



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DO CABO DE SANTO AGOSTINHO
GRADUAÇÃO EM ENGENHARIA ELETRÔNICA

WESLEY IRIO DE LIMA SILVA

Estratégias de controle aplicadas no gerador de vapor

Cabo de Santo Agostinho - PE

2025

WESLEY IRIO DE LIMA SILVA

Estratégias de controle aplicadas no gerador de vapor

Monografia apresentada ao curso de Graduação em Engenharia Eletrônica da Unidade Acadêmica do Cabo de Santo Agostinho, Universidade Rural de Pernambuco, como requisito para obtenção do grau de Bacharel em Engenharia Eletrônica.

Orientada: Prof. (a) Dr.(a) Ania Lussón Cervantes

Cabo de Santo Agostinho - PE

2025

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Suely Manzi – CRB-4 809

S586e Silva, Wesley Irio de Lima.
Estratégias de controle aplicadas no gerador de vapor / Wesley Irio de Lima Silva. - Cabo de Santo Agostinho, 2025.
72 f.

Orientador(a): Ania Lussón Cervantes.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal Rural de Pernambuco, Unidade Acadêmica Cabo de Santo Agostinho - UACSA, Bacharelado em Engenharia Eletrônica, Cabo de Santo Agostinho, BR-PE, 2025.

Inclui referências.

1. Controle automático . 2. Geradores a vapor. 3. Redes neurais (Computação). 4. Lógica Fuzzy 5. Controle preditivo . I. Cervantes, Ania Lussón, orient. II. Título

CDD 621.3

WESLEY IRIO DE LIMA SILVA

Estratégias de controle aplicadas no gerador de vapor

Monografia apresentada ao Curso de Graduação em Engenharia Eletrônica da Universidade Federal Rural de Pernambuco, como requisito parcial para a obtenção do grau de Bacharel em Engenharia Eletrônica.

Aprovada em: 18/12/2025.

BANCA EXAMINADORA

Profa. Dra Ania Lussón Cervantes – Orientadora
UFRPE/UACSA

Prof. Dr. Felipe Alberto Barbosa Simão Ferreira - Examinador Interno
UFRPE/UACSA

Prof. Dr. Reinel Beltrán Aguedo - Examinador Interno
UFRPE/UACSA

RESUMO

Os geradores de vapor desempenham um papel essencial em diversas indústrias, como geração de energia, processos químicos e alimentícios, e são sistemas complexos que apresentam desafios significativos de controle devido a interações multivariáveis, comportamento não-linear e a influência de distúrbios. Nesse contexto, o controlador Proporcional-Integral-Derivativo (PID) é o controle convencional mais usado nas indústrias. Além, a Lógica *Fuzzy* surge como uma alternativa robusta, pois permite incorporar o conhecimento e a experiência de operadores humanos na forma de regras linguísticas, lidando eficazmente com as incertezas e imprecisões inerentes a processos não-lineares, sem exigir um modelo matemático exato. Alternativamente, as Redes Neurais Artificiais surgem como outra alternativa pois conseguem aprender o comportamento dinâmico a partir de dados, lidando bem com não linearidades. Este trabalho teve como objetivo realizar um estudo comparativo de estratégias de controle num gerador de vapor, em que as variáveis controladas são a pressão de vapor e o nível de água. A metodologia envolveu a obtenção do modelo matemático em espaço de estados do gerador e o desenvolvimento de 3 estratégias de controle: PID (com sintonia baseada em Otimização por Enxame de Partículas - PSO e Algoritmo Genético - AG), um controlador neural, utilizando o algoritmo *Soft Actor-Critic* de aprendizado por reforço no ambiente *CaldeiraEnv*, e por fim um controlador *Fuzzy*. O desempenho das estratégias foi avaliado por meio de métricas baseadas em parâmetros de resposta transitória e de índices integrais. Os resultados das simulações, mostraram um bom seguimento de referência e a rejeição dos distúrbios de todos os controladores, ressaltando que os controladores PID sintonizados por PSO e AG tiveram desempenho superior em relação aos outros controladores, apresentando índices de desempenho menores.

Palavras-chave: gerador de vapor; estratégias de controle; redes neurais artificiais; PID; lógica *fuzzy*.

ABSTRACT

Steam generators play an essential role in various industries, such as power generation, chemical and food processing, and are complex systems that present significant control challenges due to multivariable interactions, non-linear behavior, and the influence of disturbances. In this context, the Proportional-Integral-Derivative (PID) controller is the most widely used conventional control in industries. Furthermore, Fuzzy Logic emerges as a robust alternative, as it allows the incorporation of the knowledge and experience of human operators in the form of linguistic rules, effectively handling the uncertainties and inaccuracies inherent in non-linear processes, without requiring an exact mathematical model. Alternatively, Artificial Neural Networks emerge as another alternative because they can learn dynamic behavior from data, handling non-linearities well. This work aimed to conduct a comparative study of control strategies in a steam generator, where the controlled variables are steam pressure and water level. The methodology involved obtaining the mathematical state-space model of the generator and developing three control strategies: PID (with tuning based on Particle Swarm Optimization - PSO and Genetic Algorithm - GA), a neural controller using the *Soft Actor-Critic* reinforcement learning algorithm in the CaldeiraEnv environment, and finally a Fuzzy controller. The performance of the strategies was evaluated using metrics based on transient response parameters and integral indices. The simulation results showed good reference tracking and disturbance rejection for all controllers, highlighting that the PID controllers tuned by PSO and GA had superior performance compared to the other controllers, presenting lower performance indices.

Keywords: steam generator; control strategies; artificial neural networks; PID; *fuzzy* logic.

LISTA DE ILUSTRAÇÕES

Figura 1 - Classificação dos geradores de vapor.....	12
Figura 2 - Gerador Aquatubular.....	13
Figura 3 - Gerador Flamotubular.....	15
Figura 4 - Esquema do neurônio da Unidade Processadora de McCulloch e Pitts.....	16
Figura 5 - Funções de Ativação.....	18
Figura 6 - Perceptron de camada única.....	19
Figura 7 - Estrutura do Multilayer Perceptron.....	20
Figura 8 - Algoritmo de <i>BackPropagation</i>	24
Figura 9 - Fluxograma do Algoritmo Genético.....	28
Figura 10 - Fluxograma da Otimização por Enxame de Partículas.....	30
Figura 11 - Sistema de Inferência <i>Fuzzy</i>	32
Figura 12 - Curva de resposta ao degrau unitário.....	33
Figura 13 - Gerador de Vapor Aquatubular.....	39
Figura 14 - Validação entre modelo e o processo para a saída de pressão.....	41
Figura 15 - Validação entre modelo e o processo para a saída de nível de água.....	41
Figura 16 - Sistema de controle do gerador de vapor.....	42
Figura 17 - Fluxograma do ambiente CE.....	45
Figura 18 - Desempenho do algoritmo PSO.....	47
Figura 19 - Desempenho do AG.....	48
Figura 20 - Função de pertinência da entrada “erro” – pressão e nível.....	49
Figura 21 - Função de pertinência da entrada “derivada do erro” – pressão e nível.....	49
Figura 22 - Função de pertinência da ação de controle “fluxo de combustível e fluxo de água”.....	50
Figura 23 - Saída de pressão de vapor para mudança na referência com controle RNA.....	52
Figura 24 - Saída de nível de água para mudança na referência com controle RNA.....	52
Figura 25 - Fluxo de combustível para mudança na referência com controle RNA.....	53
Figura 26 - Fluxo de água para mudança na referência com controle RNA.....	53
Figura 27 - Saída de pressão de vapor para mudança na referência com controle PSO PID...54	
Figura 28 - Saída de nível de água para mudança na referência com controle PSO PID.....	54
Figura 29 - Fluxo de combustível para mudança na referência com controle PSO PID.....	54
Figura 30 - Fluxo de água para mudança na referência com controle PSO PID.....	55

Figura 31 - Saída de pressão de vapor para mudança na referência com controle AG PID.....	55
Figura 32 - Saída de nível de água para mudança na referência com controle AG PID.....	55
Figura 33 - Fluxo de combustível para mudança na referência com controle AG PID.....	56
Figura 34 - Fluxo de água para mudança na referência com controle AG PID.....	56
Figura 35 - Saída de pressão de vapor para mudança na referência com controle <i>Fuzzy</i>	57
Figura 36 - Saída de nível de água para mudança na referência com controle <i>Fuzzy</i>	57
Figura 37 - Fluxo de combustível para mudança na referência com controle <i>Fuzzy</i>	57
Figura 38 - Fluxo de água para mudança na referência com controle <i>Fuzzy</i>	58
Figura 39 - Saída de pressão de vapor para mudança na referência.....	58
Figura 40 - Saída de nível de água para mudança na referência.....	59
Figura 41 - Fluxo de combustível para mudança na referência.....	59
Figura 42 - Fluxo de água para mudança na referência.....	59
Figura 43 - Saída de pressão de vapor na presença de distúrbios.....	62
Figura 44 - Saída de nível de água na presença de distúrbios.....	62
Figura 45 - Fluxo de combustível frente aos distúrbios.....	63
Figura 46 - Fluxo de água frente aos distúrbios.....	63

LISTA DE TABELAS

Tabela 1 - Parâmetros do Gerador de Vapor Aquatubular.....	40
Tabela 2 - Ganhos dos controladores PID.....	46
Tabela 3 - Parâmetros do Algoritmo PSO.....	46
Tabela 4 - Parâmetros do AG.....	47
Tabela 5 - Índices de desempenho para saída de pressão para mudança na referência.....	60
Tabela 6 - Índices de desempenho para saída de nível para mudança na referência.....	61
Tabela 7 - Índices de desempenho para saída de pressão frente ao distúrbio.....	63
Tabela 8 - Índices de desempenho para saída de nível frente ao distúrbio.....	64

LISTA DE QUADROS

Quadro 1 - Matriz de regras para o controle de pressão.....	51
Quadro 2 - Matriz de regras para o controle de nível.....	51

