



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

GUILHERME BEZERRA DE FIGUEIRÊDO NETO

**IDENTIFICAÇÃO NEURO-FUZZY APLICADA A UM SISTEMA DE NÍVEL DE
LÍQUIDO**

FORTALEZA

2021

GUILHERME BEZERRA DE FIGUEIRÊDO NETO

IDENTIFICAÇÃO NEURO-FUZZY APLICADA A UM SISTEMA DE NÍVEL DE LÍQUIDO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Wilkley Bezerra
Correia

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

F49i Figueirêdo Neto, Guilherme Bezerra de.
Identificação neuro-fuzzy aplicada a um sistema de nível de líquido / Guilherme Bezerra de Figueirêdo Neto. – 2021.
54 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Elétrica, Fortaleza, 2021.
Orientação: Prof. Dr. Wilkley Bezerra Correia.

1. ANFIS . 2. Identificação de sistemas. 3. Simulação. I. Título.

CDD 621.3

GUILHERME BEZERRA DE FIGUEIRÊDO NETO

IDENTIFICAÇÃO NEURO-FUZZY APLICADA A UM SISTEMA DE NÍVEL DE LÍQUIDO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Elétrica.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Wilkley Bezerra Correia (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Me. Rômulo Nunes de Carvalho Almeida
Universidade Federal do Ceará (UFC)

Prof. Dr. Victor de Paula Brandão Aguiar
Universidade Federal Rural do Semi-Árido
(UFERSA)

A Deus

À minha família

Aos meus amigos

A todos aqueles que fizeram parte da minha
jornada

AGRADECIMENTOS

Aos meus pais Marcos Teixeira e Karina Aragão, ao meu irmão Nicolás Bezerra, e à minha irmã Nicole Bezerra, pelo apoio e companhia durante a minha formação e minha vida. Também aos meus avós, meus tios e primos, presenças sempre constantes na minha vida.

Ao Prof. Dr. Wilkley Bezerra pelo orientação sempre presente e atenciosa ao longo desses anos, desde a iniciação científica até o trabalho de conclusão de curso, meu muito obrigado.

Ao meu querido amigo Bruno Vasconcelos, por estar sempre junto comigo e me ouvir, bem como por sempre me fazer rir, mesmo quando eu estava para baixo.

Ao meu grande amigo Leonardo Milfont, por tomar café incontáveis vezes comigo, por ser colega de laboratório e por me ouvir quando estava estressado. Também ao meu amigo Caio Braga, sempre presente nos cafezinhos.

Ao PET Engenharia Elétrica, por ter servido de casa dentro da UFC e onde fiz grandes amigos, Dimas Macedo, Melissa Viana, Rômulo Iorio, Daniel Morita, Pedro Tino, Gabriel Marçal, Pedro Victor e Yago Vaz.

Aos amigos de graduação Fernando Vasconcelos, Nathália Comberstais, Stephany Aderaldo, Airton Borges, Davi Gomes, Iury Amorim, Filipe Diógenes e Rafael Pereira pela companhia durante o curso.

Aos meus amigos do Santa Cecília Diego Agra, Mateus Arcoverde, Ian Carvalho, Gustavo Gentil, Rafaela Almeida, João Victor Tôrres, Felipe Pinheiro e Mateus Albuquerque por estarem comigo desde antes da faculdade e por permanecermos amigos até hoje.

"Podemos sempre mais do que imaginamos"

(Madre Agathe Verhelle)

RESUMO

Nem sempre é fácil ou mesmo possível modelar sistemas matematicamente, tornando sua análise difícil e motivando o desenvolvimento de técnicas de modelagem diferentes. Nesse contexto surge a área de identificação de sistemas, provendo técnicas para a modelagem de sistemas que não envolvam a modelagem direta. Algoritmos de identificação usam dados de entrada e saída da planta a ser modelada para ajustar os parâmetros do modelo resultante. O presente trabalho visa a identificar um sistema de tanques simulado baseado numa planta real por meio do algoritmo ANFIS, o qual possui características tanto de redes neurais artificiais quanto de sistemas *fuzzy*. Sua estrutura em rede permite a aplicação do algoritmo de retropropagação para treinamento, gerando um sistema *fuzzy*. O modelo ANFIS é não-linear, sendo adequado para modelar sistemas para os quais modelos lineares não são apropriados. Primeiramente, a planta foi modelada matematicamente. Após a modelagem, foi feita a identificação do sistema por meio do ANFIS. Para isso, foi gerado um sinal aleatório que foi subsequentemente aplicado na planta simulada, gerando dados de saída. Através da coleta de tais dados, o conjunto de dados de entrada e saída foi utilizado pelo algoritmo ANFIS para gerar um sistema *fuzzy* que apresentasse as características adequadas. Novas entradas foram aplicadas à planta simulada e ao modelo ANFIS, e os resultados indicaram que o modelo identificado foi capaz de captar bem tanto a dinâmica do sistema quanto seu regime permanente. Por fim, foi aplicada uma entrada no modelo ANFIS e no modelo matemático, verificando-se que seu regime transitório era próximo, mas que havia uma certa diferença entre seus valores de regime permanente, indicando uma superioridade do modelo ANFIS identificado.

Palavras-chave: ANFIS. Identificação de sistemas. Simulação.

ABSTRACT

It is not always easy to model systems mathematically, making their analysis difficult and motivating the development of different modelling techniques. In this context the area of system identification appears, providing techniques to identify systems which do not involve straightforward modelling. Identification algorithms use input and output data from the plant to be modelled to adjust the parameters of the resulting model. This work aims to identify a tank system simulated based on a real plant by using the ANFIS algorithm, which possesses characteristics both from artificial neural networks and from fuzzy systems. Its network structure allows the application of the backpropagation algorithm for training, generating a fuzzy system. The ANFIS model is non-linear, being adequate to model systems for which linear models are not appropriate. Firstly, the plant was modelled mathematically. After modelling it, ANFIS was used to identify the system. In order to do this, a random signal was generated and subsequently applied to the simulated plant, generating input-output data. By collecting these data, they were used by the ANFIS algorithm to generate a fuzzy system which had the desired characteristics. New input signals were applied to the simulated plant and to the ANFIS model, and the results showed that the identified model was able to capture both the system's dynamic and its stable state. In the end, an input signal was applied to the ANFIS model and to the mathematical model, showing that their dynamic behaviour was close, but that their stable states had different values, indicating the superiority of the identified ANFIS model.

Keywords: ANFIS. System identification. Simulation.

LISTA DE FIGURAS

Figura 1 – Representação de um neurônio artificial	21
Figura 2 – Gráfico da função sinal	22
Figura 3 – Gráfico da função logística	23
Figura 4 – Gráfico da função tangente hiperbólica	23
Figura 5 – Perceptron simples para mais de uma classe	25
Figura 6 – Perceptron multicamadas	26
Figura 7 – Controlador <i>Fuzzy</i>	30
Figura 8 – Funções de pertinência para os conjuntos alto e baixo	31
Figura 9 – Função de pertinência triangular	32
Figura 10 – Função de pertinência trapezoidal	32
Figura 11 – Função de pertinência gaussiana	33
Figura 12 – Interseção de duas funções de pertinência triangulares	34
Figura 13 – União de duas funções de pertinência triangulares	35
Figura 14 – Arquitetura ANFIS	37
Figura 15 – Sistema de tanques	41
Figura 16 – Seção de um cone	43
Figura 17 – Gráfico $q_i \times \omega$	45
Figura 18 – Gráfico $q_o \times \sqrt{\gamma h}$	46
Figura 19 – Identificação do sistema	47
Figura 20 – Autocorrelação do sinal de entrada	48
Figura 21 – Sinais de saída da planta e do modelo ANFIS para entrada em degrau	49
Figura 22 – Sinais de saída da planta e do modelo ANFIS para entrada em degrau ampliadas	49
Figura 23 – Sinal de entrada senoidal	50
Figura 24 – Sinais de saída da planta e do modelo ANFIS para entrada senoidal	50
Figura 25 – Sinais de saída da planta e do modelo ANFIS para entrada senoidal ampliadas	51
Figura 26 – Comparativo de modelo matemático com modelo ANFIS identificado	52

LISTA DE ABREVIATURAS E SIGLAS

A/D	analógico/digital
ANFIS	<i>Adaptive Network-based Fuzzy Inference Systems</i>
ARMAX	modelo autoregressivo com média móvel e entradas exógenas
ARX	modelo autoregressivo com entradas exógenas
CLP	Controlador lógico programável
D/A	digital/analógico
Lamotriz	Laboratório de Sistemas Motrizes
rpm	rotações por minuto
UFC	Universidade Federal do Ceará

LISTA DE SÍMBOLOS

y_k	Saída do k -ésimo neurônio
ω_{kj}	Peso sináptico da j -ésima sinapse do k -ésimo neurônio
ω_{0k}	Bias do k -ésimo neurônio
$\varphi(\cdot)$	Função de ativação do neurônio
x_j	j -ésima entrada do neurônio
y_k	Saída do k -ésimo neurônio
v	Entrada da função de ativação
$\mathbf{x}(n)$	Vetor de entradas do neurônio na n -ésima iteração
$\mathbf{w}(n)$	Vetor de pesos sinápticos do neurônio na n -ésima iteração
$y(n)$	Saída do neurônio na n -ésima iteração
$d(n)$	Saída desejada do neurônio
η	Parâmetro de aprendizado da rede neural
$e_j(n)$	Sinal de erro do j -ésimo neurônio na n -ésima iteração
$d_j(n)$	Saída desejada do j -ésimo neurônio
$y_j(n)$	Saída do j -ésimo neurônio na n -ésima iteração
$E(n)$	Erro quadrático na n -ésima iteração
$v_j(n)$	Entrada da função de ativação do j -ésimo neurônio na n -ésima iteração
ω_{ji}^l	Peso sináptico da conexão entre o i -ésimo neurônio da $l - 1$ -ésima camada com o j -ésimo neurônio da l -ésima camada
$\delta_j(n)$	Gradiente local do j -ésimo neurônio
$\mu_{A_1}(\cdot)$	Função de pertinência ao conjunto A_1
$\mu_{A_1'}(\cdot)$	Função de pertinência ao complemento de A_1
$\mu_{A_1 \cap A_2}(\cdot)$	Função de pertinência à interseção entre A_1 e A_2
$\mu_{A_1 \cup A_2}(\cdot)$	Função de pertinência à união entre A_1 e A_2
o	Saída da defuzzificação por centro de gravidade
b_i	Centro da função de pertinência do consequente da regra i
$\int \mu(i)$	Área sob a função de pertinência $\mu(i)$

w_i	i -ésimo peso da rede ANFIS
\bar{w}_i	i -ésimo peso normalizado da rede ANFIS
O_i^k	i -ésima saída da k -ésima camada do ANFIS
f_i	Função linear de saída do ANFIS
A	Matriz de dados de entrada
θ	Matriz de parâmetros da equação linear
B	Matriz de dados de saída
θ^*	Matriz de estimativa de θ
A^T	Matriz A transposta
$\#(k)$	Número de elementos da camada k do ANFIS
L	Número de elementos da última camada do ANFIS
E_p	Erro quadrático do p -ésimo elemento de treino
P	Número de elementos do vetor de treinamento
$T_{m,p}$	m -ésima saída do p -ésimo elemento de treino
$O_{m,p}^L$	m -ésima saída da última camada do ANFIS referente ao p -ésimo elemento de treino
α	Parâmetro da rede ANFIS
V	Volume de líquido no tanque em m
q_i	Vazão de entrada no tanque em m^3/s
q_o	Vazão de saída do tanque em m^3/s
h	Altura do tronco de cone em m
R	Raio da base maior do tanque em m
r	Raio da base menor do tanque em m
t	Tempo em s
ω	Velocidade angular em rpm
k_i	Constante que relaciona a velocidade angular da motobomba com a sua vazão
ΔP	Diferença de pressão entre as extremidades da tubulação em Pa

k_o	Constante de resistência hidráulica na tubulação de descida
P_{atm}	Pressão atmosférica
γ	Peso específico da água em N/m^3
$y[k]$	Saída do modelo ANFIS no instante discreto k
$u[k - 1]$	Entrada do modelo ANFIS no instante discreto $k - 1$

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Motivação	16
1.2	Justificativa	17
1.3	Objetivos	17
<i>1.3.1</i>	<i>Objetivos específicos</i>	17
1.4	Revisão bibliográfica	18
1.5	Organização dos capítulos	18
2	SISTEMA NEURO-FUZZY	20
2.1	Redes Neurais Artificiais	20
<i>2.1.1</i>	<i>Neurônio Artificial</i>	21
<i>2.1.2</i>	<i>Perceptron Simples</i>	24
<i>2.1.3</i>	<i>Perceptron Multicamadas</i>	26
<i>2.1.3.1</i>	<i>Algoritmo de retropropagação</i>	26
2.2	Sistemas de Inferência Fuzzy	29
<i>2.2.1</i>	<i>Lógica Fuzzy</i>	30
<i>2.2.2</i>	<i>Funções de Pertinência</i>	31
<i>2.2.2.1</i>	<i>Operações lógicas</i>	33
<i>2.2.3</i>	<i>Variáveis linguísticas e regras</i>	34
<i>2.2.4</i>	<i>Defuzzificação</i>	36
2.3	ANFIS	36
<i>2.3.1</i>	<i>Arquitetura ANFIS</i>	37
<i>2.3.2</i>	<i>Aprendizado ANFIS</i>	38
<i>2.3.2.1</i>	<i>Estimação por mínimos quadrados</i>	38
<i>2.3.2.2</i>	<i>Algoritmo de retropropagação</i>	38
3	IDENTIFICAÇÃO ANFIS	40
3.1	Identificação de Sistemas Dinâmicos	40
3.2	Descrição da bancada e modelagem matemática	41
3.3	Papel do ANFIS	46
4	RESULTADOS E APLICAÇÕES	47
4.1	Comparação entre a modelagem matemática e o modelo ANFIS	51

5	CONCLUSÕES E TRABALHOS FUTUROS	53
	REFERÊNCIAS	54

1 INTRODUÇÃO

No contexto da engenharia elétrica, pode-se elencar como problema a modelagem de determinados sistemas devido a sua complexidade. Como solução, surge a área de identificação de sistemas, a qual é uma aplicação da teoria de sistemas lineares e processamento de sinais. Ela auxilia na caracterização de sistemas os quais são difíceis de modelar.

