUZZY E PI PARA UM SISTEMA DE TANQUE**. 79 p. Programa de Pós-Graduação em Sistemas de Comunicação e Automação — UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO, Mossoró, 2017.