LISTA DE ABREVIATURAS

AG / GA	Algoritmo Genético (<i>Genetic Algorithm</i>)
ANFIS	Sistema de Inferência Neuro-Fuzzy Adaptativo (<i>Adaptive Neuro-Fuzzy Inference System</i>)
CE	Computação Evolutiva
CNN	Redes Neurais Convolucionais (<i>Convolutional Neural Networks</i>)
DDWMR	Robô Móvel com Rodas de Tração Diferencial (<i>Differential Drive Wheeled Mobile Robot</i>)
DNN	Redes Neurais Profundas (<i>Deep Neural Networks</i>)
ESO	Observador de Estados Estendidos (<i>Extended State Observer</i>)
FIS	Sistema de Inferência Fuzzy (<i>Fuzzy Inference System</i>)
FLC	Controlador Lógico Fuzzy (<i>Fuzzy Logic Controller</i>)
FRBS	Sistemas de Base de Regras Fuzzy (<i>Fuzzy Rule-Based Systems</i>)
GRU	Unidade Recorrente de Porta (<i>Gated Recurrent Unit</i>)
GV	Gerador de Vapor
IA	Inteligência Artificial
IAE / ISE	Índices de Erro (Integral do Erro Absoluto / Integral do Erro Quadrático)
ITAE / ITSE	Índices de Erro ponderados pelo Tempo
KL	Divergência de Kullback-Leibler
LSTM	Longa Memória de Curto Prazo (<i>Long Short-Term Memory</i>)
MAE / MSE	Erro Médio Absoluto / Erro Quadrático Médio
MF	Funções de Pertinência (<i>Membership Functions</i>)
MLP	Perceptron Multicamadas (<i>Multilayer Perceptron</i>)
NARMA	Média Móvel Autorregressiva Não Linear (<i>Non-linear Auto-Regressive Moving Average</i>)

PID / PI / PD	Controladores (Proporcional Integral Derivativo e variações)
PPO	Otimização de Política Próxima (<i>Proximal Policy Optimization</i>)
PSO	Otimização por Enxame de Partículas (<i>Particle Swarm Optimization</i>)
ReLU	Unidade Linear Retificada (<i>Rectified Linear Unit</i>)
RL	Aprendizado por Reforço (<i>Reinforcement Learning</i>)
RMSP	Raiz do Erro Quadrático Médio (<i>Root Mean Square Propagation</i>)
RNA	Redes Neurais Artificiais
RNN	Redes Neurais Recorrentes (<i>Recurrent Neural Networks</i>)
SAC	<i>Soft Actor-Critic</i> (Algoritmo de Aprendizado por Reforço)
SVM	Máquinas de Vetores de Suporte (<i>Support Vector Machines</i>)

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Justificativa.....	9
1.2	Objetivos.....	10
1.2.1	Geral.....	10
1.2.2	Especificos	10
1.3	Estrutura da Monografia	10
2	FUNDAMENTAÇÃO TEÓRICA.....	11
2.1	Gerador de vapor	11
2.1.1	Geradores de Vapor do tipo Aquatubular	12
2.1.2	Geradores de Vapor Fumotubular	14
2.2	Redes neurais artificiais	15
2.2.1	Introdução	15
2.2.2	Arquitetura e treinamento das redes neurais.....	18
2.3	Controle PID.....	25
2.4	Lógica <i>Fuzzy</i>	30
2.5	Critérios de desempenho	32
2.6	Estado da arte.....	34
3	CONTROLE.....	38
3.1	Caso de estudo	38
3.2	Controle do gerador de vapor	40
3.2.1.	Controle com RNA.....	41
3.2.2.	Controle PID	45
3.2.3.	Controle <i>fuzzy</i>	47
4	RESULTADOS.....	51
4.1	Resultado do controle com RNA	51
4.2	Resultado do controle PID	52
4.3	Resultado do controle <i>Fuzzy</i>.....	55
4.4	Análise do desempenho dos controladores	57
5	CONCLUSÕES E TRABALHOS FUTUROS.....	64
	REFEÊNCIAS.....	66

1 INTRODUÇÃO

O controle e a automação de processos industriais constituem pilares fundamentais para a competitividade e a sustentabilidade da manufatura moderna. Em um cenário global que exige máxima eficiência, qualidade padronizada e segurança operacional, a implementação de sistemas automatizados deixa de ser uma vantagem para se tornar uma necessidade. Esses sistemas garantem a precisão na execução de rotinas complexas, otimizam o consumo de recursos energéticos e materiais, e mitigam riscos através do monitoramento contínuo e de respostas rápidas a anomalias, assegurando a proteção de colaboradores, ativos e do meio ambiente.

Dentro desse universo, os geradores de vapor (GV), comumente conhecidos como caldeiras, destacam-se como equipamentos críticos, cuja operação manual é inviável dada a sua dinâmica complexa e os perigos intrínsecos. O controle automatizado torna-se, portanto, imprescindível para manter o equilíbrio dinâmico entre a geração e o consumo de vapor, gerenciando de forma integrada e em tempo real variáveis interrelacionadas como pressão, nível de água e temperatura (Marques, 2005).

Os geradores de vapor são equipamentos termodinâmicos essenciais em setores como geração de energia e processos industriais, responsáveis pela produção de vapor em condições controladas de pressão, temperatura e vazão. Sua operação segura e eficiente, contudo, é um desafio complexo devido à natureza dinâmica do sistema, sujeito a perturbações e a fortes interações entre suas principais variáveis. Nesse contexto, as estratégias de controle atuam como elemento central para garantir o equilíbrio entre os fluxos de massa e energia, assegurando estabilidade, prevenindo condições perigosas e otimizando o desempenho. Muitos trabalhos abordam diferentes controladores para o controle eficiente dos geradores de vapor (Li *et al.*, 2023; Hussein *et al.*, 2024; Guimarães, 2023; Cervantes, 2002; Queiroz, 2017).

1.1 JUSTIFICATIVA

Os geradores de vapor apresentam desafios significativos de controle, como interações multivariáveis, comportamento não-linear, múltiplas variáveis acopladas, incertezas operacionais e a influência de distúrbios externos, como variações na demanda de vapor e na temperatura da água de entrada. A complexidade dinâmica dos geradores de vapor tem motivado o avanço de estratégias de controle capazes de lidar com esses comportamentos (Jiibril; Tadese; Tadese, 2020; Hussein *et al.*, 2024).

O controle dos geradores de vapor é tradicionalmente baseado em controladores PID, valorizados pela simplicidade, baixo custo e facilidade de implementação (Liu *et al.*, 2024). No entanto, PIDs sintonizados por métodos tradicionais possuem um desempenho limitado em processos com forte não linearidade, grande inércia e acoplamento de variáveis, exigindo sintonia complexa e pouco adaptativa. Estratégias mais avançadas buscam superar essas limitações. A lógica *Fuzzy* tem se mostrado uma alternativa eficaz ao controle clássico ao incorporar regras heurísticas derivadas da experiência operacional, apresentando bons resultados em sistemas de alimentação de água e controle de nível em caldeiras (Liu *et al.*, 2024). Por outro lado, estratégias de otimização multiobjetivo vem sendo aplicadas a caldeiras de leito fluidizado para equilibrar eficiência, estabilidade e redução de emissões, evidenciando o potencial de técnicas avançadas para aprimorar o desempenho global do processo (Chen; Yin; Chen, 2025).

Nesse contexto, as Redes Neurais Artificiais (RNAs) surgem como uma alternativa promissora por sua capacidade de aprender o comportamento dinâmico a partir de dados e lidar intrinsecamente com não linearidades e distúrbios, permitindo ajustes automáticos. Essas redes têm sido utilizadas em aplicações envolvendo tanto o controle quanto a previsão de variáveis operacionais, contribuindo para ajustes dinâmicos e eficientes nos processos industriais (Liu *et al.*, 2024; Windmann *et al.*, 2024).

Desta forma, torna-se interessante a exploração de diferentes estratégias de controle em geradores de vapor, visando a proposta de arquiteturas inteligentes capazes de identificar e compensar as não-linearidades do processo e atuar eficientemente como controladores adaptativos ou modelos internos preditivos. Este trabalho concentra-se justamente na análise das estratégias de controle aplicadas a estes sistemas, investigando as arquiteturas dos controladores por RNA, PID e logica *Fuzzy*, além dos seus desafios e contribuições para uma operação segura, estável e eficiente.

1.2 OBJETIVOS

1.2.1 Geral

Realizar um estudo comparativo de estratégias de controle num gerador de vapor.

1.2.2 Específicos

- Estudar o funcionamento e o modelo do gerador de vapor.
- Estudar as estratégias de controle aplicadas no GV.
- Desenhar e implementar via simulação estratégias de controle num gerador de vapor, para um desempenho satisfatório do sistema.

1.3 ESTRUTURA DA MONOGRAFIA

A monografia está estruturada em cinco capítulos. O capítulo 1 apresenta uma introdução geral, mostrando a justificativa e os objetivos. No capítulo 2 é apresentada a fundamentação teórica sobre o gerador de vapor e as diversas estratégias de controle estudadas. O desenho das estratégias de controle implementadas no gerador de vapor mostra-se no capítulo 3. Finalmente, no capítulo 4 apresentam-se os resultados de simulação dos controladores neural, PID e *Fuzzy* no GV, utilizando linguagem de programação *Python*. No capítulo 5 tem-se as conclusões a respeito do trabalho e alguns comentários sobre possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica necessária para o desenvolvimento deste trabalho é apresentada nesta seção. Serão abordados os princípios operacionais dos geradores de vapor e a base conceitual das RNAs. Em seguida, serão abordados os controladores: PID e *Fuzzy*.

2.1 GERADOR DE VAPOR

O gerador de vapor representa um sistema complexo amplamente utilizado para transformar a energia contida em combustíveis em energia térmica, a qual é transferida à água para a geração de vapor. Este vapor pode ser utilizado em diversos processos, como aquecimento, acionamento de turbinas, esterilização e, principalmente, na geração de energia elétrica.