1.1 Motivação

O estudo da área de identificação de sistemas serve como base para a caracterização de plantas e subsequente projeto de controladores para essas plantas. Plantas muito complexas podem ter uma modelagem matemática inviável, tornando necessária a sua identificação a partir de dados de entrada e saída coletados. Assim, os modelos e algoritmos de identificação são ferramentas muito úteis.

Uma área do conhecimento que vem sendo desenvolvida mais intensamente nos últimos anos é a de inteligência artificial. Apesar do receio que evoca em algumas pessoas, a verdade é que a inteligência artificial é muito benéfica para a humanidade, sendo aplicada em diversos campos dos conhecimento, tais como na detecção de doenças, nos sistemas de recomendação encontrados em *sites* de compras e em aplicações financeiras (AL-SHAYEA, 2011; ISINKAYE *et al.*, 2015; BAHRAMMIRZAEI, 2010).

O número cada vez maior de aplicações de inteligência artificial serviu de motivação para um estudo de duas subáreas desta área: redes neurais artificiais e lógica *fuzzy* (também chamada de lógica nebulosa). Para além destas duas subáreas, foi proposto em Jang (1993) uma junção das duas abordagens, chamada *Adaptive Network-based Fuzzy Inference Systems* (ANFIS). O conceito do ANFIS é usar uma arquitetura em rede e uma lei de aprendizado (características presentes em redes neurais) num sistema de inferência *fuzzy*. Desta forma, ele traz as vantagens combinadas dos dois métodos, como a lei de aprendizado e uma grande flexibilidade.

O modelo ANFIS, bem como outros de inteligência artificial, pode ser usado no processamento de sinais. Várias aplicações de controle inteligente já foram demonstradas, como destacam Antsaklis e Passino (1993). Assim, surge a ideia de aplicar o ANFIS para a identificação de um sistema.

A maior vantagem ao usar o ANFIS como modelo para uma identificação é o fato de ser um modelo não-linear, podendo incorporar características não-lineares que modelos lineares,

tais como modelos determinísticos de primeira e segunda ordem ou o estimador de mínimos quadrados.

Diversos processos industriais envolvem o armazenamento de líquidos, tais como processamento de alimentos, tratamento de efluentes e geração de energia nuclear (PAN *et al.*, 2005). Levando isso em consideração, a motivação do presente trabalho foi fazer a identificação em modelo ANFIS de um processo de armazenamento de líquido. A vantagem de se utilizar tal modelo é sua maior proximidade com modelos reais, devido à característica não-linear do ANFIS, melhorando o projeto de controladores.

1.2 Justificativa

No Laboratório de Sistemas Motrizes (Lamotriz) da Universidade Federal do Ceará (UFC) há diversas plantas de caráter industrial, tais como esteiras, exaustores e um sistema de armazenamento de líquido. Frente à necessidade de melhor entender a dinâmica deste último, técnicas de identificação de sistemas podem ser usadas. Tendo em vista a crescente utilização de técnicas de inteligência artificial na indústria, sendo chamada de indústria 4.0, optou-se por usar o ANFIS com o intuito de fazer tal identificação, permitindo a trabalhos futuros um melhor estudo da planta em questão.

1.3 Objetivos

Os objetivos do trabalho são gerar um modelo identificado da planta que sirva como meio para estudos mais aprofundados sobre ela, podendo ser utilizada na avaliação de controladores de diferentes tipos num sistema de tanques, bem como de avaliar a eficácia da topologia ANFIS na identificação de sistemas dinâmicos.

1.3.1 *Objetivos específicos*

- Modelar o sistema de tanques;
- Identificar o sistema de tanques em simulação usando o ANFIS;
- Verificar a qualidade da identificação;
- Comparar o modelo matemático da planta com o modelo ANFIS identificado.

1.4 Revisão bibliográfica

Alguns trabalhos serviram de base para o presente trabalho, estando destacados nesta seção. Alguns outros relacionados, expandindo o exposto aqui, também são relacionados.

Primeiramente, Jang (1993) propõe a estrutura ANFIS, aplicando algoritmos de controle já conhecidos como retropropagação e estimação de mínimos quadrados para treinar a rede. Mostra também exemplos que atestam as capacidades do ANFIS. Jang e Chuen-Tsai Sun (1995) expande o trabalho anterior, detalhando o ANFIS e mostrando maneiras de usá-lo em controle e identificação. Por fim, Jang (1996) usa o ANFIS para identificar um secador de cabelo e o compara com um modelo autoregressivo com entradas exógenas (ARX) identificado.

Pode-se citar outros trabalhos que exploram o uso do ANFIS:

- Zhixiang Hou *et al.* (2003) mostra o uso do ANFIS para identificação de sistemas não-lineares;
- Daneshwar e Noh (2013) usa o ANFIS na identificação de válvulas de controle inteligentes com fricção estática;
- Muslim e Nusantoro (2016) usa um modelo autoregressivo com média móvel e entradas exógenas (ARMAX) para fazer a identificação ANFIS dum destilador a vácuo.

Outros trabalhos modificam alguma característica do ANFIS, seja sua estrutura, seja o algoritmo usado no treinamento. Haznedar e Kalinli (2018) propõe o uso do algoritmo de recozimento simulado para o treino da rede em detrimento do algoritmo de retropropagação, usado originalmente em Jang (1993), para a identificação de sistemas não-lineares, apresentando resultados melhores. Por outro lado, Martins e Araújo (2015) apresentam uma estrutura modificada do ANFIS e a valida por meio da identificação de um sistema de tanques.

1.5 Organização dos capítulos

O presente trabalho possui cinco capítulos, os quais são divididos da seguinte maneira:

- Capítulo 1: é o capítulo atual, o qual introduz o trabalho, apontando sua motivação, sua justificativa e seus objetivos, bem como a organização dos capítulos;
- Capítulo 2: este capítulo serve de fundamentação teórica, introduzindo o conceito de sistema neuro-fuzzy. Ademais, explica o conceito de redes neurais até o funcionamento de um perceptron multicamadas, além de sistemas fuzzy e, por fim, o sistema ANFIS, o qual

é a junção dos conceitos de redes neurais com sistemas fuzzy.

- Capítulo 3: este capítulo trata da modelagem da planta, bem como de conceitos relacionados à identificação de sistemas.
- Capítulo 4: este capítulo mostra os resultados referentes à identificação ANFIS do sistema. Os resultados são analisados para que se possa validar ou rejeitar o modelo identificado. Por fim, a modelagem matemática da planta é comparada com o modelo ANFIS identificado.
- Capítulo 5: este capítulo é a conclusão do trabalho. Discute também acerca de possíveis trabalhos futuros.

2 SISTEMA NEURO-FUZZY

Sistemas neuro-fuzzy são sistemas que misturam as capacidades de inferência de sistemas *fuzzy* com características vindas da teoria de redes neurais artificiais, tais como uma arquitetura em camadas e a noção de aprendizado com base em dados. O capítulo 2 está dividido em três partes. A seção 2.1 trata de redes neurais, passando pelo funcionamento de um neurônio artificial, arquitetura de uma rede neural simples, regra de aprendizado, arquitetura multicamadas e retropropagação. A seção 2.2 trata de sistemas de inferência *fuzzy*, explicitando o conceito de funções de pertinência e como é feita uma inferência *fuzzy*. Por fim, a seção 2.3 traz o conceito de sistema de inferência *fuzzy* adaptativo baseado em redes, ou ANFIS no acrônimo em inglês. Essa seção é baseada principalmente no trabalho de Jang (1993).

2.1 Redes Neurais Artificiais

O cérebro humano é uma máquina altamente complexa, capaz de realizar tarefas difíceis aparentemente sem dificuldade, tais como reconhecimento de imagens e voz. As vantagens do cérebro humano são seu processamento altamente complexo, não linear e a capacidade de processamento paralelo e de aprendizado (HAYKIN, 1998). Redes neurais artificiais são uma tentativa de reproduzir o processamento de informações realizado pelo cérebro humano (redes neurais biológicas) de forma artificial, utilizando-se de estratégias similares. Da mesma forma como um cérebro humano é formado de diversos neurônios, as redes neurais artificiais (chamadas simplesmente de redes neurais a partir desse ponto) são formadas por neurônios artificiais. Esses neurônios artificiais são agrupados em camadas, cada uma com um ou mais neurônios operando de forma paralela. A partir de um conjunto de dados, os neurônios são treinados (aprendizado supervisionado). Essas características das redes neurais serão exploradas nas subseções seguintes. A subseção 2.1.1 trata do funcionamento de um neurônio artificial, discorrendo sobre suas entradas, os pesos associados, a função de ativação e a saída. A subseção 2.1.2 discorre sobre o perceptron simples, uma rede neural de uma camada proposta por Rosenblatt (1958) para o problema de classificação. Em seguida, na subseção 2.1.3, o perceptron multicamadas é apresentado, sendo apresentado também o algoritmo de retropropagação utilizado para treinamento da rede.

2.1.1 Neurônio Artificial

O neurônio é a unidade básica das redes neurais. É ele que realiza o processamento de informações do sistema. É composto de três partes (HAYKIN, 1998):

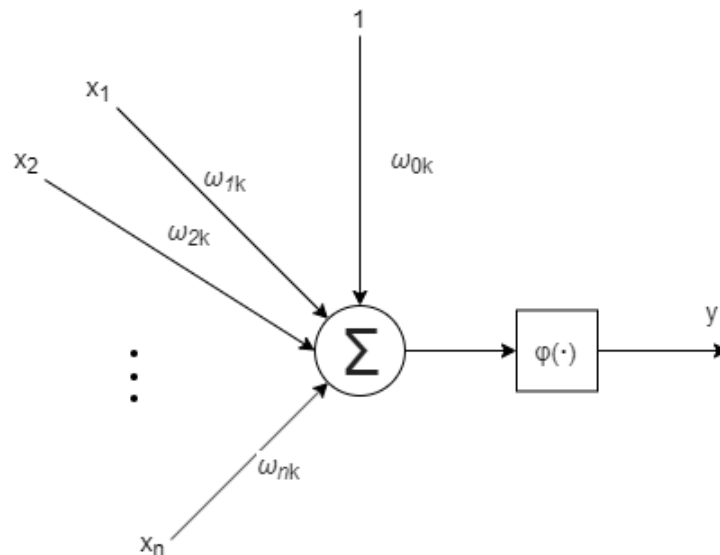
- Um conjunto de n entradas ou sinapses, cada uma com seu peso sináptico ω_{kj} , sendo j a sinapse e k o neurônio em questão. Além das sinapses, o neurônio possui um *bias* ω_{0k} , que não depende das suas entradas;
- Um somador, que soma os sinais de entrada multiplicados por seus pesos e o *bias*;
- Uma função de ativação $\varphi(\cdot)$, que define se o neurônio está ativado ou não. A ativação do neurônio é o que define se o sinal é propagado por ele.

Assim, pode-se escrever a saída de um neurônio matematicamente como:

$$y_k = \varphi\left(\sum_{j=0}^n x_j \omega_{jk}\right) \quad (2.1)$$

onde y_k é a saída do k -ésimo neurônio e x_j a j -ésima entrada do neurônio. A figura 1 mostra a representação de um neurônio.

Figura 1 – Representação de um neurônio artificial



Fonte: Próprio autor

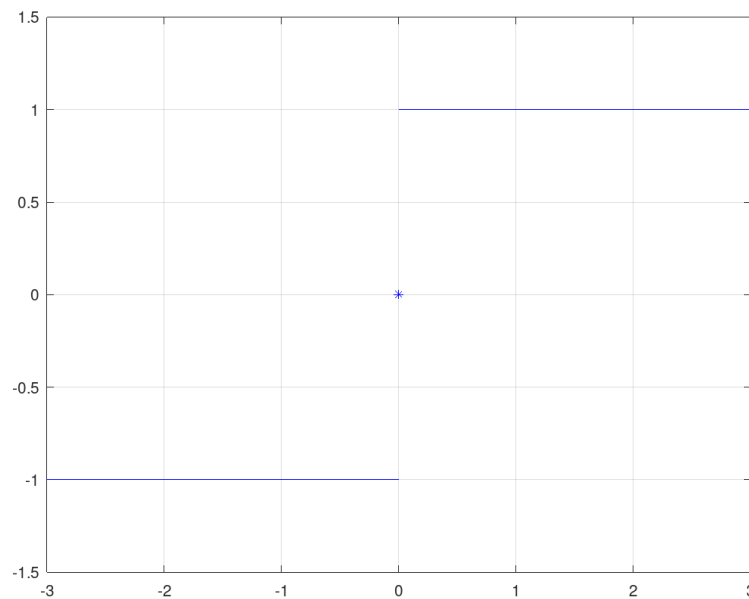
A função de ativação do neurônio pode assumir diversas formas. A função mais simples que pode ser utilizada é a função sinal, que tem como saída o valor 1 se sua entrada for

positiva e -1 se sua entrada for negativa:

$$\text{sgn}(v) = \begin{cases} 1, & \text{se } v > 0 \\ 0, & \text{se } v = 0 \\ -1, & \text{se } v < 0 \end{cases} \quad (2.2)$$

A figura 2 mostra o gráfico da função sinal

Figura 2 – Gráfico da função sinal



Fonte: Próprio autor

A função sinal é muito útil em problemas de classificação, onde um vetor de entrada deve ser classificado como pertencente a um determinado número de classes pré-definidas. Uma desvantagem dessa função é não ser continuamente diferenciável, inviabilizando seu uso em redes com mais de uma camada devido à necessidade de se ter a derivada da função para a aplicação do algoritmo de retropropagação (HAYKIN, 1998). Para contornar esse problema, pode-se utilizar uma função sigmoide, ou seja, que possui formato da letra *s* aproximadamente. Duas dessas funções são:

- Função logística: essa função pode ser escrita como:

$$\varphi(v) = \frac{1}{1 + e^{(-av)}} \quad (2.3)$$

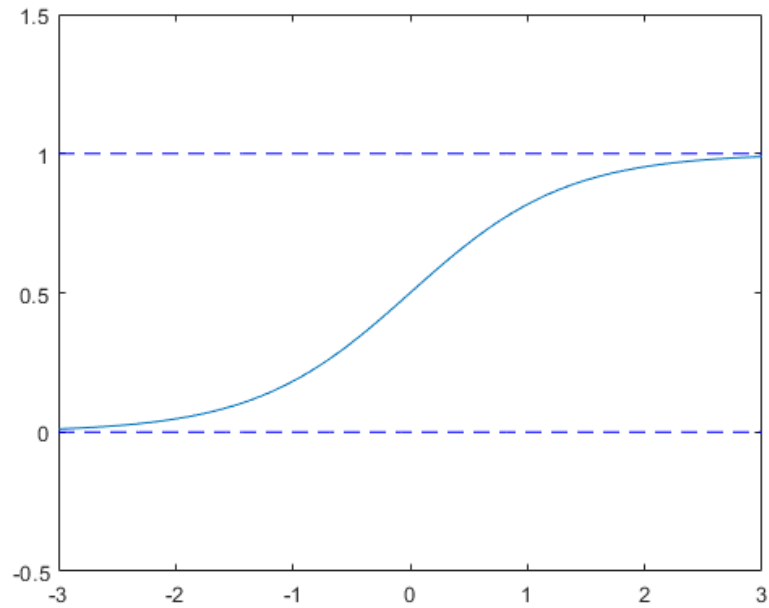
onde v é a entrada da função e a é um parâmetro que determina a inclinação da função sigmoide. A figura 3 mostra o gráfico dessa função para $a = 1,5$

- Função tangente hiperbólica: essa função pode ser escrita como:

$$\varphi(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} \quad (2.4)$$

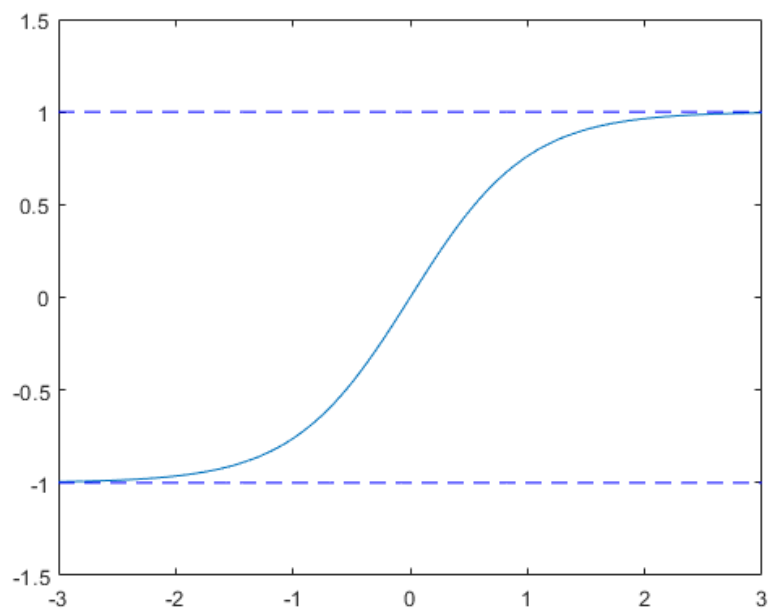
onde v é a entrada da função. A figura 4 mostra o gráfico dessa função.

Figura 3 – Gráfico da função logística



Fonte: Próprio autor

Figura 4 – Gráfico da função tangente hiperbólica



Fonte: Próprio autor

É importante notar que a função logística está limitada entre 0 e 1, enquanto a função tangente hiperbólica está limitada entre -1 e 1.

2.1.2 *Perceptron Simples*

O perceptron simples é uma implementação de uma rede neural de camada única com o objetivo de classificação, desenvolvido por Rosenblatt (1958). Apesar de aplicações de redes neurais em controle e identificação de sistemas serem essencialmente problemas de regressão, o estudo do perceptron simples é interessante para que se possa mostrar a mais simples implementação de uma rede neural. Assim, essa subseção não se propõe a entrar em detalhes sobre esse classificador, mas apenas mostrar sua estrutura básica e a regra delta utilizada no treinamento dessa rede.

Considere primeiramente um problema de classificação contendo duas classes linearmente separáveis (um perceptron simples é um classificador linear, não conseguindo separar classes não linearmente separáveis). A partir de elementos contidos em cada classe, existe a necessidade de classificar novos elementos em alguma das duas classes presentes. Para resolver esse problema, uma rede neural contendo apenas um neurônio pode ser utilizada. Considera-se que a função de ativação de tal neurônio seja a função sinal, classificando um elemento x como pertencente a uma classe A se a saída do neurônio for positiva e como pertencente a uma classe B se a saída do neurônio for negativa. A pergunta que fica é: como ajustar os pesos de tal neurônio para que essa rede consiga classificar com a menor taxa de erro novos elementos? O algoritmo de convergência do perceptron é baseado em dados já disponíveis, sendo um exemplo de aprendizado supervisionado, e pode ser resumido em Haykin (1998).

Primeiramente, deve-se fazer algumas considerações. Considera-se que o vetor de entradas do neurônio é de dimensão $(m + 1)$ por 1, sendo m o número de entradas. Isso ocorre porque considera-se uma entrada virtual $x_0 = 1$ que é multiplicada pelo bias do neurônio. Assim:

$$\mathbf{x}(n) = [1, x_1(n), x_2(n), \dots, x_m(n)]^T \quad (2.5)$$

$$\mathbf{w}(n) = [\omega_0(n), \omega_1(n), \omega_2(n), \dots, \omega_m(n)]^T \quad (2.6)$$

Utiliza-se notação vetorial para facilitar a leitura, sendo as letras em negrito vetores. Deve-se notar a utilização de um índice n , que representa a iteração presente. Por fim, $y(n)$ representa a resposta atual da rede, $d(n)$ representa a resposta desejada e η representa o parâmetro

de aprendizagem da rede, um valor positivo que controla o ajuste aplicado aos pesos sinápticos, e, geralmente, η é considerado constante.

Primeiramente, considera-se que o vetor de pesos é nulo. Depois, deve-se realizar uma sequência de passos iterativamente, mostrando todo o conjunto de dados em um número de épocas pré-definidas. Os passos são:

1. Calcula-se o valor de saída $y(n)$ para o vetor de entrada $\mathbf{x}(n)$:

$$y(n) = \text{sgn}[\mathbf{w}(n)^T \mathbf{x}(n)] \quad (2.7)$$

2. Depois, deve-se atualizar o valor do vetor de pesos, de acordo com a fórmula 2.8:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n) \quad (2.8)$$

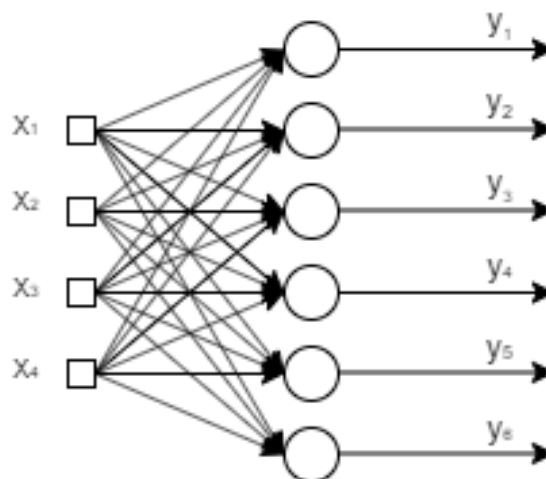
onde $d(n) = 1$ se $\mathbf{x}(n)$ pertence à classe A ou $d(n) = -1$ se $\mathbf{x}(n)$ pertence à classe B.

3. Repetir até esgotarem-se as épocas.

É interessante notar que, se a resposta desejada e a resposta da rede forem iguais, o vetor de pesos não é atualizado.

Se houver mais de uma classe, é possível utilizar um neurônio para cada classe, ou seja, o neurônio que estiver ativado corresponderá à classe a que o elemento apresentado à rede pertencerá. A figura 5 mostra a estrutura de uma rede neural simples com mais de um neurônio:

Figura 5 – Perceptron simples para mais de uma classe



Fonte: Próprio autor

A rede neural da figura 5 possui quatro entradas e classifica algum elemento a uma de seis classes, devendo apenas um neurônio estar ativado por vez.

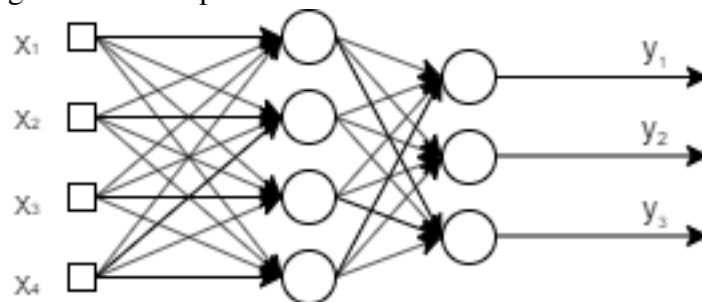
2.1.3 Perceptron Multicamadas

Perceptrons multicamadas são implementações de redes neurais com mais de uma camada. Pode-se diferenciar três tipos de camadas (HAYKIN, 1998):

- Camada de entrada: não é uma camada de neurônios propriamente dita, serve apenas para representar o vetor de entrada;
- Camada oculta: camada intermediária entre a camada de entrada e a camada de saída. Redes neurais multicamadas podem ter mais de uma camada oculta, e essas camadas servem para extrair mais características dos padrões de entrada;
- Camada de saída: última camada de neurônios, resulta na saída da rede neural.

Em problemas de regressão, pode-se utilizar uma função linear como função de ativação dos neurônios da camada de saída, enquanto geralmente se utiliza uma função sigmóide como função de ativação dos neurônios da camada oculta. A função linear nos neurônios de saída é interessante para que os valores de saída não fiquem limitados, enquanto as funções sigmóides são interessantes porque apresentam não-linearidade e porque são funções contínuas, diferentemente da função sinal comumente utilizada em perceptrons simples, permitindo o cálculo de sua derivada. A figura 6 mostra um exemplo de perceptron multicamadas

Figura 6 – Perceptron multicamadas



Fonte: Próprio autor

2.1.3.1 Algoritmo de retropropagação

O algoritmo de retropropagação é o algoritmo pelo qual se treina redes neurais multicamadas, tendo sido proposto por Rumelhart *et al.* (1986). Seu nome deriva do fato de que, para treinar neurônios de camadas ocultas, o erro associado a neurônios de camadas mais externas é propagado no sentido inverso ao da rede. Antes de se explicar o algoritmo, algumas definições são feitas. De acordo com Haykin (1998), o sinal de erro na saída de um neurônio j

na iteração n pode ser escrito como:

$$e_j(n) = d_j(n) - y_j(n) \quad (2.9)$$

onde $d_j(n)$ representa a saída desejada e $y_j(n)$ representa a saída do neurônio. De forma similar, pode-se definir o erro quadrático como:

$$E(n) = \frac{1}{2} \sum_{j=1}^m e_j^2(n) \quad (2.10)$$

Essa função é a função custo considerada, e deve ser minimizada em relação aos pesos sinápticos ω_{ji}^l , sendo esse o peso da conexão entre o neurônio i da camada $l - 1$ com o neurônio j da camada l . Utilizando a regra da cadeia:

$$\frac{\partial E(n)}{\partial \omega_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_i(n)} \frac{\partial y_i(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial \omega_{ji}(n)} \quad (2.11)$$

Fazendo-se as manipulações matemáticas necessárias, chega-se à seguinte fórmula:

$$\frac{\partial E(n)}{\partial \omega_{ji}(n)} = -e_j(n) \phi_j'(v_j(n)) y_i(n) \quad (2.12)$$

A correção dos pesos pode ser feita através de:

$$\omega_{ji}(n+1) = \omega_{ji}(n) - \eta \frac{\partial E(n)}{\partial \omega_{ji}(n)} \quad (2.13)$$

onde η é uma taxa de aprendizado do algoritmo. É interessante definir um gradiente local $\delta_j(n)$ (HAYKIN, 1998):

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} \quad (2.14)$$

$$\delta_j(n) = -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \quad (2.15)$$

$$\delta_j(n) = e_j(n) \phi_j'(v_j(n)) \quad (2.16)$$

Com essa definição de gradiente local (equação 2.16) e a equação 2.12, pode-se reescrever a regra de atualização dos pesos como:

$$\omega_{ji}(n+1) = \omega_{ji}(n) + \eta \delta_j(n) y_i(n) \quad (2.17)$$

De posse da regra de atualização, surgem duas possibilidades:

1. A primeira possibilidade é de que o neurônio em questão esteja na camada de saída. Nesse caso, o erro é conhecido, e a regra de aprendizado é utilizada normalmente.
2. A segunda possibilidade é de que o neurônio em questão esteja numa camada oculta. Nesse caso, não se sabe o valor de erro de sua saída, pois não é sabida a saída desejado. Deve-se utilizar a retropropagação para estimar um valor de saída desejado para o neurônio, e a partir dele utilizar a regra de aprendizado.

O sinal de erro para um neurônio oculto depende do erro de todos os neurônios que vêm adiante na rede. Utiliza-se o índice k para representar os neurônios da camada de saída. O gradiente local para um neurônio, utilizando-se a regra da cadeia na equação 2.14 fica:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial y_j(n)} \frac{y_j(n)}{\partial v_j(n)} \quad (2.18)$$

$$\delta_j(n) = -\frac{\partial E(n)}{\partial y_j(n)} \phi'_j(v_j(n)) \quad (2.19)$$

Tomando a equação 2.10, substituindo os índices j por k para representar a última camada e derivando-a em relação a $y_j(n)$, obtém-se:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \quad (2.20)$$

Utilizando-se a regra da cadeia:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \quad (2.21)$$

As derivadas parciais presentes na equação 2.21 têm seus valores dados por:

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\phi'_k(v_k(n)) \quad (2.22)$$

$$\frac{\partial v_k(n)}{\partial y_j(n)} = \omega_{kj}(n) \quad (2.23)$$

Substituindo as equações 2.22 e 2.23 na equação 2.21 obtém-se:

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_k e_k \phi'_k(v_k(n)) \omega_{kj}(n) \quad (2.24)$$

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_k \delta_k(n) \omega_{kj}(n) \quad (2.25)$$

Usando as equações 2.19 e 2.25 chega-se a:

$$\delta_j(n) = \varphi'_j(v_j) \sum_k \delta_k(n) \omega_{kj}(n) \quad (2.26)$$

Com o gradiente local dos neurônios ocultos dado pela equação 2.26 é possível atualizar os pesos de tais neurônios através da equação 2.17.

2.2 Sistemas de Inferência *Fuzzy*

Sistemas de inferência *fuzzy* são aqueles nos quais lógica *fuzzy*, também chamada de lógica nebulosa em português, é utilizada para a tomada de ações, sejam ações discretas ou ações contínuas. Sistemas *fuzzy* são baseados no conhecimento humano sobre os processos, na forma intuitiva de se representar os conhecimentos sobre um processo. Um exemplo seria o controle de velocidade de um carro. Quem tem o hábito de dirigir sabe intuitivamente como utilizar o acelerador para controlar a velocidade do carro, mesmo que haja perturbações tais como mudança de inclinação da via ou mudança de pavimento. A lógica *fuzzy* permite embarcar os conhecimentos humanos intuitivos em um controlador automático.

A forma como o conhecimento humano pode ser embarcado em sistemas *fuzzy* é através de variáveis linguísticas e de um conjunto de regras. Definem-se regras que atuam diretamente em cima das variáveis linguísticas para se tomar uma decisão acerca da ação a se tomar. No exemplo do carro, é possível definir uma variável linguística de entrada chamada de *velocidade* e uma variável linguística de saída denominada de *acelerador* para controlar o carro, além de se definir regras que unam as duas. Pode-se dizer:

- Se velocidade estiver baixa, então pisa no acelerador;
- Se velocidade estiver média, então acelerador permanece como está;
- Se velocidade estiver alta, então solta o acelerador.

A seção 2.2.3 trata melhor das variáveis linguísticas e regras.

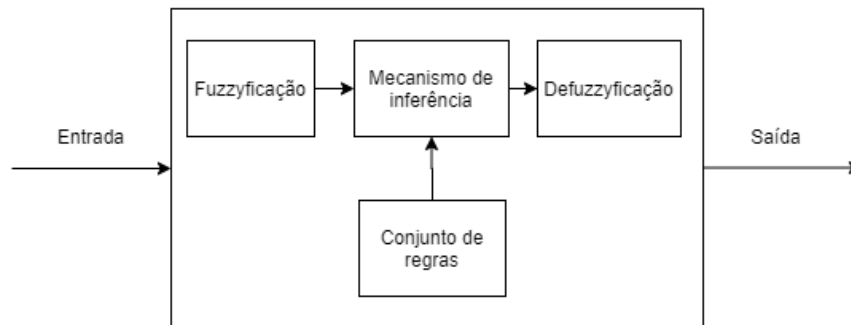
Passino *et al.* (1998) divide controladores *fuzzy* da seguinte forma:

- Base de regras: conjunto de regras Se...Então... que define que ações devem ser tomadas com base no conhecimento embarcado do especialista;
- Interface de fuzzyficação: converte dados de entrada em variáveis linguísticas que podem ser interpretadas pelo mecanismo de inferência para determinar que regras estão ativas;
- Mecanismo de inferência: interpreta as informações linguísticas para gerar ações de controle desejadas;

- Interface de defuzzyficação: converte a saída gerada pelo mecanismo de inferência em um valor numérico que poderá ser mandado para uma planta a ser controlada.

A figura 7 mostra um controlador *fuzzy* e seus constituintes.

Figura 7 – Controlador *Fuzzy*



Fonte: Adaptado de Passino *et al.* (1998)

Esta estrutura também vale para representar sistemas, visto que ele recebe uma entrada e gera uma saída com base num conjunto de regras, as quais podem conter as informações pertinentes ao sistema.

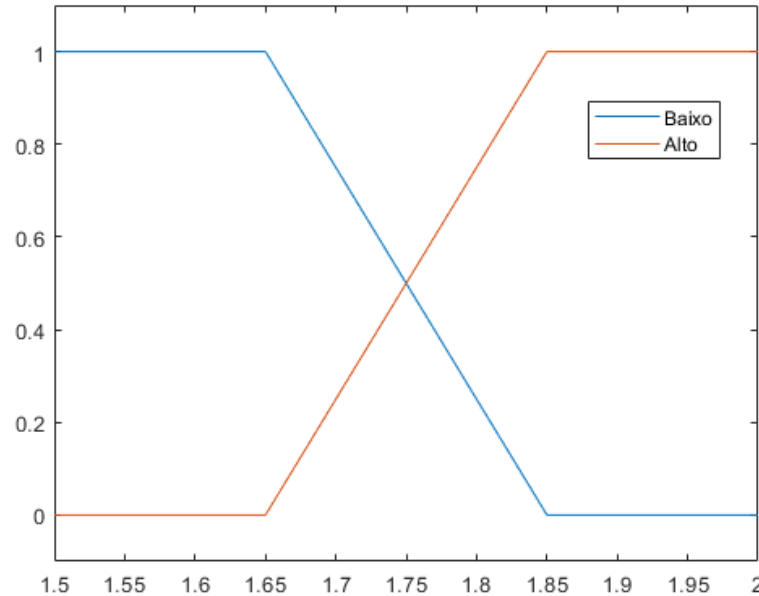
2.2.1 Lógica Fuzzy

A lógica *fuzzy* difere da lógica booleana na medida em que define funções de pertinência para quantificar o grau de pertencimento de determinado valor a um determinado conjunto. Tomando por exemplo o conjunto de todas as pessoas altas e o conjunto de todas as pessoas baixas, de acordo com a lógica booleana, ou alguém pertence ou não pertence a um grupo, surgindo a seguinte pergunta: como traçar um limiar? Se for definido que uma pessoa com menos de 1,80m é baixa, enquanto uma pessoa a partir dessa altura é alta, parece injusto classificar uma pessoa que mede 1,79m metros como baixa e classificar uma pessoa com 1,81m como alta, quando a diferença de altura é apenas 2 cm. Na lógica *fuzzy*, por outro lado, funções de pertinência definem o grau de pertencimento de um elemento a um conjunto. Pode-se definir, por exemplo, que pessoas entre 1,65m e 1,85m possuam tanto um grau de pertencimento ao conjunto de pessoas baixas quanto um grau de pertencimento ao conjunto de pessoas altas. A figura 8 exemplifica duas funções de pertinência, uma para cada conjunto.

É importante ressaltar que a função de pertinência só retorna valores entre 0, denotando não pertencimento ao conjunto, e 1, quando o elemento pertence totalmente ao conjunto. Pode-se imaginar a lógica booleana como tendo funções de pertinência que só admitem os

valores 0 e 1, o que configura a lógica *fuzzy* como uma extensão da lógica booleana.

Figura 8 – Funções de pertinência para os conjuntos alto e baixo



Fonte: Próprio autor

2.2.2 Funções de Pertinência

Como exemplificado na subseção anterior, funções de pertinência podem assumir um formato linear por partes, mas não são limitadas somente a isso. Outras formas possíveis de funções de pertinência são:

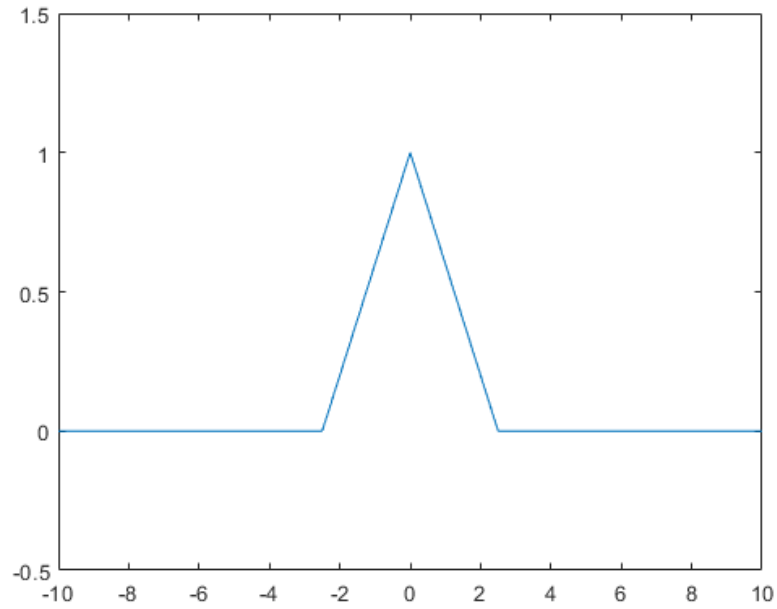
- Triangular: assemelha-se a um triângulo;
- Trapezoidal: parecida com a triangular, possui um topo achatado para englobar uma faixa de valores que pertencem completamente ao conjunto;
- Gaussiana: assemelha-se com a distribuição normal, mas tendo formato de sino. A equação 2.27 representa uma função de pertinência neste formato com base em três parâmetros a , b , e c .

$$\mu_{A_1}(x) = \frac{1}{1 + \left[\left(\frac{x-c}{a} \right)^2 \right]^b} \quad (2.27)$$

É importante notar que o valor gerado por uma função de pertinência não é o mesmo que uma probabilidade. Tal valor apenas indica o grau de certeza de que um elemento pertence a um

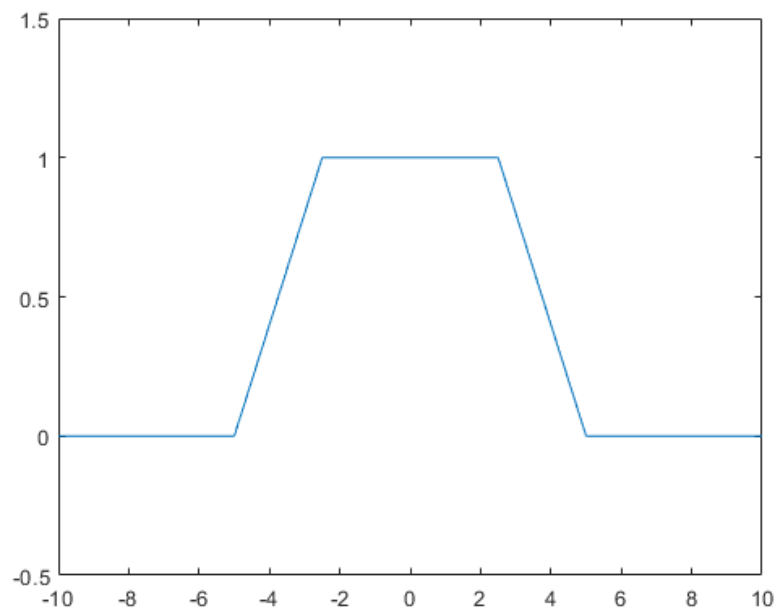
grupo, não caracterizando comportamento aleatório. As figuras 9 a 11 mostram as funções de pertinência expostas nesta seção.

Figura 9 – Função de pertinência triangular



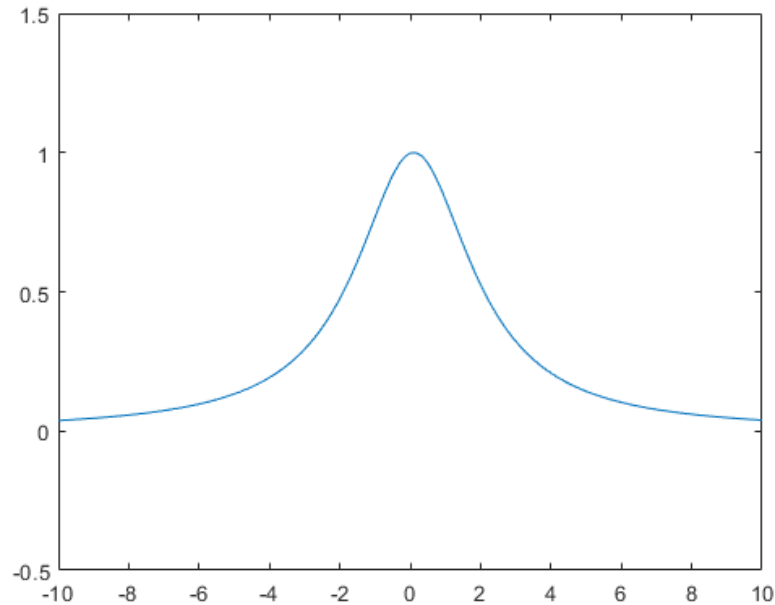
Fonte: Próprio autor

Figura 10 – Função de pertinência trapezoidal



Fonte: Próprio autor

Figura 11 – Função de pertinência gaussiana



Fonte: Próprio autor

2.2.2.1 Operações lógicas

Como na lógica booleana, existem operações lógicas que podem ser estabelecidas entre conjuntos *fuzzy*. Da mesma forma que o pertencimento de um elemento a um conjunto *fuzzy* depende das funções de pertinência de tal conjunto, as operações lógicas ocorrem sobre essas funções (PASSINO *et al.*, 1998). Considere dois conjuntos *fuzzy* A_1 e A_2 , com funções de pertinência $\mu_{A_1}(\cdot)$ e $\mu_{A_2}(\cdot)$, respectivamente. As operações lógicas são (PASSINO *et al.*, 1998):