O funcionamento de um gerador de vapor envolve uma série de elementos, como a câmara de combustão, a caldeira, os queimadores, superfícies auxiliares de transferência de calor e outros dispositivos complementares. A combustão do combustível ocorre na fornalha, em que o ar primário é introduzido junto ao combustível por meio de queimadores. O ar secundário, previamente aquecido em aquecedores de ar, é insuflado na câmara de combustão por ventiladores de tiro forçado, completando o processo de combustão. Os gases quentes gerados passam por superfícies de transferência de calor, como economizadores, superaquecedores e reaquecedores, em que cedem energia ao fluido de trabalho (geralmente água ou vapor). Após isso, os gases são conduzidos à chaminé com auxílio de ventiladores de tiro induzido.

Os geradores de vapor podem ser classificados de diversas formas (ver Figura 1). Segundo o combustível utilizado, podem empregar combustíveis sólidos, líquidos, gasosos ou combinações, o que afeta o projeto da fornalha e o controle da combustão. Quanto à construção, dividem-se em flamotubulares (fumatubulares), com gases passando por dentro dos tubos, e aquotubulares, em que a água circula internamente, sendo estes últimos mais adequados para altas potências devido à maior eficiência e segurança (Annaratone, 2008). Quanto à circulação, existem os de circulação natural, que se baseiam na diferença de densidade entre água e vapor, os de circulação assistida, que utilizam bombas para melhorar o fluxo e permitir projetos mais compactos, e os de circulação forçada, nos quais toda a água é bombeada e transformada em vapor em um único percurso, permitindo operação em altas pressões, mas exigindo controle e manutenção mais rigorosos.

Com base na razão entre conteúdo de água e superfície aquecida, classificam-se em geradores de alto, médio e baixo conteúdo. Os de alto conteúdo (50 a 100 kg/m²) oferecem inércia térmica e maior estabilidade. Os de conteúdo médio (20 a 50 kg/m²) e baixo exigem maior precisão no controle e qualidade da água, mas permite uma resposta rápida à carga. Quanto à pressão na fornalha, os geradores podem ter fornalhas em depressão, hoje pouco comuns, ou pressurizadas, que predominam nas aplicações modernas por requererem apenas ventiladores insufladores, reduzirem custos, melhorarem o controle e por evitarem infiltrações de ar. Apesar disso, exigem maior cuidado com vedação e resistência estrutural (Annaratone, 2008).

Figura 1 - Classificação dos geradores de vapor.



Fonte: elaborado pelo próprio autor.

2.1.1 Geradores de Vapor do tipo Aquatubular

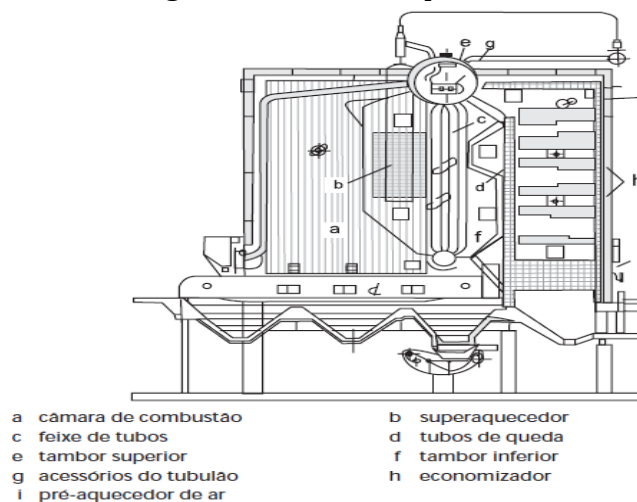
Os geradores de vapor aquatubulares (ver Figura 2) caracterizam-se por conduzir a água dentro de tubos expostos ao calor da combustão ou de gases quentes. Esses geradores são amplamente utilizados quando se deseja produzir vapor em grandes quantidades e pressões elevadas (Annaratone, 2008). Podem ser classificados com base no conteúdo de água, na pressão da fornalha e no modo de aquecimento. Quanto ao conteúdo de água, existem modelos com grande volume (mais tradicionais) e outros com volume reduzido, como os geradores rápidos. Em relação à pressão da fornalha, pode ser pressurizada e vedada ou aberta à atmosfera com tiragem natural.

O modo de aquecimento pode ocorrer por radiação, por convecção ou de forma mista. Nos geradores que operam por radiação, os tubos são expostos diretamente à chama dentro da fornalha, que é cercada por paredes membranadas formadas por tubos conectados a tambores superior e inferior. Essa configuração permite elevada transferência de calor e, portanto, alto rendimento térmico. São comuns em grandes caldeiras industriais e costumam incluir

equipamentos auxiliares como superaquecedores, aquecedores de ar e economizadores (Annaratone, 2008).

Os geradores de calor residual aproveitam o calor sensível dos gases provenientes de outros processos industriais, sem que haja uma fornalha própria. Por essa razão, pertencem à categoria de aquecimento por convecção. Sua estrutura é composta por bancos de tubos conectados a tambores, podendo conter ainda superaquecedores e economizadores no mesmo duto. São aplicados em fornos industriais, motores a combustão e, especialmente, em sistemas de ciclo combinado com turbinas a gás e a vapor. Apesar de simples em construção, requerem atenção quanto à limpeza e manutenção, especialmente quando os gases são sujos ou corrosivos. A circulação pode ser natural ou forçada, e há diferentes arranjos possíveis, como bobinas horizontais empilhadas (Annaratone, 2008).

Figura 2 - Gerador Aquatubular.



Fonte: INSPCAL, 2016.

A seguir se apresentam alguns dos componentes dos geradores aquatubulares:

- **Câmara de combustão e Paredes Tubulares Radiantes:** é a região na qual ocorre a combustão. As paredes são formadas por tubos refrigerados a água que absorvem grande parte do calor por radiação.
- **Tambor:** é o componente no qual se separa a mistura água-vapor, envia o vapor para o superaquecedor e a água de volta à circulação. Pode conter dispositivos internos como separadores e dutos para melhoria da separação.
- **Banco Gerador de Vapor:** é muito importante em unidades de pequeno e médio porte sendo responsável por até 40% da transferência total de calor e está formado por tubos curvados conectados aos tambores.

- **Curvatura dos Tubos:** utilizada na construção de vários componentes como superaquecedores, economizadores e paredes da fornalha. Deve ser feita com cuidado para evitar tensões excessivas e deformações.
- **Expansão dos Tubos:** é a técnica usada para fixar os tubos aos tambores, usando um expansor com rolos que forçam o tubo contra a parede do furo.
- **Superaquecedor e Reaquecedor:** aumentam a temperatura do vapor além da saturação e podem ser do tipo radiante (nas paredes da fornalha) ou por convecção. O reaquecedor aquece novamente o vapor após expansão parcial na turbina.
- **Economizador:** pré-aquece a água de alimentação com os gases quentes antes de entrar no tambor e aumenta a eficiência térmica em até 2–3%. Pode ser feito com tubos de ferro fundido em geradores a óleo.
- **Aquecedor de Ar:** recupera o calor dos gases de combustão para aquecer o ar de entrada. Melhora a eficiência e favorece a combustão e está presente em geradores de todos os portes (Annaratone, 2008).

2.1.2 Geradores de Vapor Fumotubular

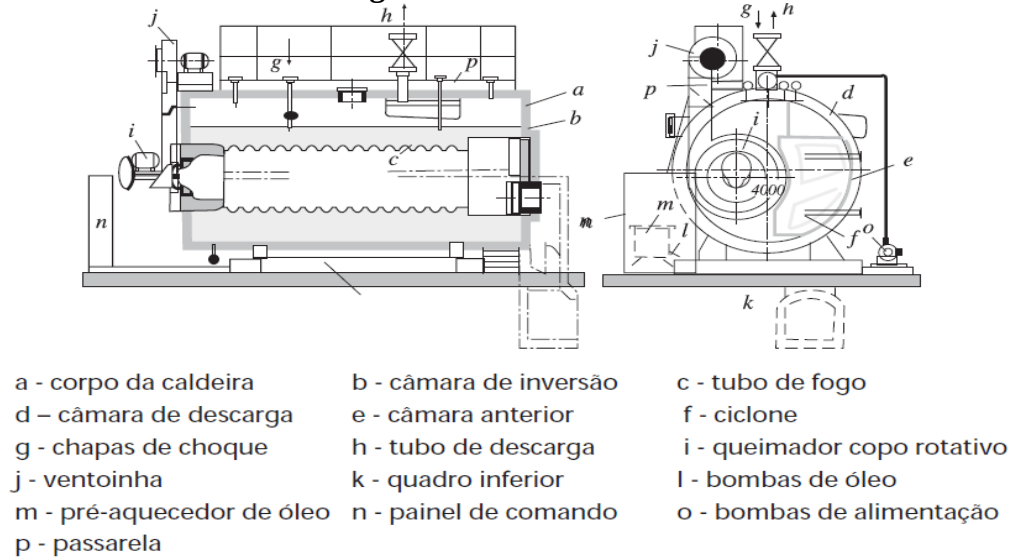
Os geradores fumotubulares (ou caldeiras fumotubulares/flamotubulares) (ver Figura 3) são equipamentos nos quais os gases quentes de combustão percorrem o interior de tubos, enquanto a água a ser aquecida ou vaporizada está localizada externamente aos tubos, dentro de um casco cilíndrico. Podem fornecer tanta água quente (inclusive superaquecida) quanto vapor, para fins de aquecimento ou processos industriais. Esses geradores geralmente apresentam estrutura compacta, sendo construídos com três passagens de gases, o que aumenta a eficiência térmica. Os gases atravessam os tubos em diferentes direções antes de serem liberados pela chaminé. Existem dois tipos principais (Annaratone, 2008):

- **Placa seca:** região em que os gases retornam em uma câmara sem contato com água.
- **Placa molhada:** região em que essa câmara é refrigerada pela própria água da caldeira, melhorando a durabilidade do equipamento.