- Complemento: o complemento de um conjunto A_1 é o conjunto A_1' , representado pela função de pertinência:

$$\mu_{A_1'}(\cdot) = 1 - \mu_{A_1}(\cdot) \quad (2.28)$$

- Interseção: há mais de uma maneira de se representar o conjunto *fuzzy* resultante da interseção de dois conjuntos *fuzzy*. Aqui serão apresentados dois métodos. O primeiro é por meio do mínimo entre as funções de pertinência dos conjuntos A_1 e A_2 . Matematicamente:

$$\mu_{A_1 \cap A_2}(\cdot) = \min(\mu_{A_1}(\cdot), \mu_{A_2}(\cdot)) \quad (2.29)$$

De outra forma, pode-se utilizar o produto das funções de pertinência para obter a função de pertinência do conjunto interseção:

$$\mu_{A_1 \cap A_2}(\cdot) = \mu_{A_1}(\cdot) \times \mu_{A_2}(\cdot) \quad (2.30)$$

- União: assim como a interseção, há mais de uma maneira de representar a função de pertinência do conjunto *fuzzy* resultante da união entre outros dois conjuntos. Aqui serão apresentados dois métodos. O primeiro é por meio do máximo entre as funções de pertinência dos conjuntos A_1 e A_2 . Matematicamente:

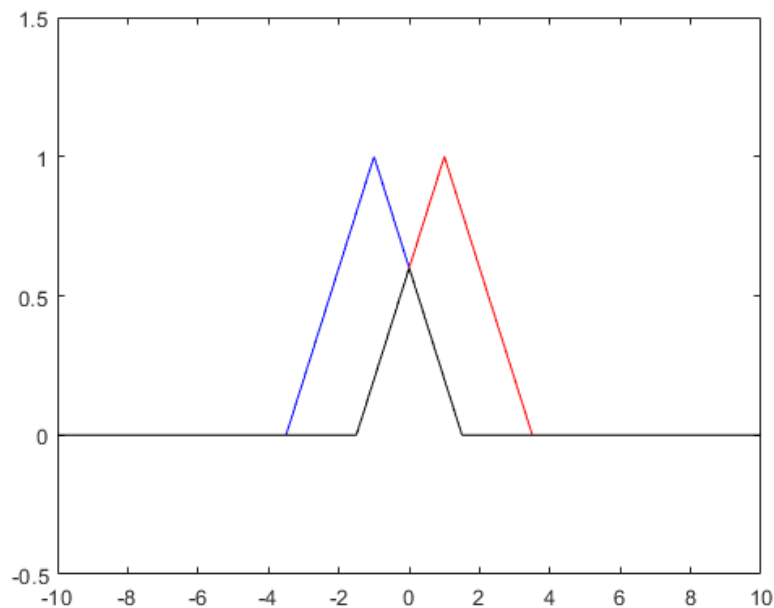
$$\mu_{A_1 \cup A_2}(\cdot) = \max(\mu_{A_1}(\cdot), \mu_{A_2}(\cdot)) \quad (2.31)$$

O segundo método é por meio de uma soma algébrica representada abaixo:

$$\mu_{A_1 \cup A_2}(\cdot) = \mu_{A_1}(\cdot) + \mu_{A_2}(\cdot) - \mu_{A_1}(\cdot) \times \mu_{A_2}(\cdot) \quad (2.32)$$

A figura 12 mostra a função de pertinência da interseção entre dois conjuntos *fuzzy*, enquanto a figura 13 mostra o resultado da união.

Figura 12 – Interseção de duas funções de pertinência triangulares



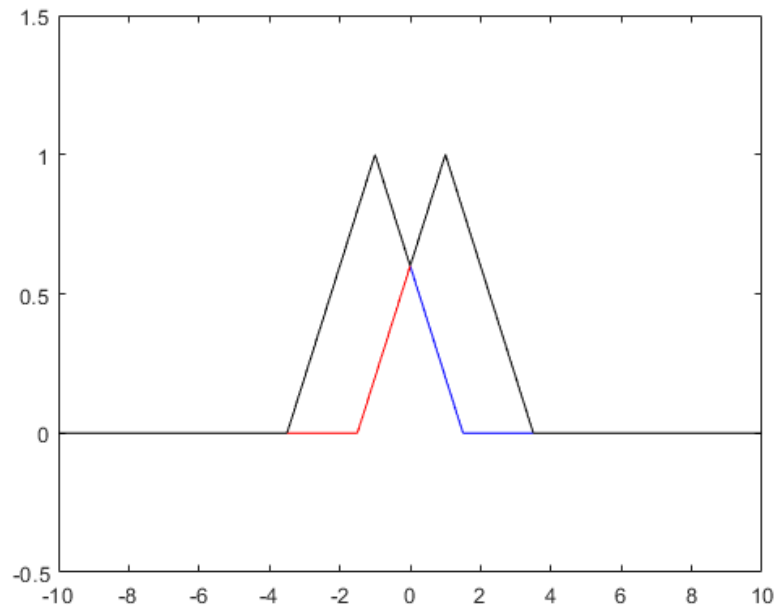
Fonte: Próprio autor

As operações lógicas serão importantes na próxima subseção, onde serão abordadas as regras *fuzzy*.

2.2.3 Variáveis linguísticas e regras

Variáveis linguísticas são assim chamadas porque recebem nomes, e não valores numéricos, facilitando a compreensão de como um sistema *fuzzy* funciona. Exemplos de variáveis linguísticas são *velocidade* e *acelerador*, introduzidas no início desta seção.

Figura 13 – União de duas funções de pertinência triangulares



Fonte: Próprio autor

O conjunto de regras de um sistema *fuzzy* é o que define as ações que o sistema tomará com base em determinadas entradas. Como exemplo, considere um sistema de inferência *fuzzy* que toma como entrada o tempo e a distância até um destino, e retorne se um indivíduo deve ir de carro, de ônibus ou a pé. Um possível conjunto de regras será:

- Se o dia estiver ensolarado e o destino for próximo, vá a pé;
- Se o dia estiver ensolarado e o destino for distante, vá de ônibus;
- Se no dia estiver chovendo, vá de carro.

Das três regras, duas possuem o operador interseção atuando na premissa das regras, definida como a parte referente às entradas, e geram um conjunto *fuzzy* de saída, sendo chamado de consequente.

É importante frisar que o sistema *fuzzy* aqui descrito possui uma saída discreta (carro, ônibus ou a pé, sem ação intermediária), servindo apenas como exemplo, mas controladores e sistemas identificados *fuzzy* são sistemas essencialmente contínuos, limitados apenas pela resolução do conversor digital/analógico (D/A) e analógico/digital (A/D) presente no computador onde o sistema for implementado.

2.2.4 Defuzzificação

A defuzzificação é o processo pelo qual conjuntos *fuzzy* de saída são convertidos em valores que podem ser utilizados por sistemas reais, sendo um sinal de saída predito do sistema no caso de identificação de sistemas *fuzzy* (PASSINO *et al.*, 1998). Existem diversas formas de se implementar a defuzzificação. As formas citadas aqui são:

- Centro de gravidade: utiliza as áreas das curvas do conjunto *fuzzy* de saída em uma média ponderada para gerar o valor de saída;
- Bissetor de área: calcula o ponto que particiona a área embaixo da curva em duas áreas iguais (MATHWORKS, 2021)

A defuzzificação por centro de gravidade pode ser melhor entendida vendo a equação 2.33

$$o = \frac{\sum_i b_i \int \mu_{(i)}}{\sum_i \int \mu_{(i)}} \quad (2.33)$$

onde o é a saída, b_i é o centro da função de pertinência do consequente da regra i , e $\int \mu_{(i)}$ representa a área sob a função de pertinência $\mu_{(i)}$. Nota-se que as áreas das funções de pertinência servem como pesos em uma média ponderada dos centros dos consequentes das regras.

É interessante notar que as funções de pertinência das variáveis linguísticas de saída nunca podem saturar, devendo as funções que estão localizadas nos extremos zerarem para valores muito altos ou muito baixos da saída. Isso ocorre porque, caso pudessem saturar, a defuzzificação não seria possível, pois a área sob as curvas seria infinita, não sendo possível achar centroide para o método do centro de gravidade ou a bissetora para o método do bissetor de área.

2.3 ANFIS

O acrônimo ANFIS significa, na sigla em inglês, *Adaptive Network-Based Fuzzy Inference System*, ou Sistema de Inferência *Fuzzy* Baseado em Rede (Jang, 1993), na tradução livre. O ANFIS utiliza uma estrutura de rede para implementar sistemas de inferência *fuzzy*, o que lhe permite o uso de algoritmos de aprendizagem para atualização de seus parâmetros. Tal estrutura foi proposta primeiramente por Jang (1993).

2.3.1 Arquitetura ANFIS

A arquitetura ANFIS possui cinco camadas, divididas da seguinte maneira (Jang, 1993):

1. Passagem por uma função de pertinência, de preferência uma com formato de sino, tal como a equação 2.27.
2. Avaliação das regras *fuzzy*, através de um produto entre as entradas correspondentes da regra, gerando pesos. Outras normas T também podem ser aplicadas.

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1, 2 \quad (2.34)$$

3. Normalização dos pesos, onde cada um será dividido pela soma de todos os outros.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad (2.35)$$

4. Multiplicação dos pesos por funções f que dependem das entradas e de parâmetros próprios.

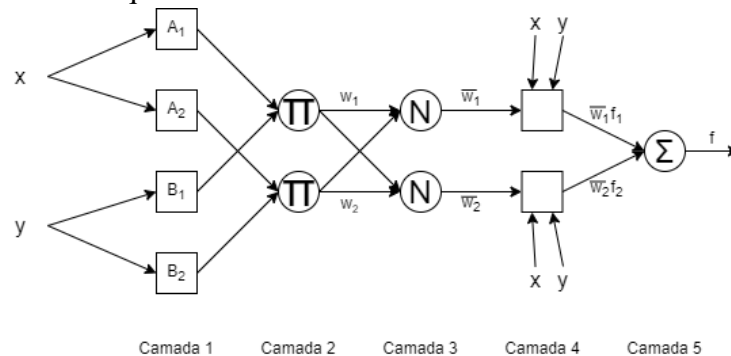
$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (2.36)$$

5. Por fim, somam-se todas as saídas da camada anterior, gerando a saída final.

$$O_i^5 = \sum_i \bar{w}_i f_i \quad (2.37)$$

É importante notar que tanto os parâmetros das funções de pertinência quanto das funções f serão alterados, mas são utilizados dois paradigmas de atualização diferentes, o chamada aprendizado híbrido (Jang, 1993). A figura 14 mostra um exemplo de arquitetura ANFIS mostrada em Jang (1993).

Figura 14 – Arquitetura ANFIS



Fonte: Adaptado de Jang (1993)

A próxima seção trata dos métodos de aprendizado utilizados no ANFIS.

2.3.2 *Aprendizado ANFIS*

De acordo com Jang (1993), o aprendizado de redes ANFIS pode seguir quatro abordagens diferentes, sendo a adotada pelo artigo a abordagem híbrida. Isso significa que a regra de atualização dos parâmetros das funções de pertinência é diferente da regra de atualização dos parâmetros das funções lineares de saída. Os dois métodos de aprendizado são apresentados em seguida.

2.3.2.1 *Estimação por mínimos quadrados*

O método dos mínimos quadrados pode ser utilizado para a atualização dos parâmetros consequentes, pois a função consequente é linear em seus parâmetros. Logo, é possível montar uma equação matricial a partir dos dados de treinamento:

$$A\theta = B \quad (2.38)$$

onde A é a matriz de entrada do consequente, B é a matriz de saída e θ é a matriz de parâmetros. É importante salientar que A não representa a entrada à rede ANFIS, mas à entrada na quarta camada, ou seja, representa a saída da terceira camada quando avaliada por cada elemento de treino. De posse da equação 2.38, a estimação por mínimos quadrados que minimiza o erro é dada por:

$$\theta^* = (A^T A)^{-1} A^T B \quad (2.39)$$

onde θ^* representa a estimativa de θ .

2.3.2.2 *Algoritmo de retropropagação*

Sendo o ANFIS uma rede adaptativa, é possível utilizar um algoritmo de retropropagação para propagar os erros até os parâmetros de entrada e utilizar a regra delta para a correção de tais parâmetros. Assim, o conceito é o mesmo do algoritmo de retropropagação utilizado na seção 2.1.3.1, apenas adaptado à arquitetura ANFIS. Considerando $\#(k)$ o número de elementos da camada k e L o número de elementos da última camada, pode-se escrever (Jang, 1993):

$$E_p = \sum_{m=1}^{\#(L)} (T_{m,p} - O_{m,p}^L) \quad (2.40)$$

onde E_p representa a medida de erro quadrático do p -ésimo elemento de treino, sendo $1 \leq p \leq P$ e P o número de elementos do vetor de treino, $T_{m,p}$ a m -ésima saída do p -ésimo elemento de

treino e $O_{m,p}^L$ a m -ésima saída da última camada do ANFIS referente ao p -ésimo elemento de treino. Derivando em relação à i -ésima saída da última camada:

$$\frac{\partial E_p}{\partial O_{i,p}^L} = -2(T_{i,p} - O_{m,p}^L) \quad (2.41)$$

Para outras camadas, utiliza-se a retropropagação, segundo a equação seguinte:

$$\frac{\partial E_p}{\partial O_{i,p}^k} = \sum_{m=1}^{\#(k+1)} \frac{\partial E_p}{\partial O_{m,p}^{k+1}} \frac{\partial O_{m,p}^{k+1}}{\partial O_{i,p}^k} \quad (2.42)$$

Com base nas equações 2.41 e 2.42, pode-se utilizar a regra delta e atualizar um parâmetro genérico α com base na equação seguinte:

$$\Delta\alpha = -\eta \frac{\partial E_p}{\partial \alpha} \quad (2.43)$$

onde η é a taxa de aprendizagem determinada pelo projetista.

3 IDENTIFICAÇÃO ANFIS

Muitas plantas reais possuem um comportamento complexo, sendo difíceis de serem modeladas matematicamente. Esta dificuldade suscita a busca por um método alternativo de modelagem para que possam ser estudadas e controladas. A área de estudo que abrange esses métodos chama-se identificação de sistemas dinâmicos. Este capítulo introduz conceitos de identificação de sistemas, bem como a modelagem da planta e define o papel do ANFIS como modelo de identificação.

3.1 Identificação de Sistemas Dinâmicos

Há diversas formas de modelar sistemas dinâmicos. A primeira e mais conhecida delas é a modelagem matemática, tendo a vantagem de ser transparente a quem modela o sistema ou o utiliza. Apesar disso, este tipo de modelagem rapidamente fica difícil com o aumento da complexidade dos sistemas, necessitando-se de técnicas de modelagem que não envolvam o conhecimento do sistema. Muitos desses modelos usam apenas os dados de entrada e saída do sistema para serem modelados, enquanto outros utilizam tanto esses dados como um conhecimento *a priori* do sistema para determinar um modelo. Assim, há três maneiras de classificar tais modelos (AGUIRRE, 2004):

- Caixa branca: são os modelos matemáticos. São denominados assim porque são transparentes e quem o modela tem acesso a todos os seus componentes;
- Caixa preta: são modelos que usam unicamente os dados de entrada e saída para treinar uma estrutura pré-definida;
- Caixa cinza: são modelos que utilizam os dados de entrada e saída juntamente com conhecimentos prévios acerca da planta a ser identificada. São mais transparentes que os modelos de caixa preta, mas mais obscuros que aqueles de caixa branca.

Quando se identifica um sistema, é preciso testá-lo para verificar se reproduz bem o comportamento esperado. Para fazer isso, usam-se *dados de validação*, que devem ser diferentes dos *dados de identificação* (AGUIRRE, 2004). Não existe uma regra fixa de como deve ser feita a escolha desses dados, porém, é importante que busquem capturar os comportamentos de regime transitório e de regime permanente do sistema. É possível que os dados de identificação e de validação venham de um mesmo ensaio, desde que sejam oriundos de intervalos diferentes. Além de escolher dados adequados para excitar o sistema, é importante escolher um modelo

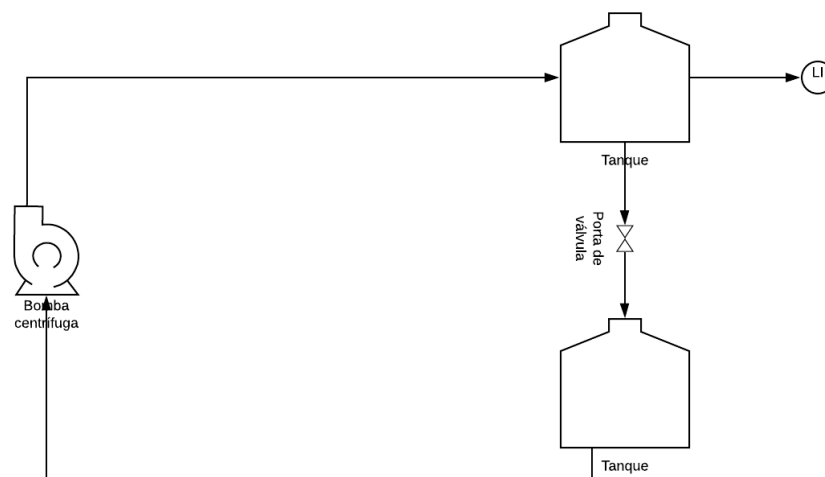
adequado (LANDAU; ZITO, 2007; ÅSTRÖM; WITTENMARK, 2013).

3.2 Descrição da bancada e modelagem matemática

Antes de se fazer a identificação propriamente dita do sistema em questão, ele será modelado para que se possam comparar os modelos fenomenológico e determinado por meio de identificação ANFIS. A modelagem do tanque foi baseada na seção 1.4 de Aguirre (2004).

Começa-se pela descrição da planta. O sistema consiste em um conjunto de dois tanques com formato de tronco de cone invertido, estando um no nível do piso (tanque inferior) e outro a aproximadamente 5 metros de altura (tanque superior). Ambos os tanques possuem volume de 2000 litros, diâmetro de 127 cm na sua parte inferior, diâmetro de 146 cm na sua parte superior e altura de 138 cm. Uma bomba centrífuga de 1,5 cv é usada para bombear água do tanque inferior ao tanque superior, havendo um ladrão por onde a água do tanque superior escoava de volta ao tanque inferior, formando-se um circuito fechado para a água. Por fim, diferentemente do sistema descrito em Aguirre (2004), onde a tubulação de recalque jogava água pelo nível zero do tanque superior, a tubulação de recalque do presente sistema joga água por uma abertura no topo do tanque superior. A figura 15 mostra o sistema.

Figura 15 – Sistema de tanques



Fonte: Próprio autor

A figura 15 mostra uma válvula na tubulação de descida do sistema. Ela representa a válvula no sistema real em que a simulação foi baseada e serve para regular a descida da água no sistema.

A variável de entrada do sistema é a velocidade angular da bomba, cujo valor altera

o valor de vazão da bomba, enquanto a variável de saída é a altura da água no tanque superior.

Cabe elencar dois pontos onde a modelagem deste trabalho e a de Aguirre (2004) são diferentes. O primeiro deles diz respeito ao formato do reservatório de água. No trabalho referido, o reservatório possui área transversal constante, enquanto a planta aqui presente tem o formato de um tronco de cone invertido, fazendo com que sua área transversal aumente com o aumento do nível de líquido. O segundo ponto é referente à bomba que joga água no tanque. A bomba de Aguirre (2004) é acionada por um inversor que, por sua vez, é acionada por sinal de comando de corrente entre 4 e 20 mA. Já o conjunto motobomba deste trabalho foi modelado como sendo acionado por um sinal de velocidade angular ideal, fazendo com que a vazão de entrada no tanque seja proporcional a essa velocidade angular.

Depois de explicitadas as diferenças, inicia-se a modelagem da planta descrevendo a variação de volume do tanque superior. Essa variação depende das vazões de entrada e de saída, escrevendo-se:

$$\frac{dV}{dt} = q_i - q_o \quad (3.1)$$

onde V é o volume de líquido no tanque, q_i é a vazão de entrada e q_o é a vazão de saída. O volume de um tronco é determinado pela seguinte equação:

$$V = \frac{\pi h}{3} (R^2 + Rr + r^2) \quad (3.2)$$

onde h é a altura do tronco de cone, R é o raio da base maior e r é o raio da base menor. É possível escrever R em função de r e de h . Para isso, usa-se semelhança de triângulos a partir da seção de um cone, mostrada na figura 16:

A variável h' pode ser determinada a partir de simetria de triângulos:

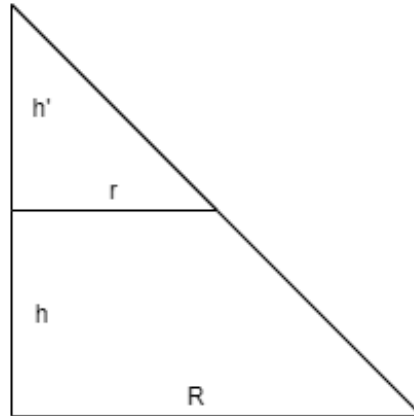
$$\frac{h'}{r} = \frac{h' + h}{R} \quad (3.3)$$

$$h' = \frac{rh}{R - r} \quad (3.4)$$

A partir da equação 3.3, R pode ser determinada rearrajando a equação, resultando:

$$R = \frac{r(h' + h)}{h'} \quad (3.5)$$

Figura 16 – Seção de um cone



Fonte: Próprio autor

Substituindo R em 3.2:

$$V = \frac{\pi h}{3} \left(\left(\frac{r(h'+h)}{h'} \right)^2 + \frac{r^2(h'+h)}{h'} + r^2 \right) \quad (3.6)$$

Derivando a equação 3.6 em relação ao tempo:

$$\frac{dV}{dt} = \frac{\pi}{3} \frac{dh}{dt} \left(\left(\frac{r(h'+h)}{h'} \right)^2 + \frac{r^2(h'+h)}{h'} + r^2 \right) + \frac{\pi h}{3} \left(\frac{2r^2(h'+h)}{h'^2} \frac{dh}{dt} + \frac{r^2}{h'} \frac{dh}{dt} \right) \quad (3.7)$$

É possível substituir a equação 3.7 na equação 3.1:

$$\frac{\pi}{3} \frac{dh}{dt} \left(\left(\frac{r(h'+h)}{h'} \right)^2 + \frac{r^2(h'+h)}{h'} + r^2 \right) + \frac{\pi h}{3} \left(\frac{2r^2(h'+h)}{h'^2} \frac{dh}{dt} + \frac{r^2}{h'} \frac{dh}{dt} \right) = q_i - q_o \quad (3.8)$$

Reorganizando a equação:

$$\frac{dh}{dt} = \frac{3}{\pi} (q_i - q_o) \left(\frac{r^2(2h+1)(h'+h)}{h'^2} + \frac{r^2(h'+2h)}{h'} + r^2 \right)^{-1} \quad (3.9)$$

A vazão q_i de regime permanente depende da velocidade angular da máquina ω , como na equação seguinte:

$$q_i = k_i \omega \quad (3.10)$$

onde k_i é uma constante que relaciona a vazão q_i com a velocidade angular ω . Pode-se utilizar a lei de Bernoulli para se estimar a vazão de saída (AGUIRRE, 2004):

$$q_o = k_o \sqrt{\Delta P} \quad (3.11)$$

onde k_o é a constante de resistência hidráulica na tubulação de descida e ΔP é a diferença de pressão entre as extremidades da tubulação. Assim:

$$q_o = k_o \sqrt{P - P_{atm}} \quad (3.12)$$

onde P representa a pressão na entrada do tubo e P_{atm} representa a pressão atmosférica. P pode ser escrita como:

$$P = P_{atm} + \gamma h \quad (3.13)$$

onde γ representa o peso específico da água. Logo, tem-se:

$$q_o = k_o \sqrt{\gamma h} \quad (3.14)$$

Substituindo 3.10 e 3.14 em 3.9:

$$\frac{dh}{dt} = \frac{3}{\pi} \left(k_i \omega - k_o \sqrt{\gamma h} \right) \left(\frac{r^2 (2h + 1) (h' + h)}{h'^2} + \frac{r^2 (h' + 2h)}{h'} + r^2 \right)^{-1} \quad (3.15)$$

Depois de desenvolvida a equação, resta determinar os valores das constantes. A variável h' pode ser calculada a partir dos valores que já se tem dos diâmetros inferior e superior do tanque, e de sua altura na equação 3.4:

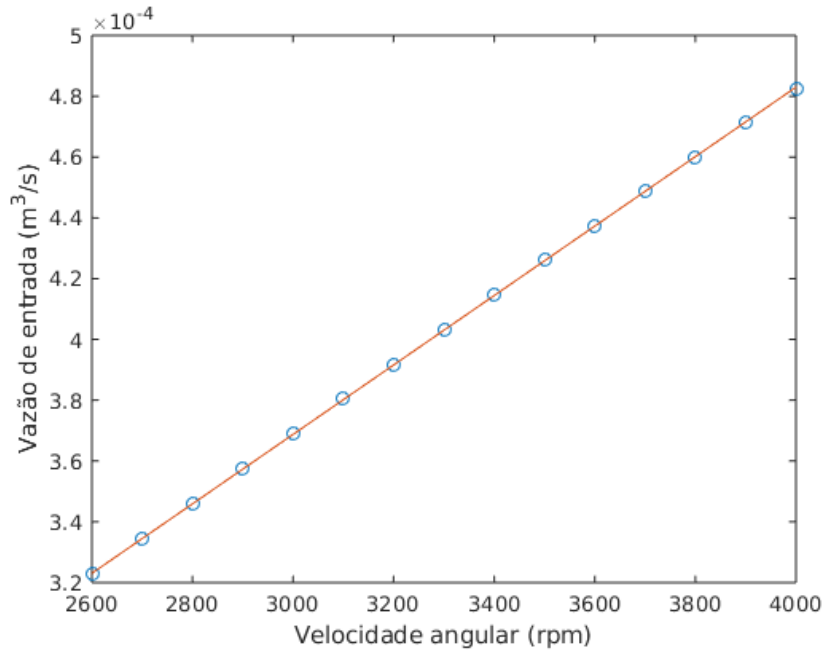
$$h' = \frac{127/2 \times 138}{146/2 - 127/2} \quad (3.16)$$

É importante notar que a equação usava o raio, portanto, os diâmetros foram divididos por 2 para chegar-se a esses valores.

Logo, tem-se o valor $h' = 922,42cm = 9,22m$. Resta determinar os valores de k_i e de k_o . Esses valores são geralmente determinados experimentalmente, mas foram feitas simulações, sendo eles determinados a partir do modelo construído em simulação.

Para determinar o valor de k_i , é necessário traçar o gráfico $q_i \times \omega$. Os valores são coletados depois de atingido o regime permanente para sinais de entrada determinados. Com base no gráfico, pode-se regressir os dados e, com base na equação encontrada, determinar o valor de k_i . A figura 17 mostra o gráfico já com a reta de regressão, enquanto a 3.17 é a equação dessa reta.