Como características principais de operação estão: apresentam altas cargas térmicas volumétricas e superficiais, superiores às dos geradores tubulares de água; necessitam de excesso de ar na combustão, especialmente ao utilizar óleo combustível, para evitar a formação de monóxido de carbono (CO), o que impacta a eficiência; a chama deve ser estreita e longa devido ao espaço reduzido, limitando a turbulência e dificultando a mistura ideal de ar e combustível; a manutenção é essencial, com inspeções periódicas e limpeza interna dos tubos para manter a eficiência; são sensíveis a variações de carga térmica, pois a eficiência depende

fortemente da velocidade dos gases nos tubos; o processo de geração de vapor ocorre diretamente no casco, diferentemente do que ocorre nos geradores aquatubulares, em que a separação entre vapor e água é mais clara. Por último, há risco de carreamento de água pelo vapor, o que exige proteções internas (como chapas defletoras) e cuidados com a posição da entrada de água fria para evitar choques térmicos.

Figura 3 - Gerador Flamotubular.



Fonte: INSPCAL, 2016.

2.2 REDES NEURAIIS ARTIFICIAIS

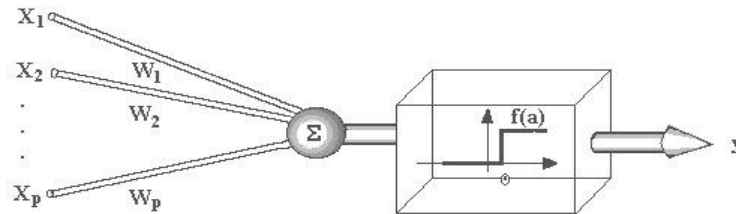
2.2.1 Introdução

As redes neurais artificiais, inspiradas no cérebro humano, aprendem relações de entrada e saída por meio do ajuste de pesos sinápticos, sendo amplamente aplicadas em áreas como engenharia, medicina e agronomia. O campo surgiu em 1943 com McCulloch e Pitts, avançou com a lei de aprendizado de Hebb (1949) e o neurocomputador Snark de Minsky (1951), e consolidou-se com o Perceptron de Rosenblatt (1957-1958) e o Adaline de Widrow, ainda utilizado hoje. Após um período de baixa credibilidade, a área ganhou força com eventos como a *IEEE International Conference on Neural Networks* (1987) e permanece em contínuo desenvolvimento (Furtado, 2019).

Um neurônio artificial possui três características principais: entradas, pesos e função de ativação. As entradas, vindas do ambiente ou de outros neurônios, são ponderadas pelos pesos, que determinam sua relevância para a saída da rede. A soma ponderada dessas entradas é processada por uma função de ativação, que decide gerar ou não uma saída. O modelo proposto por McCulloch e Pitts (Figura 4) apresenta um neurônio de processamento binário capaz de

realizar operações lógicas complexas, descrito matematicamente pelas Equações 1 e 2, sendo X_p são os sinais de entrada, W_p os pesos sinápticos, $f(a)$ a função de ativação e y sinais de saída (Furtado, 2019).

Figura 4 – Esquema do neurônio da Unidade Processadora de McCulloch e Pitts.



Fonte: Furtado, 2019

$$u_p = \sum_{p=1}^m W_p X_p \quad (1)$$

$$y_p = f(u_p) \quad (2)$$

As funções de ativação, apresentadas na Figura 5, são componentes essenciais em redes neurais artificiais, determinando a saída de um neurônio com base em suas entradas ponderadas. A escolha apropriada da função de ativação impacta diretamente a capacidade da rede em modelar relações complexas nos dados. Dentre as diversas funções existentes, destacam-se:

Função Linear: produz uma saída diretamente proporcional à entrada, sem introduzir não linearidade no modelo. Embora simples, sua capacidade de modelar relações complexas é limitada, tornando-a mais adequada para problemas em que as relações entre variáveis são essencialmente lineares (Rizzo; Canato, 2020) e é representada pela equação 3, na qual a saída $f \in \mathbb{R}, \forall x$.

$$f(x) = x \quad (3)$$

Função Sigmoid: caracterizada por sua forma em "S", a função sigmoide mapeia qualquer valor de entrada para um intervalo entre 0 e 1. Isso é particularmente útil em tarefas de classificação binária, em que a saída representa a probabilidade de pertencimento a uma classe específica. No entanto, a função Sigmoid pode sofrer com o problema do "desvanecimento do gradiente" durante o treinamento de redes profundas, dificultando a convergência (Fleck *et al.*, 2016). A função Sigmoid é representada pela equação 4, na qual β

é a referência que define a inclinação (ganho) da função.

$$f(x) = \frac{1}{1 + e^{(-\beta x)}} \quad (4)$$

Função ReLu (*Rectified Linear Unit*): definida como zero para entradas negativas e linear para entradas positivas, a ReLu introduz não linearidade de forma simples e eficiente. Sua popularidade deve-se à eficácia em redes profundas, pois mitiga o problema do desvanecimento do gradiente e acelera a convergência durante o treinamento. Contudo, a ReLu pode enfrentar o problema de "neurônios mortos" quando muitas unidades ficam inativas, não contribuindo para o aprendizado (Rizzo; Canato, 2020). A função ReLu é representada pela equação 5.

$$f(x) = \max(0, x) \quad (5)$$

Função Tangente Hiperbólica: para qualquer valor de x a saída ficará entre -1 e 1 e é representada pela equação 6.

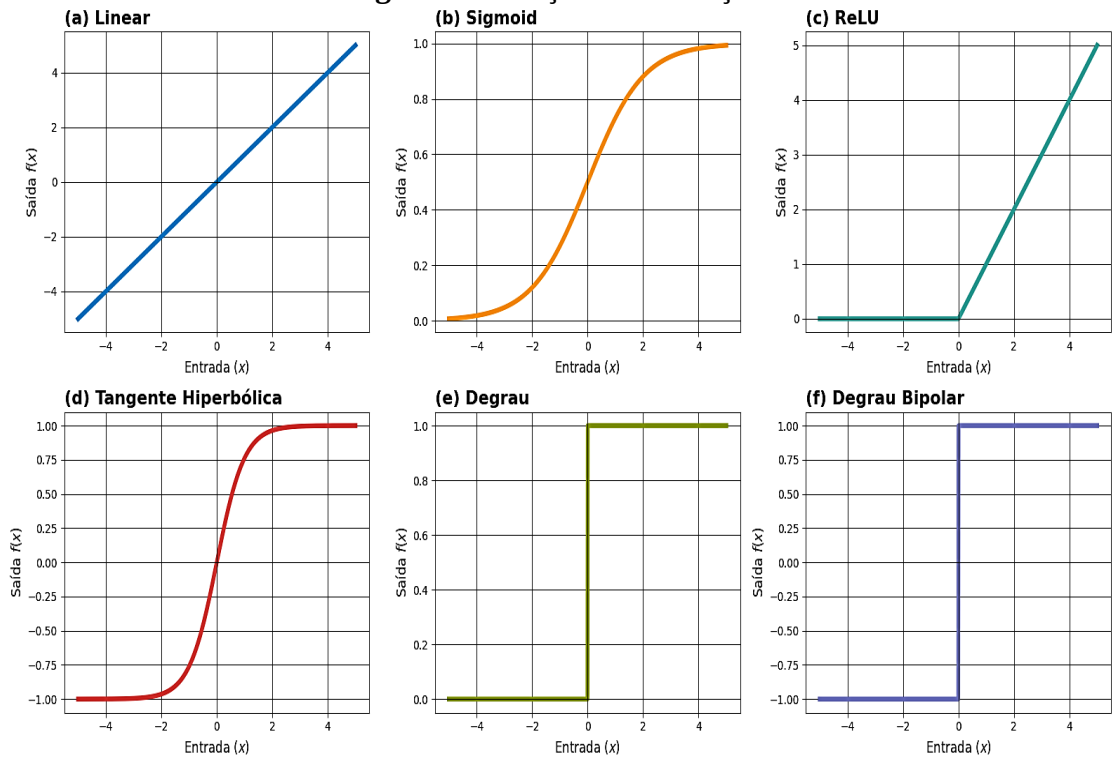
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (6)$$

Função Degrau: para valores de x maiores ou igual a zero temos a saída igual a 1, para valores negativos temos a saída igual a zero. A função degrau é representada pela equação 7.

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases} \quad (7)$$

Função Degrau Bipolar: para valores positivos de x a saída é igual a 1 e para valores negativos a saída é igual a -1. A função degrau bipolar é representada pela equação 8.

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ -1 & \text{se } x < 0 \end{cases} \quad (8)$$

Figura 5 – Funções de Ativação.

Fonte: Autor, 2025

As funções utilizadas nesse trabalho foram a ReLU e Linear, pois essa combinação é ideal para redes neurais que realizam regressão. A ReLU é escolhida para as camadas ocultas por introduzir a não-linearidade de maneira eficiente e, crucialmente, evitar o problema do gradiente evanescente (*vanishing gradient*) que afeta funções como a Sigmoide e tanh, garantindo um treinamento rápido e estável. Já a Função Linear é empregada na camada de saída da rede, pois não restringe o valor de predição, sendo essencial para tarefas de regressão em que a saída deve ser um valor contínuo e ilimitado.

2.2.2 Arquitetura e Treinamento das Redes Neurais

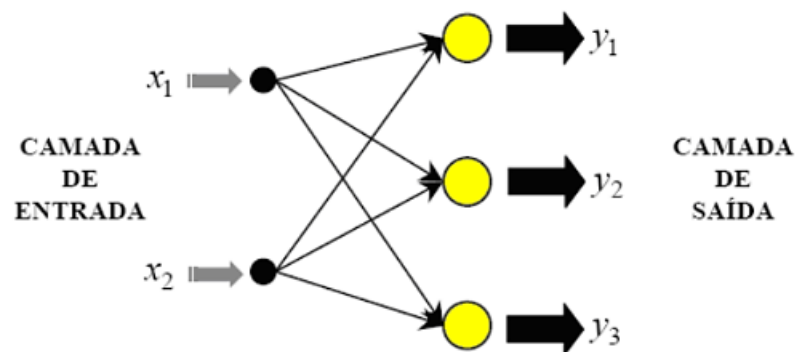
As redes neurais artificiais são organizadas em camadas (entrada, oculta e saída), podendo ser estáticas ou dinâmicas (Finocchio, 2014). A camada de entrada recebe os dados, a oculta processa informações complexas, e a de saída produz os resultados finais. As redes se classificam pela arquitetura, topologia, características dos nós e regras de treinamento. Dentre as arquiteturas, as Redes Neurais Recorrentes (RNNs) e suas variantes, como LSTM (*Long Short-Term Memory*) e GRU (*Gated Recurrent Unit*), se destacam como os exemplos mais proeminentes de arquiteturas dinâmicas ou recorrentes. Essas arquiteturas são projetadas especificamente para lidar com dados sequenciais (como texto, fala, ou séries temporais)

porque, ao contrário das redes *feedforward*, elas possuem conexões de *feedback* ou memória interna. Isso permite que a rede leve em consideração o contexto e o histórico das entradas anteriores ao processar a entrada atual, tornando-as ideais para tarefas que exigem compreensão de dependências temporais longas, como tradução automática e reconhecimento de voz. Outras arquiteturas notáveis incluem as Redes Neurais Convolucionais (CNNs), que são redes *feedforward* especializadas em processar dados com topologia de grade (como imagens e vídeos).

As redes do tipo *feedforward*, são as redes no qual fazem a informação fluir apenas em uma direção, da entrada para a saída, sem ciclos ou realimentação. Isso simplifica o treinamento e a aplicação da rede, garantindo que os dados de treinamento sejam independentes uns dos outros, sem depender de entradas ou saídas anteriores. Um exemplo clássico de rede neural *feedforward* é o Perceptron e o MLP (*Multilayer Perceptron*). A seguir algumas características dessas redes:

Perceptron de camada única: a rede neural artificial mais simples é o perceptron de camada única (Figura 6) que foi proposto por Rosenblatt em 1957 (Finocchio, 2014). Suas principais características são as entradas que são multiplicadas pelos pesos sinápticos, um somador que soma o conjunto de entradas multiplicados pelos pesos correspondentes somados a um viés e uma função de ativação que limita a amplitude do sinal do somador e pode ativar o neurônio ou não. O perceptron como a unidade de rede mais simples pode ser utilizado para criação de camadas complexas através da ligação e aninhamento de vários perceptrons. Essas camadas mais complexas podem, por sua vez, formar arquiteturas de redes neurais mais sofisticadas.

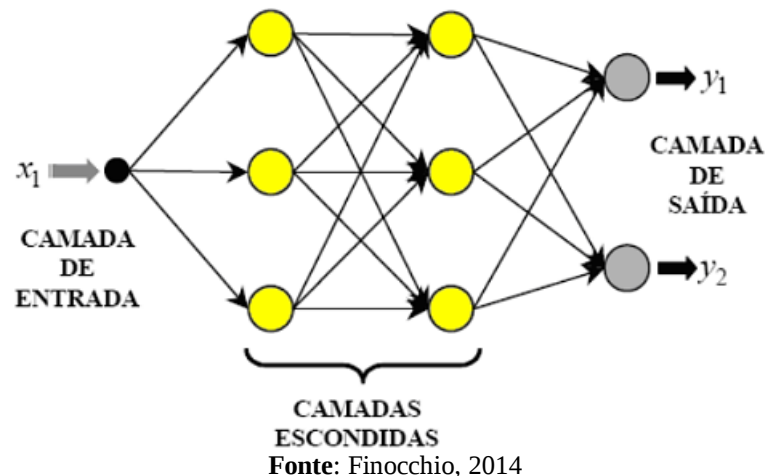
Figura 6 - Perceptron de camada única.



Fonte: Finocchio, 2014

Multilayer Perceptron: é uma estrutura de rede neural com pelo menos uma camada oculta (Figura 7), sendo capaz de aprender a representar problemas não lineares e, quanto mais complexa a rede, mais desafiador é o processo de atualização dos pesos. Uma das aplicações notáveis dos MLPs é em problemas de regressão linear. Como afirmado em (Bonala, 2009), os MLPs podem superar as limitações dos perceptrons mais simples.

Figura 7 - Estrutura do Multilayer Perceptron.



Treinamento de Redes Neurais Artificiais: a principal característica das redes neurais é sua capacidade de aprender por meio do ambiente e melhorar seu desempenho. Esse aprendizado é um processo iterativo de ajuste dos pesos da rede, o qual termina quando a rede consegue generalizar soluções para uma classe de problemas (Finocchio, 2014).

O aprendizado de máquina é uma subárea da inteligência artificial que utiliza técnicas estatísticas para capacitar sistemas computacionais a aprender com dados, ao invés de serem programados explicitamente. Esse aprendizado pode ocorrer de diversas maneiras, dependendo dos dados e do algoritmo utilizado. As principais categorias de aprendizado incluem (Figueiredo; Rejaili, 2018):

- **Aprendizado supervisionado:** a rede recebe um conjunto de dados de entrada e saída conhecidos, ajustando seus pesos para minimizar a diferença entre a saída prevista e a real, geralmente avaliada por uma função de perda.
- **Aprendizado não supervisionado:** aqui, a rede recebe apenas dados de entrada e o objetivo é aprender padrões e estruturas desses dados.
- **Aprendizado por reforço:** um agente interage com o ambiente e aprende a maximizar recompensas ao longo do tempo, através da otimização de uma política de ações. A principal abordagem de treinamento nesse método é o *Q-Learning*, que busca

determinar a melhor ação para cada estado com base em uma tabela de qualidade (*Q-table*), ajustando os valores de ação com a equação do Q-St.

Dentro de outros algoritmos do aprendizado por reforço, podemos citar:

Os métodos *actor-critic* do aprendizado por reforço representam uma classe de algoritmos que combinam duas estruturas fundamentais: a política do agente (*actor*) e a função de valor (*critic*). O *actor* é responsável por selecionar ações com base no estado do ambiente, enquanto o *critic* avalia a qualidade dessas ações, fornecendo um sinal de aprendizado para a melhoria da política (Sutton; Barto, 2018).

O *Soft Actor-Critic* (SAC) é uma variante do método *actor-critic* que incorpora um conceito fundamental: a maximização da entropia. Diferente dos métodos tradicionais que apenas buscam maximizar a soma das recompensas ao longo do tempo, o SAC introduz um termo de entropia que incentiva o agente a explorar múltiplas estratégias antes de convergir para uma política final. Essa característica o torna um dos algoritmos mais robustos para aprendizado profundo em ambientes de ação contínua (Haarnoj *et al.*, 2018).

No aprendizado por reforço tradicional, a política aprendida de modo a maximizar a expectativa da soma das recompensas acumuladas (Sutton; Barto, 2018):

$$J(\pi) = \sum_t E_{s_t, a_t} [r(s_t, a_t)] \quad (9)$$

em que, E é a esperança (ou valor esperado) e $r(s_t, a_t)$ representa a recompensa obtida ao executar a ação a_t no estado s_t . O SAC modifica essa função objetivo introduzindo um termo de entropia, resultando na seguinte formulação (Haarnoj *et al.*, 2018):

$$J(\pi) = \sum_t E_{s_t, a_t} [r(s_t, a_t) + \alpha H(\pi(\cdot | s_t))] \quad (10)$$

Sendo, $H(\pi(\cdot | s_t))$ denota a entropia da política, que mede a incerteza na escolha das ações, e α é um hiperparâmetro que equilibra a maximização da recompensa e a maximização da entropia. Esse termo permite que o agente mantenha políticas diversificadas, evitando convergência prematura para soluções subótimas e incentivando uma melhor exploração do espaço de estados e ações.

A função Q , ou função de ação-valor, mede a recompensa esperada ao tomar uma determinada ação a no estado s , seguindo a política. Formalmente, ela é definida como:

$$Q(s, a) = E[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a] \quad (11)$$

em que, γ é um fator de desconto (entre 0 e 1) que pondera a importância das recompensas futuras (Bertsekas, 2012).

No SAC, duas redes Q são utilizadas em vez de uma única, reduzindo o viés positivo e melhorando a estabilidade do treinamento (Haarnoj *et al.*, 2018). Durante o processo de aprendizado, o mínimo entre as duas funções Q é usado para calcular os gradientes da política. O SAC utiliza redes neurais profundas (*Deep Neural Networks* - DNNs) para representar tanto a política quanto a função de valor Q (Haarnoj *et al.*, 2018). Cada uma dessas redes desempenha um papel específico:

- Rede Actor (Ator): modela a política ($a|s$), que define como as ações são escolhidas com base nos estados observados; aprende a selecionar ações que maximizam a soma da recompensa e a entropia e no SAC, a política é estocástica, ou seja, as ações são amostradas de uma distribuição probabilística em vez de serem determinísticas.
- Rede Critic (Crítico): modela a função $Q(s,a)$, avaliando a qualidade das ações escolhidas; utiliza duas redes críticas para reduzir o viés positivo e melhorar a estabilidade e durante o treinamento, a atualização dos pesos da rede crítica é feita com base em amostras de um *replay buffer*.
- Redes Target (Alvo): são redes auxiliares utilizadas para estabilizar as atualizações da função Q e utilizam a média móvel exponencial das redes críticas para suavizar as atualizações e reduzir oscilações.

A divergência de *Kullback-Leibler* (KL) mede a diferença entre duas distribuições de probabilidade e é usada no SAC para comparar a distribuição atual da política com a distribuição *softmax* da função Q (Kullback; Leibler, 1951). Essa abordagem permite que a política se aproxime gradualmente de uma distribuição ótima de ações, resultando em um aprendizado mais estável e eficiente. A política é atualizada minimizando essa divergência, ajustando os parâmetros da rede *actor* para garantir que a distribuição de ações se alinhe melhor com as recompensas esperadas fornecidas pela rede *critic*.