Figura 17 – Gráfico $q_i \times \omega$



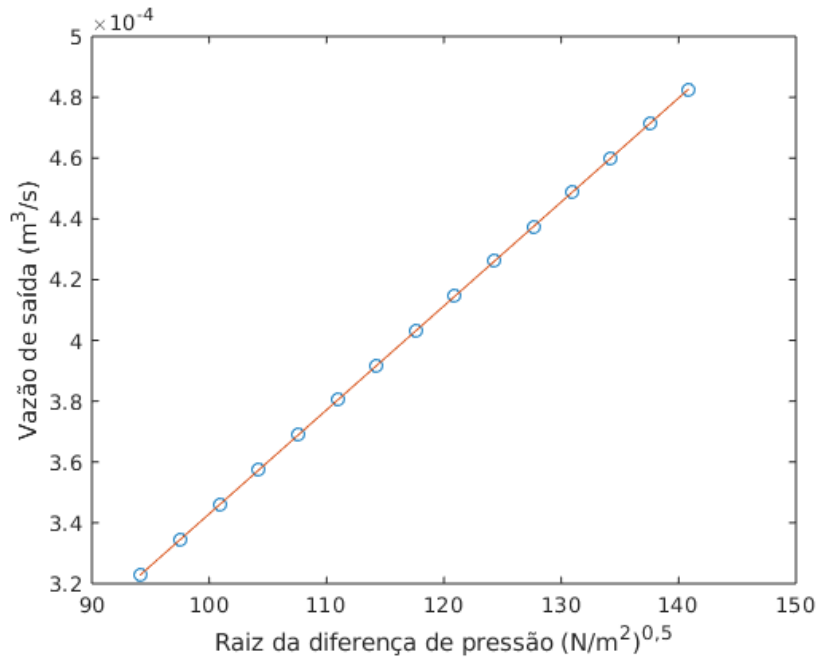
Fonte: Próprio autor

$$q_i = 2,61 \times 10^{-5} + 1,14 \times 10^{-7} \omega \quad (3.17)$$

Assim, tem-se:

$$k_i = \frac{\partial q_i}{\partial \omega} = 1,14 \times 10^{-7} \quad m^3 \cdot rpm/s \quad (3.18)$$

De forma similar, para determinar o valor de k_o , é preciso traçar o gráfico $q_o \times \sqrt{\gamma h}$. Novamente, os valores são coletados depois de atingido o regime permanente para sinais de entrada determinados. A partir do gráfico, pode-se verificar o formato aproximado da curva e determinar a quantidade de coeficientes da regressão linear. Assim, é possível determinar a equação que governa a relação entre a vazão de saída e a raiz quadrada da diferença de pressão entre a entrada e a saída da tubulação de saída. A figura 18 mostra o gráfico já com a curva de regressão, equação 3.19. É importante notar que se trata de uma reta.

Figura 18 – Gráfico $q_o \times \sqrt{\gamma h}$ 

Fonte: Próprio autor

$$q_o = 9,51 \times 10^{-7} + 3,42 \times 10^{-6} \sqrt{\gamma h} \quad (3.19)$$

Por fim, tem-se:

$$k_o = \frac{\partial q_o}{\partial \sqrt{\gamma h}} = 3,42 \times 10^{-6} \quad m^4 / (s \cdot N^{1/2}) \quad (3.20)$$

Os valores encontrados das constantes h' , k_i e k_o podem ser substituídos na equação 3.9.

3.3 Papel do ANFIS

Como exposto na seção 2.3, o ANFIS é não-linear, sendo mais flexível no que ele pode fazer do que um modelo linear comum. Devido a isso, ele é um bom candidato a modelo de identificação do sistema de tanques e deve conseguir captar bem tanto a dinâmica não-linear do processo quanto seu comportamento em regime permanente. O capítulo 4 demonstra a capacidade de identificação do ANFIS. Sua seção 4.1 compara o desempenho do modelo matemático encontrado na seção 3.2 e do modelo ANFIS identificado naquele capítulo.

4 RESULTADOS E APLICAÇÕES

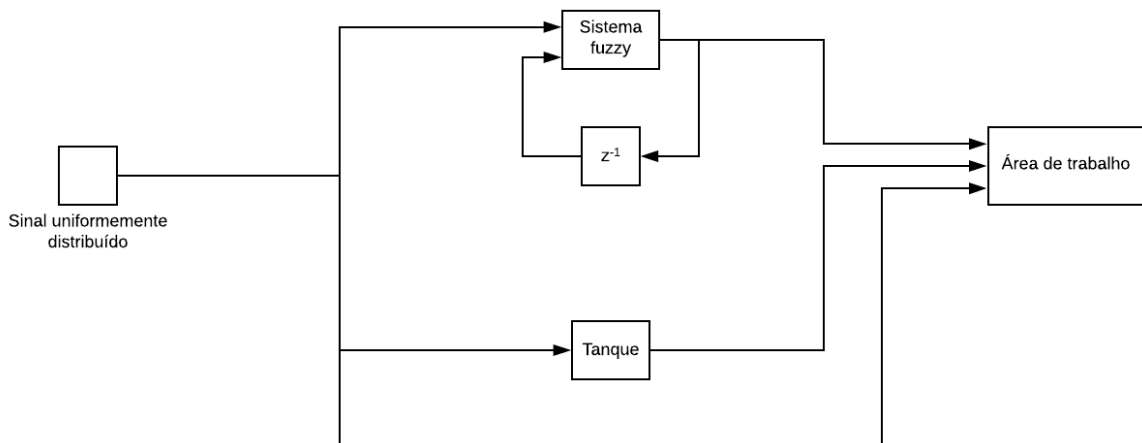
A intenção do presente trabalho é o controle de nível de líquido de um sistema de tanques. Um conjunto motobomba, sendo o motor de indução trifásico e a bomba centrífuga, bombeia água de um tanque localizado ao nível do solo a um tanque localizado a aproximadamente 5 metros acima do nível do solo. Um ladrão localizado na parte de baixo do tanque superior escoar a água para o tanque inferior, fechando o circuito hidráulico. A tarefa do controlador projetado é controlar a quantidade de água bombeada para o tanque superior de modo que, em regime permanente, o nível de água nele permaneça constante em um dado valor de referência.

No sistema real em que a simulação se baseia, o controlador implementado em pacote computacional comunicar-se-ia com uma placa de aquisição de dados da National Instruments que, por sua vez, repassaria a informação a um Controlador lógico programável (CLP) S7-200 da Siemens para comandar o conjunto motobomba. Além disso, havia uma válvula no ladrão para regular o fluxo de água que retorna ao tanque inferior.

Deve-se dar ênfase ao fato de que a simulação foi feita para auxiliar uma planta existente no Lamotriz, servindo de modelo para estudos futuros. Foram feitas medidas e coletados dados referentes ao tanque e ao conjunto motobomba, como foram descritos na seção 3.2. Assim, os resultados aqui descritos servem como base para trabalhos feitos na planta real.

A figura 19 mostra o diagrama de blocos utilizado para a coleta de dados do sistema a fins de sua identificação.

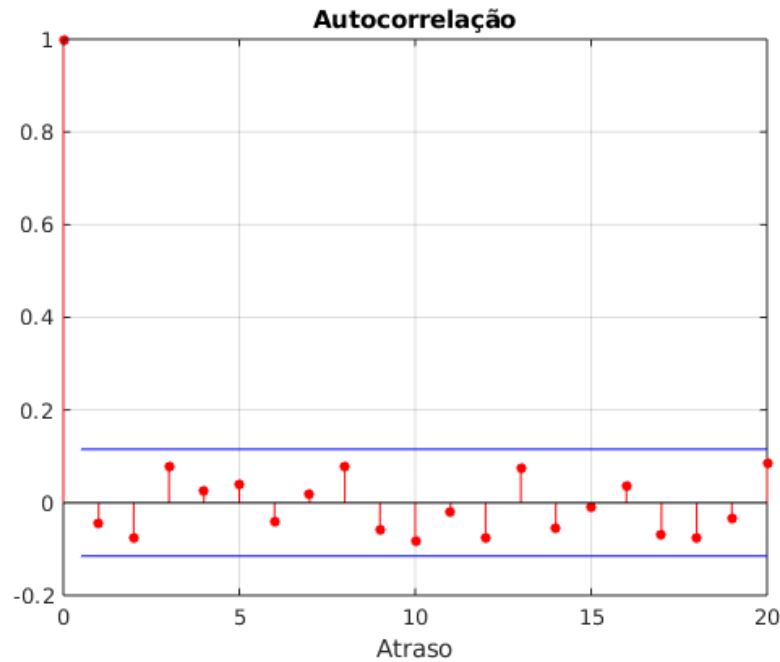
Figura 19 – Identificação do sistema



Fonte: Próprio autor

Primeiramente, uma entrada aleatória uniformemente distribuída com valores entre 0 e 3600 rotações por minuto (rpm) e com tempo de amostragem de 5000 segundos foi aplicada na entrada do sistema e dados de entrada e saída foram coletados a cada 50 segundos. A planta é lenta, demorando a atingir o regime permanente, o que explica os tempos de amostragem aparentemente elevados. A figura 20 mostra a autocorrelação do sinal de entrada.

Figura 20 – Autocorrelação do sinal de entrada



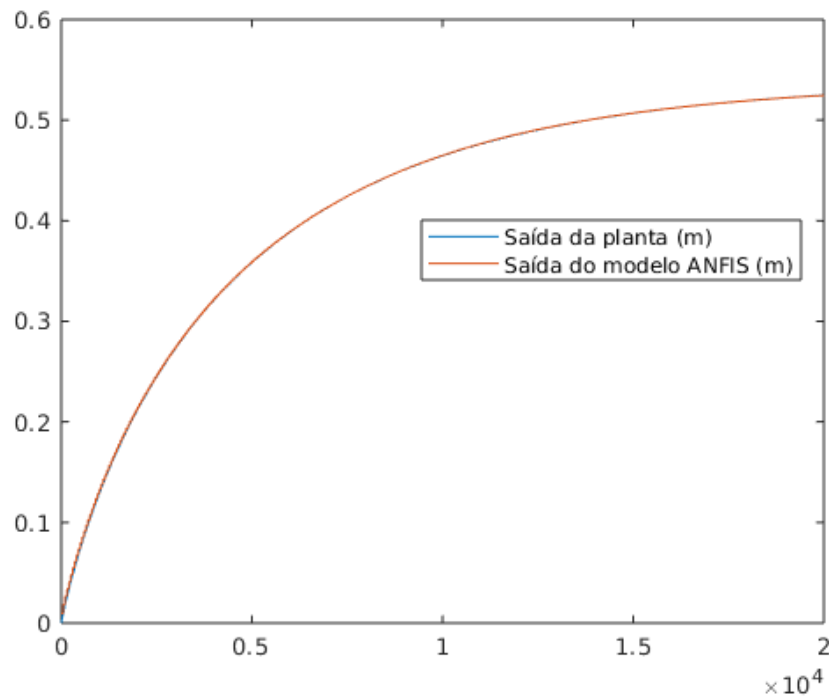
Fonte: Próprio autor

O ANFIS é, em essência, um sistema estático. Para contornar o problema, realimenta-se uma entrada, fazendo com que sua saída fique da forma $y[k] = f(u[k-1], y[k-1])$, sendo $u[k-1]$ e $y[k-1]$ a entrada e a saída no instante $k-1$, respectivamente. Foram atribuídas 6 funções de pertinência para cada variável e foram simuladas 50 épocas de treinamento.

De posse dos dados da planta e com a estrutura do sistema ANFIS definido, o modelo foi treinado e, por fim, validou-se o sistema através da comparação entre a saída real da planta e a saída gerada pelo sistema ANFIS.

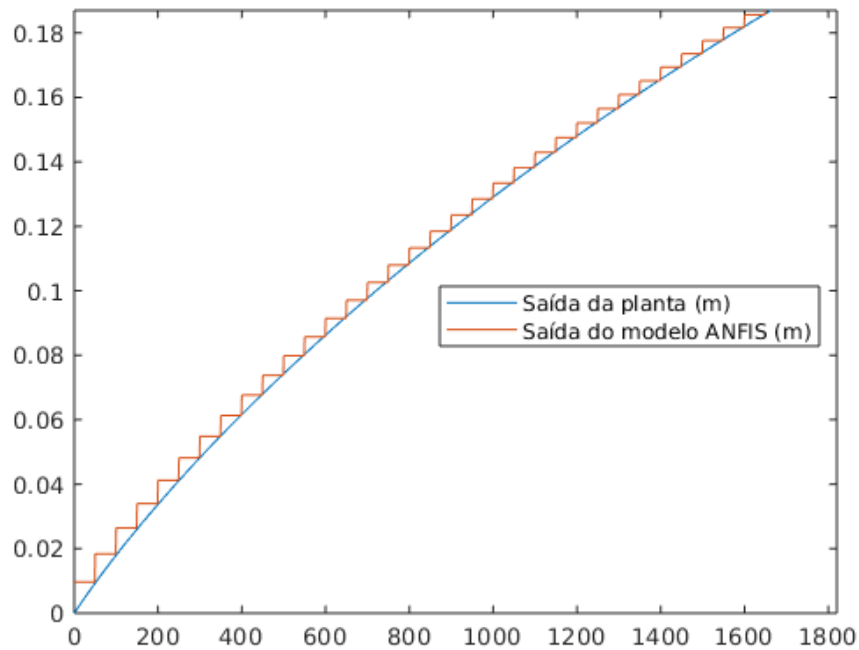
Duas entradas foram usadas para atestar o bom desempenho do sistema ANFIS treinado. A primeira entrada foi um degrau de 2000 rpm, enquanto a segunda entrada foi uma senoide com amplitude de 1800 rpm e *offset* também de 1800 rpm, além de ter uma frequência de 0,001 rad/s. Tanto no caso da entrada em degrau quanto no caso da entrada senoide foram simulados 20000 segundos. A figura 21 mostra as saídas da planta e do modelo para a entrada em degrau, ao passo que a figura 22 mostra uma ampliação dos sinais:

Figura 21 – Sinais de saída da planta e do modelo ANFIS para entrada em degrau



Fonte: Próprio autor

Figura 22 – Sinais de saída da planta e do modelo ANFIS para entrada em degrau ampliadas