A seguir se apresentam os algoritmos de treinamento referentes aos processos das redes neurais:

Método de Levenberg-Marquardt: é uma técnica de aprendizagem que visa ajustar um modelo matemático a um conjunto de dados, minimizando a diferença entre os valores previstos e reais, utilizando a função residual. A cada iteração, calcula-se a direção para ajustar os parâmetros e melhorar o modelo, resolvendo um sistema linear (Equação 12) (Schwertner,

2019). Em que, $J(X_{K'})$ representa a matriz Jacobiana das funções residuais em relação aos parâmetros, $J(X_{K'})^T$ é a sua transposta, $\lambda_{K'}$ denota o parâmetro de amortecimento que controla a transição entre os métodos de Gauss-Newton e o gradiente descendente, I é a matriz identidade, d_{LM} corresponde ao vetor de atualização dos parâmetros buscados e $R(X_{K'})$ é o vetor de resíduos do sistema no ponto atual.

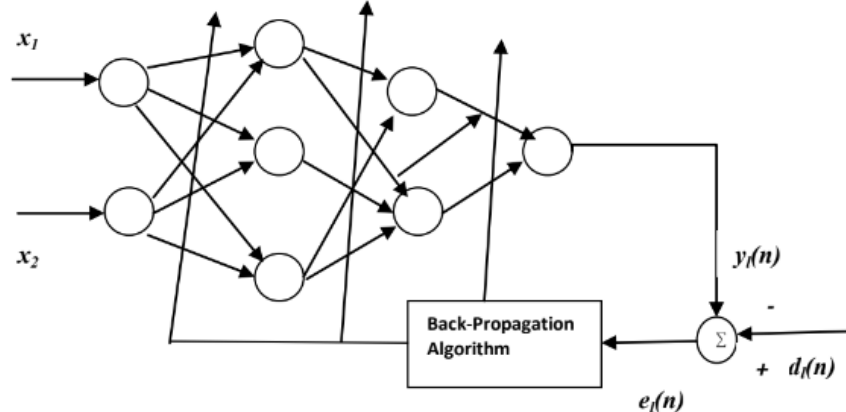
$$(J(X_{K'})^T J(X_{K'}) + \lambda_{K'} I) d_{LM}(\lambda_{K'}) = -J(X_{K'})^T R(X_{K'}) \quad (12)$$

BackPropagations: calcula as derivadas parciais da função de custo em relação aos parâmetros da rede, propagando as respostas da camada de saída até a primeira camada. Ajusta os parâmetros da rede com base no gradiente da função de custo, visando minimizar o erro. As funções de custo são essenciais para medir o erro ou perda durante o treinamento de redes neurais, sendo as mais comuns o MAE (erro absoluto médio) e o MSE (diferença quadrática média). O MAE (Equação 13) prioriza erros menores (0-1), enquanto o MSE (Equação 14) dá mais peso a valores maiores. A escolha da função de erro é crucial para o sucesso do treinamento (Schwertner, 2019). Durante o treinamento, o uso de algoritmos de otimização como o gradiente descendente ajusta os pesos para minimizar a função de custo, com o gradiente indicando a direção de menor erro. Além disso, técnicas como *Momentum*, *AdaGrad*, *RMSProp* e *Adam* aprimoram a convergência de diferentes maneiras, com o algoritmo *Adam* destacando-se por adaptar a taxa de aprendizado e incorporar momentum, tornando-o eficaz para redes neurais. O esquema se mostra na Figura 8.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}_i| \quad (13)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2 \quad (14)$$

Figura 8 – Algoritmo de *BackPropagation*.



Fonte: Bonala, 2009

O processo de desenvolvimento das RNA's é realizado com a sequência das seguintes etapas (Finocchio, 2014):

1. *Coleta de dados*: se coletam os dados relativos ao problema em questão, requer uma cuidadosa análise do problema, usando técnicas de amostragem de forma a minimizar ambiguidades e erros nos dados. Os dados coletados devem ser significativos e cobrir amplamente o domínio do problema.

2. *Separação em conjuntos*: os dados são separados em duas categorias: (i) Dados de treinamento, utilizados para o treinamento da rede e (ii) Dados de teste, utilizados para verificar o desempenho referente às condições reais de utilização e a capacidade de generalização da rede. Os conjuntos são colocados em ordem aleatória para a prevenção de tendências associadas à ordem da apresentação dos dados. Pode ser necessário pré processar estes dados, através das normalizações, escalonamentos e conversões de formato para torná-los mais apropriados à sua utilização de rede.

3. *Configuração da rede*: pode ser conceituada em três etapas: A seleção do paradigma neural apropriado à aplicação; a determinação da topologia da rede a ser utilizada: o número de quantidade em cada camada, etc. A determinação de parâmetros do algoritmo de treinamento e das funções de ativação dos neurônios, tendo esta etapa como uma de fundamental impacto no sistema resultante.

4. *Treinamento*: seguindo o treinamento de algoritmo escolhido, serão feitos os ajustes para os pesos das conexões. É importante levar em conta, alguns aspectos, como a inicialização da rede, o modo e o tempo de treinamento. Uma escolha mais correta, dos valores iniciais dos pesos da rede pode diminuir o tempo necessário para o treinamento. Geralmente, os valores iniciais dos pesos da rede são números aleatórios pequenos, uniformemente distribuídos dentro de um intervalo já definido. Quanto ao tempo de treinamento, sempre será necessário utilizar

algum elemento de parada. Os critérios de parada do algoritmo de treinamento principais são: o número máximo de ciclos (época), e o erro quadrático médio por ciclo. O final do treinamento se dá quando a rede mostrar uma boa capacidade de generalização e sua margem de erro for mínima em relação a um nível previamente estabelecido.

5. *Teste e Integração*: é utilizado para determinar o desempenho da rede com dados que ainda não foram utilizados. Uma rede só pode ser integrada a um sistema quando for devidamente treinada e avaliada.

2.3 CONTROLE PID

O controlador PID é um dos mais utilizados na indústria devido à sua capacidade de atender satisfatoriamente à maioria dos processos de controle. Desenvolvido por Nicholas Minorsky em 1922, inspirado no controle de navios, e comercializado na década de 1930, o PID evoluiu a partir do simples controle liga/desliga, que aciona a saída máxima ou mínima conforme o erro, mas causa oscilações constantes e desgaste nos equipamentos. Métodos de sintonia clássicos foram introduzidos por Ziegler e Nichols em 1942, e atualmente existem técnicas baseadas em lógica *Fuzzy*, redes neurais e algoritmos evolutivos, permitindo ajustes mais precisos e automáticos (Tavares, 2019).

O controle PID combina três ações: a proporcional, que origina uma atuação de controle corretiva proporcional ao erro; a integral, que elimina o erro estacionário acumulando os desvios ao longo do tempo; e a derivativa, que considera a taxa de variação do erro para melhorar a estabilidade e reduzir o sobressinal. Controladores PI e PD apresentam limitações isoladas, o PI é mais lento, enquanto o PD não elimina erro estacionário, que são superadas pela combinação PID. A implementação pode ser feita em paralelo, com ações independentes somadas ao final, ou em série, sendo que avanços em microprocessadores e amplificadores operacionais tornaram o controle PID paralelo cada vez mais comum em sistemas modernos (Tavares, 2019).

A forma clássica do PID em tempo contínuo é dada pela Equação 15:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (15)$$

em que:

$u(t) \rightarrow$ sinal de controle;

$e(t)=r(t)-y(t) \rightarrow$ erro entre referência $r(t)$ e saída $y(t)$;

$K_p \rightarrow$ ganho proporcional;

$K_i \rightarrow$ ganho integral (ou K_p/T_i , dependendo da notação);

$K_d \rightarrow$ ganho derivativo (ou $K_p T_d$, dependendo da notação).

Aplicando a transformada de Laplace à equação 15, considerando suas condições iniciais nulas, obtêm-se a função transferência do controlador PID:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{K_i}{s} + K_d s \right) \quad (16)$$

A lei de controle PID em tempo discreto é dada pela Equação 17 (Bobál *et al.*, 2006):

$$u(k) = K_p e(k) + K_i T_s \sum_{i=0}^k e(i) + \frac{K_d}{T_s} (e(k) - e(k-1)) \quad (17)$$

em que:

$T_s \rightarrow$ período de amostragem;

$k \rightarrow$ índice do instante de tempo discreto.

Nesta monografia para sintonizar os parâmetros do controlador PID serão usadas duas técnicas de aprendizado de máquina: Algoritmo Genético (AG) e Otimização por Enxame de Partículas (PSO).

O Algoritmo Genético, inspirado na teoria da evolução de Darwin e desenvolvido por John Holland na década de 1970, é um método de busca e otimização que simula o processo de seleção natural. Cada solução potencial para o problema é representada por um cromossomo, composto pelos parâmetros a serem ajustados. Inicialmente, uma população de soluções é gerada aleatoriamente e avaliada por meio de uma função de aptidão, que indica quão boa é cada solução (Tavares, 2019).

Entre as técnicas de seleção de indivíduos no algoritmo genético, a roleta russa é uma das mais conhecidas. Nela, cada indivíduo recebe uma fatia proporcional à sua aptidão em uma “roda” imaginária. A escolha é feita girando a roleta, de modo que indivíduos mais aptos têm maior chance de serem selecionados, mas sem excluir completamente os menos aptos, o que preserva a diversidade da população. A probabilidade de seleção de um indivíduo i é calculada pela Equação 18:

$$P(\text{seleção}_i) = \frac{e^{-\beta \cdot \text{custo}_i}}{\sum_{j=1}^{N_{\text{pop}}} e^{-\beta \cdot \text{custo}_j}} \quad (18)$$

em que, β é o parâmetro que controla a pressão de seleção, custo_i é o valor da função de custo

para o indivíduo i e N_{pop} é o tamanho total da população.

Além desse método, existem outros mecanismos de seleção, como o torneio, a seleção por classificação e a seleção aleatória, que variam no critério de escolha, mas seguem o mesmo objetivo de balancear exploração e diversidade no processo evolutivo (Tavares, 2019).

Cruzamento (Crossover) e Mutação: Após a seleção dos pais, o algoritmo aplica as operações de cruzamento e mutação para gerar os filhos. O cruzamento combina material genético de dois pais para formar novas soluções, enquanto a mutação introduz pequenas alterações aleatórias, garantindo diversidade e a capacidade de explorar novas áreas do espaço de busca. Um dos tipos de cruzamento é o cruzamento linear, no qual a posição dos filhos é uma combinação ponderada das posições dos pais (Equação 19).

$$Filho1 = \alpha \cdot Pai1 + (1 - \alpha) \cdot Pai2 \quad (19)$$

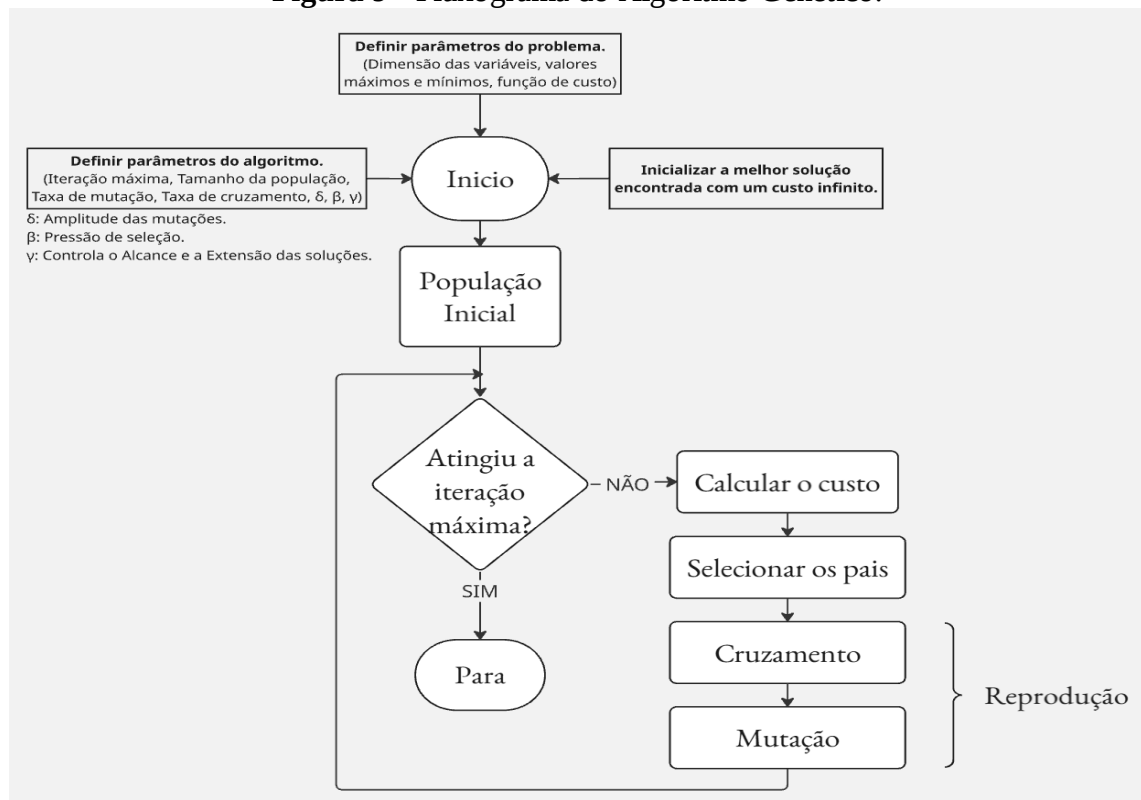
em que, α é um número aleatório gerado no intervalo $(-\gamma, 1+\gamma)$, e γ é um parâmetro de controle que permite ao algoritmo explorar além dos limites dos pais.

A mutação atua como uma força de inovação. Com uma baixa probabilidade, um gene (ou parâmetro) de um indivíduo é alterado aleatoriamente pela adição de um valor (Equação 20).

$$Filho' = Filho + \delta \quad (20)$$

em que, δ é um termo de variação aleatória, retirado de uma distribuição normal (Gaussiana) com desvio padrão σ . Após o cruzamento e mutação, os genes dos filhos são limitados aos valores mínimos e máximos para garantir que a solução permaneça dentro dos limites do problema. O fluxograma do algoritmo genético se mostra na Figura 9.

Figura 9 - Fluxograma do Algoritmo Genético.



Fonte: elaborado pelo próprio autor.

A Otimização por Enxame de Partículas, proposta por Kennedy e Eberhart em 1995, é uma técnica de Inteligência de Enxames inspirada no comportamento social de grupos, como bandos de aves ou cardumes de peixes. Diferente dos algoritmos evolutivos, que utilizam operadores genéticos como mutação e reprodução, o PSO baseia-se na cooperação entre partículas e no uso do histórico de melhores soluções já encontradas. Cada partícula representa uma possível solução para o problema e possui posição, velocidade e memória de sua melhor posição pessoal (*pBest*), além de compartilhar informações sobre a melhor solução global do grupo (*gBest*) (Roncaglio, 2020).

O método utiliza três componentes principais para atualizar o movimento das partículas: a inércia (que mantém parte da direção anterior), a tendência de retornar ao *pBest* e a atração em direção ao *gBest*. O equilíbrio entre exploração (busca global) e exploração local depende dos parâmetros c_1 , c_2 e do coeficiente inercial ω . Ajustar corretamente esses valores é fundamental para que o algoritmo evite mínimos locais e encontre soluções de melhor qualidade. A atualização da velocidade e da posição de cada partícula é realizada a cada iteração (t) com base nas Equações 21 e 22 (Roncaglio, 2020).

$$v_{i,n}(t+1) = \omega v_{i,n}(t) + c_1 r_1 (pBest_{i,n}(t) - x_{i,n}(t)) + c_2 r_2 (gBest_n(t) - x_n) \quad (21)$$

$$x_{i,n}(t+1) = x_{i,n}(t) + v_{i,n}(t+1) \quad (22)$$

em que,

$x_{i,n}(t) \rightarrow$ a posição da partícula;

$\omega v_{i,n}(t) \rightarrow$ o termo de inércia que mantém a partícula se movendo em sua direção atual, permitindo a exploração do espaço de busca;

$c_1 r_1 (pBest_{i,n}(t) - x_{i,n}(t)) \rightarrow$ o termo cognitivo que representa a tendência da partícula de retornar à sua melhor posição pessoal ($pBest$);

$c_2 r_2 (gBest_n(t) - x_n) \rightarrow$ o termo social que representa a tendência da partícula de ser atraída pela melhor posição encontrada por todo o enxame ($gBest$);

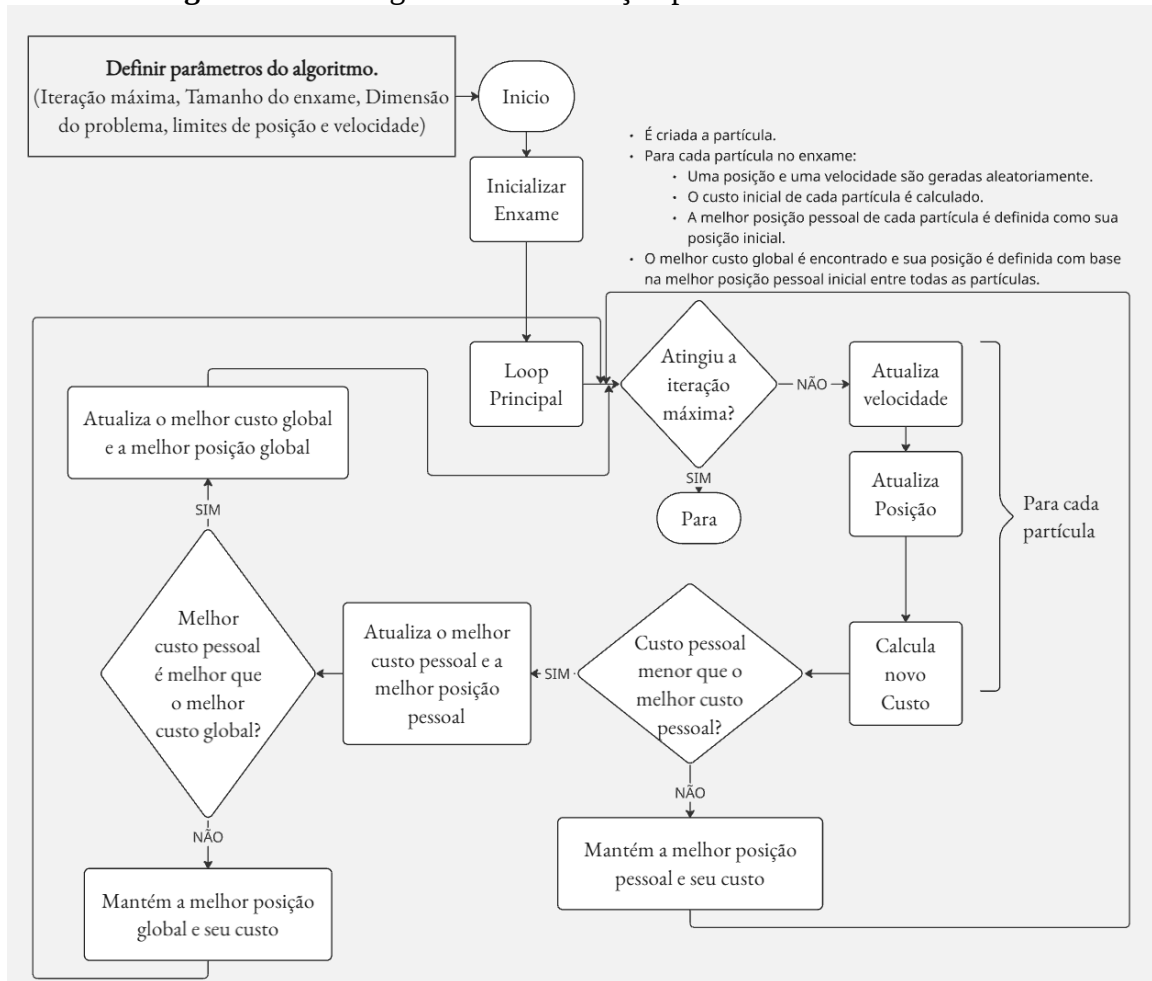
$c_1, c_2 \rightarrow$ são constantes de ponderação;

$r_1, r_2 \rightarrow$ são números aleatórios.