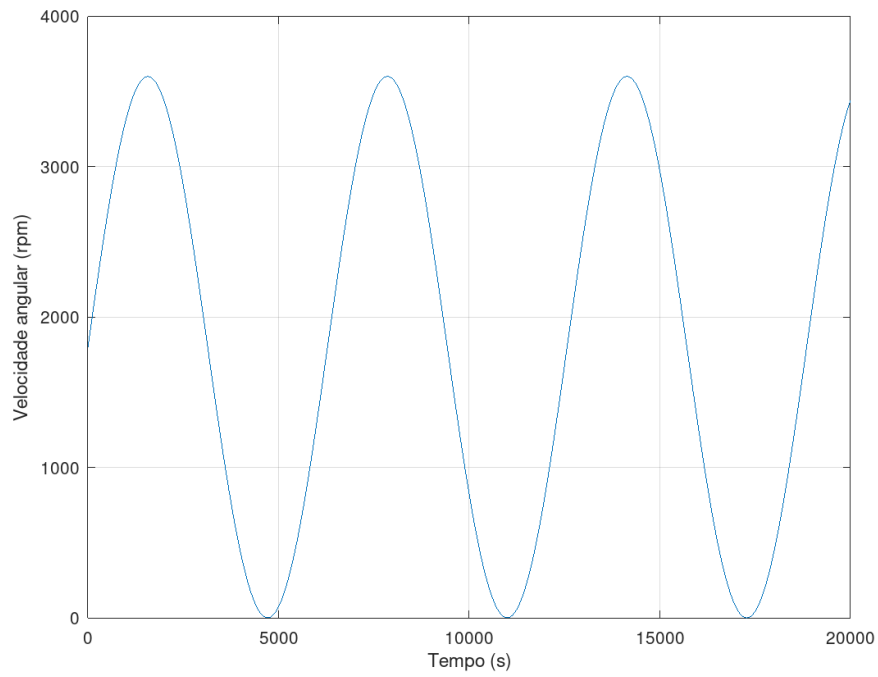


Fonte: Próprio autor

A figura 23 mostra o sinal de senoide usado como entrada na segunda simulação; a figura 24 mostra as saídas do modelo e da planta, enquanto a figura 25 mostra as mesmas saídas

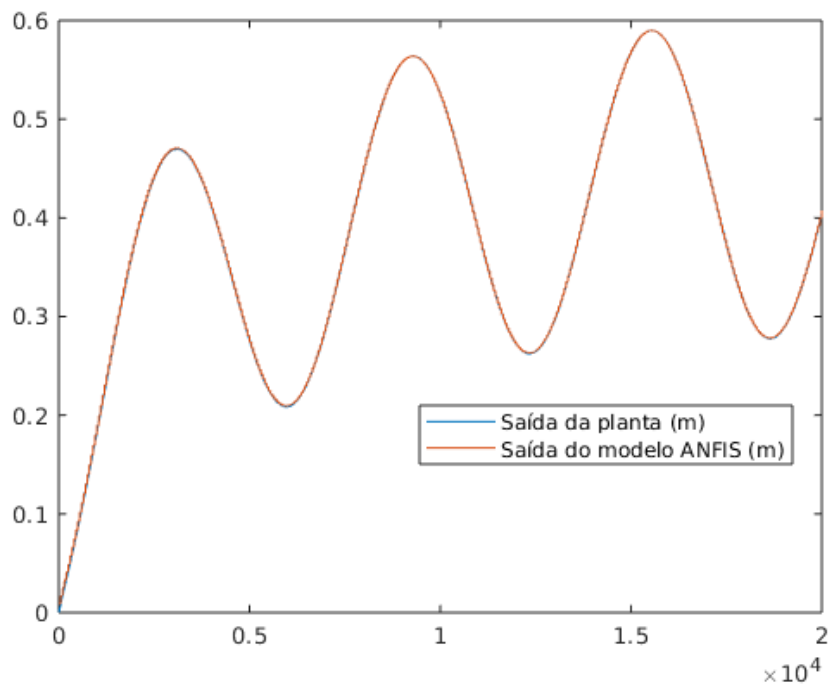
de forma ampliada.

Figura 23 – Sinal de entrada senoidal



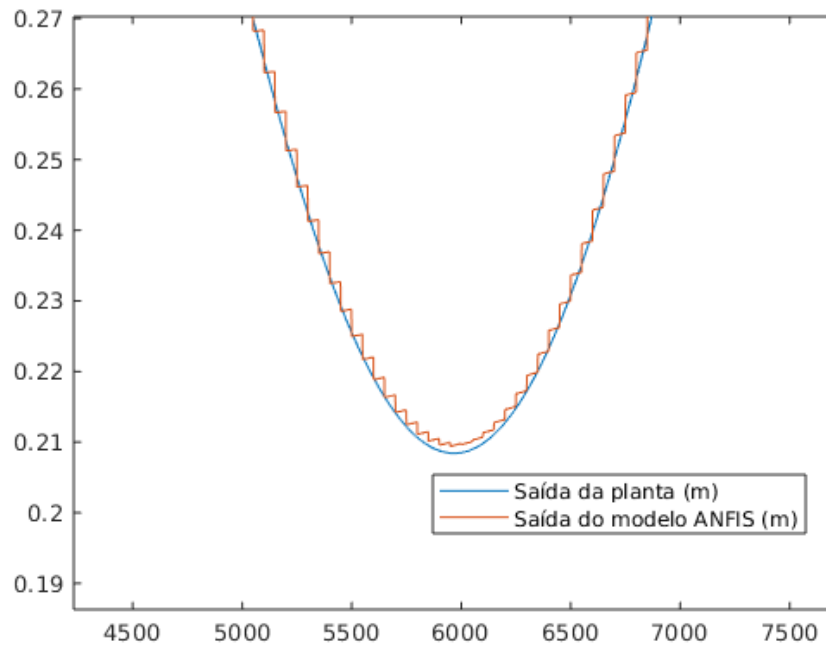
Fonte: Próprio autor

Figura 24 – Sinais de saída da planta e do modelo ANFIS para entrada senoidal



Fonte: Próprio autor

Figura 25 – Sinais de saída da planta e do modelo ANFIS para entrada senoidal ampliadas



Fonte: Próprio autor

Percebe-se que o sistema ANFIS treinado captou muito bem a dinâmica do sistema, atestando a capacidade do ANFIS de servir como modelo de identificação.

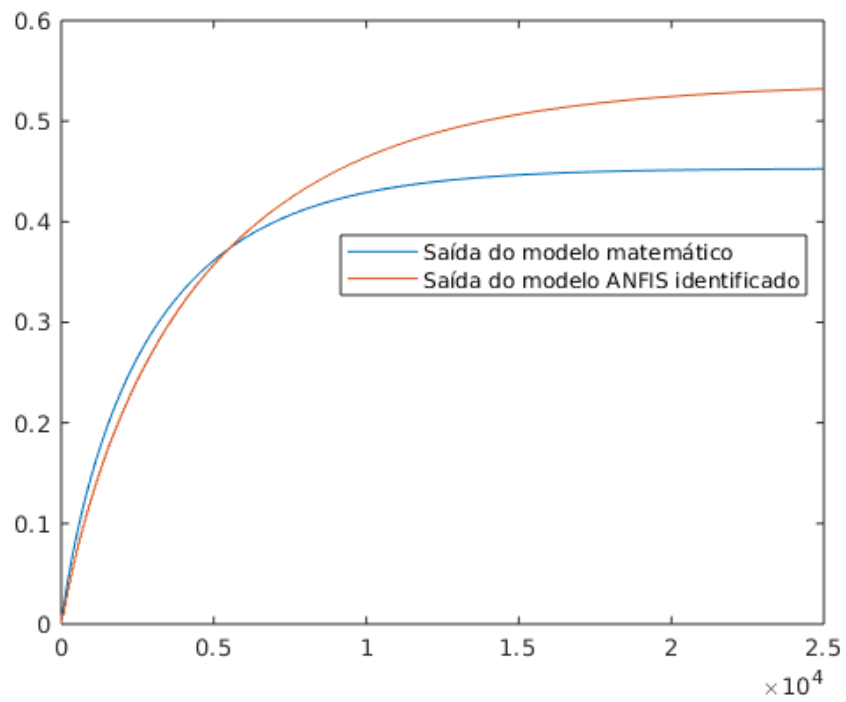
É importante perceber que as saídas do sistema simulado não possuem ruído. Se fosse uma planta real, tal ruído estaria presente e poderia dificultar a identificação.

4.1 Comparação entre a modelagem matemática e o modelo ANFIS

O modelo determinado matematicamente foi comparado com o modelo ANFIS identificado. A imagem 26 mostra a resposta de ambos os modelos para uma entrada de 2000 rpm durante 25000 segundos.

Percebe-se que o transitório do modelo matemático se aproxima do transitório do modelo ANFIS, mas seu valor de regime permanente é diferente, sendo 14,96% menor que o valor do modelo ANFIS. As limitações do modelo matemático ficam evidentes, enquanto o modelo ANFIS, como mostrado nas figuras 21 e 24, apresentou desempenho superior. Assim, verifica-se a superioridade da identificação de sistemas neste caso.

Figura 26 – Comparativo de modelo matemático com modelo ANFIS identificado



Fonte: Próprio autor

5 CONCLUSÕES E TRABALHOS FUTUROS

O trabalho se propôs a aplicar o algoritmo ANFIS na identificação de um sistema de tanques simulado, similar a um sistema real montado no Laboratório de Sistemas Motrizes da UFC. Através da modelagem e simulação do sistema de tanques, foi possível coletar dados de entrada e saída que foram usados para identificar o sistema.

A saída gerada pelo sistema identificado foi então comparada com a saída obtida pelo sistema montado, quando sujeitas à mesma entrada. Foram feitos dois testes deste tipo, sendo o primeiro com uma entrada em degrau e o segundo com uma entrada senoidal.

Verificou-se que o sistema identificado pelo algoritmo do ANFIS comportou-se de forma similar ao original, confirmando a tese inicial de que seria possível utilizá-lo para resolver problemas de identificação de sistemas. Ainda, o modelo ANFIS identificado mostrou-se superior ao modelo matemático.

A partir dos resultados aqui obtidos, pode-se concluir que os objetivos propostos foram satisfeitos. Ficam como sugestão de trabalhos futuros os seguintes trabalhos:

- **Identificação da planta real:** a identificação realizada neste trabalho foi a partir de dados de simulação. Usando os dados recolhidos a partir da planta real seria possível identificá-la;
- **Controle ANFIS do sistema de tanques:** usando o sistema identificado, fica facilitado o projeto de controladores diversos. Aqui propõe-se o uso também do ANFIS;
- **Uso do ANFIS noutras plantas:** por fim propõe-se que o ANFIS seja usado também noutras plantas, seja visando a identificá-las ou até mesmo controlá-las.

REFERÊNCIAS

- AGUIRRE, L. A. Introdução à identificação de sistemas. **Editora UFMG**, 2004.
- AL-SHAYEA, Q. K. Artificial neural networks in medical diagnosis. **International Journal of Computer Science Issues**, Citeseer, v. 8, n. 2, p. 150–154, 2011.
- ANTSAKLIS, P. J.; PASSINO, K. M. **An introduction to intelligent and autonomous control**. [S.l.]: Kluwer Academic Publishers, 1993.
- ÅSTRÖM, K. J.; WITTENMARK, B. **Computer-controlled systems: theory and design**. [S.l.]: Courier Corporation, 2013.
- BAHRAMMIRZAEI, A. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. **Neural Computing and Applications**, v. 19, n. 8, p. 1165–1195, 2010.
- Daneshwar, M. A.; Noh, N. M. Adaptive neuro-fuzzy inference system identification model for smart control valves with static friction. In: **2013 IEEE International Conference on Control System, Computing and Engineering**. [S.l.: s.n.], 2013. p. 122–126.
- HAYKIN, S. S. **Neural networks: a comprehensive foundation**. [S.l.]: Prentice-Hall, 1998.
- HAZNEDAR, B.; KALINLI, A. Training anfis structure using simulated annealing algorithm for dynamic systems identification. **Neurocomputing**, Elsevier, v. 302, p. 66–74, 2018.
- ISINKAYE, F. O.; FOLAJIMI, Y.; OJOKOH, B. A. Recommendation systems: Principles, methods and evaluation. **Egyptian Informatics Journal**, Elsevier, v. 16, n. 3, p. 261–273, 2015.
- Jang, J. . R.; Chuen-Tsai Sun. Neuro-fuzzy modeling and control. **Proceedings of the IEEE**, v. 83, n. 3, p. 378–406, 1995.
- Jang, J.-S. Anfis: adaptive-network-based fuzzy inference system. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 23, n. 3, p. 665–685, 1993.
- Jang, J.-S. R. Neuro-fuzzy modeling for dynamic system identification. In: **Soft Computing in Intelligent Systems and Information Processing. Proceedings of the 1996 Asian Fuzzy Systems Symposium**. [S.l.: s.n.], 1996. p. 320–325.
- LANDAU, I. D.; ZITO, G. **Digital control systems: design, identification and implementation**. [S.l.]: Springer Science & Business Media, 2007.
- Martins, J. K. E. da C.; Araújo, F. M. U. de. Nonlinear system identification based on modified anfis. In: **2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)**. [S.l.: s.n.], 2015. v. 01, p. 588–595.
- MATHWORKS. **MATLAB Fuzzy Logic Toolbox User's Guide**. 2021. The MathWorks, Natick, MA, USA.
- Muslim, M. A.; Nusantoro, G. D. System identification of a vacuum distiller using anfis with an armax structure. In: **2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)**. [S.l.: s.n.], 2016. p. 661–664.

PAN, H.; WONG, H.; KAPILA, V.; de Queiroz, M. S. Experimental validation of a nonlinear backstepping liquid level controller for a state coupled two tank system. **Control Engineering Practice**, v. 13, n. 1, p. 27–40, 2005. ISSN 0967-0661. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0967066103002946>>.

PASSINO, K. M.; YURKOVICH, S.; REINFRANK, M. **Fuzzy control**. [S.l.]: Citeseer, 1998. v. 42.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.

Zhixiang Hou; Quntai Shen; Heqing Li. Nonlinear system identification based on anfis. In: **International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003**. [S.l.: s.n.], 2003. v. 1, p. 510–512 Vol.1.