A Equação 23 define uma diminuição linear de ω de um valor máximo (ω_{max}) para um valor mínimo (ω_{min}) durante as I_{max} primeiras iterações. Após esse ponto, ω é mantido constante em ω_{min} , permitindo que o algoritmo comece com mais exploração e depois se concentre na exploração local. O fluxograma do algoritmo de otimização por enxame de partículas se mostra na Figura 10.

$$\omega = \begin{cases} \omega_{max} - \frac{\omega_{max} - \omega_{min}}{I_{max}} I, & \text{se } I \leq I_{max} \\ \omega_{min}, & \text{se } I > I_{max} \end{cases} \quad (23)$$

Figura 10 - Fluxograma da Otimização por Enxame de Partículas.



2.4 LÓGICA FUZZY

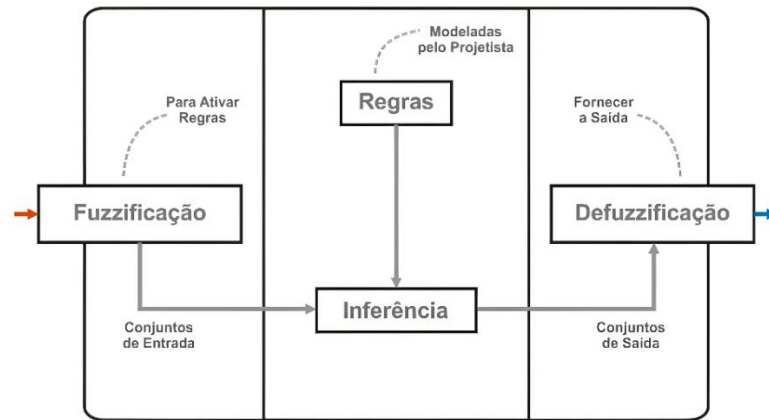
A Lógica *Fuzzy* (LF) surge como uma extensão da lógica booleana tradicional, desenvolvida para lidar com a imprecisão e a incerteza inerentes aos sistemas do mundo real e à linguagem humana. Enquanto a lógica clássica opera com valores binários (verdadeiro ou falso; 0 ou 1), a LF introduz o conceito de grau de pertinência (*membership degree*), permitindo que um elemento pertença parcialmente a múltiplos conjuntos. Essa capacidade de analisar a informação através de uma escala de verdadeiro e falso (intervalo contínuo (0, 1)) é crucial para a modelagem de fenômenos complexos (Štefek *et al.*, 2021). Na prática, a lógica *Fuzzy* é aplicada para traduzir conhecimentos humanos qualitativos em algoritmos computacionais. Inicialmente desenvolvida pelo matemático Lotfi Zadeh na década de 1960, a lógica *Fuzzy* se tornou uma ferramenta de inteligência artificial em sistemas que requerem raciocínio aproximado e flexível.

Em sistemas de controle e tomada de decisão, a lógica *Fuzzy* é empregada para desenvolver sistemas de apoio à decisão e para descrever soluções para problemas de controle. A grande vantagem reside na sua capacidade de representar o comportamento do objeto modelado através de regras *Fuzzy*, tornando os modelos altamente interpretáveis por humanos (característica de "caixa branca") (Romanov; Filippov; Yarushkina, 2025).

O emprego da lógica *Fuzzy* é particularmente benéfico em ambientes que apresentam incertezas, ruídos nos dados de entrada e não-linearidades. Em contraste com os métodos convencionais de controle, que frequentemente exigem um modelo matemático detalhado e exato do sistema, o controle por lógica *Fuzzy* é preferível por permitir a condução do sistema sem essa exigência, baseando-se em conhecimento heurístico ou empírico. Controladores baseados em LF (FLC) são notavelmente robustos e tolerantes a incertezas e ruídos, fornecendo respostas dinâmicas rápidas e precisas (Štefek *et al.*, 2021).

O componente central de um sistema *Fuzzy* é o Sistema de Inferência *Fuzzy* (FIS), ou Sistema Baseado em Regras *Fuzzy* (FRBS), que é responsável por processar as entradas e gerar as saídas de controle ou classificação. O FRBS busca o equilíbrio entre a qualidade da solução e a interpretabilidade do modelo. A arquitetura de um FIS segue um fluxo de processamento de três etapas principais mostradas na Figura 11 (Romanov; Filippov; Yarushkina, 2025).

1. Fuzzificação (*Fuzzification*): esta é a primeira etapa, no qual os dados de entrada nítidos (*crisp*) são convertidos em conjuntos *Fuzzy*. O grau de pertinência (μ) para cada valor de entrada é calculado em relação a um conjunto *Fuzzy* (Romanov; Filippov; Yarushkina, 2025). A escolha das Funções de Pertinência (MFs), como as triangulares, trapezoidais ou gaussianas, e a otimização de seus parâmetros, é um desafio crítico nesta fase para representar a incerteza de forma adequada.
2. Inferência *Fuzzy* (*Fuzzy Inference*): o coração do FIS é a base de conhecimento (*knowledge base*), que contém o conjunto de regras *Fuzzy*. As regras são tipicamente estruturadas no formato SE (condição antecedente) ENTAO (conclusão consequente). Exemplos de métodos de inferência incluem Mamdani, Sugeno e Tsukamoto. Neste processo, o grau de verdade do antecedente (agregado das entradas) é usado para calcular o grau de verdade do consequente.
3. Defuzzificação (*Defuzzification*): por fim, o resultado agregado do processo de inferência, que é um conjunto *Fuzzy*, precisa ser convertido novamente em um valor nítido de saída para ser utilizado pelo atuador do sistema de controle. O método do Centro de Gravidade (*Center of Gravity*) é um dos mais utilizados para realizar essa conversão.

Figura 11 – Sistema de Inferência *Fuzzy*

Fonte: Kridi, 2025.

Nesta monografia, se utilizam funções de pertinência trapezoidais e triangulares, devido à sua simplicidade geométrica, facilidade de ajuste (sintonia) e eficiência computacional. O método de inferência será o Mamdani e ele está caracterizado por utilizar a agregação máxima das saídas das regras ativadas e, principalmente, por empregar funções *Fuzzy* como conjuntos de saída, tornando-o um sistema intuitivo baseado em regras linguísticas. No caso da etapa de defuzzificação, se utiliza o método Centróide (ou Centro de Área/COA), o qual calcula o centro de gravidade da área sob a função de pertinência agregada (o resultado do processo de inferência) e é considerado o método mais popular e robusto para representar a saída *Fuzzy* como um valor *crisp* (nítido), e está representado pela equação 24:

$$COA = \frac{\int \mu_A(z)z \, dz}{\int \mu_A(z) \, dz} \quad (24)$$

Em que, z é a variável de saída do universo, $\mu_A(z)$ é o grau de pertinência agregado (a função *Fuzzy* combinada de saída), o numerador é o momento estático da área sob a curva e o denominador é a área total sob a curva.

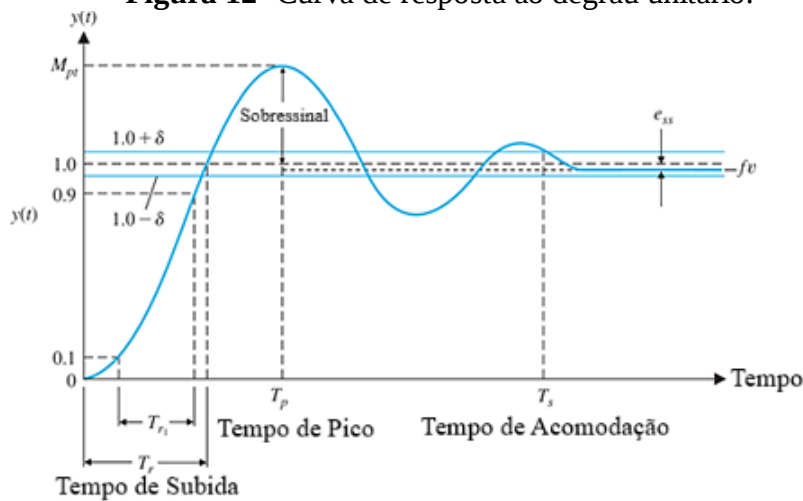
2.5 CRITÉRIOS DE DESEMPENHO

Para avaliar o desempenho de controladores, dispomos de várias métricas de qualidade para entender como o sistema responde a uma determinada entrada e nos permitem comparar diferentes tipos de controladores. Essas métricas ou índices de desempenho podem ser divididos em dois tipos: baseadas na resposta dinâmica e na integral do erro.

As métricas baseadas na resposta dinâmica (Figura 12) permitem avaliar características

como a velocidade e a precisão com que o sistema atinge seu objetivo. Algumas dessas métricas incluem o tempo de subida (t_r , tempo que o sistema leva para atingir um percentual do valor final), o tempo de acomodação (t_s , tempo para a resposta se estabilizar dentro de uma faixa de tolerância em torno do valor final), tempo de pico (t_p , o tempo que leva para a resposta atingir o máximo sobressinal), o máximo sobressinal (M_p , o quanto a resposta ultrapassa o valor final), e o erro em regime estacionário (e_{ss} , a diferença final entre o valor desejado e de regime permanente).

Figura 12- Curva de resposta ao degrau unitário.



Fonte: Dorf; Bishop, 2017.

As métricas baseadas na integral do erro utilizam critérios integrais de desempenho. Esses índices quantificam a resposta do sistema ao integrar uma função do erro ao longo do tempo, resultando em um único valor que resume sua qualidade. O objetivo do projeto de controle é minimizar esse valor para determinar a configuração ótima do controlador. Os principais índices são:

- **IAE (Integral of the Absolute Error):** está descrito pela equação 25 e soma o valor absoluto dos erros ao longo do tempo. Ele penaliza todos os erros igualmente, independentemente de seu tamanho, e é útil para obter uma resposta suave e com pouco sobressinal.

$$IAE = \int_0^{\infty} |e(t)| dt \quad (25)$$

- **ISE (Integral of the Squared Error):** está descrito pela equação 26 e soma o quadrado dos erros. Como eleva o erro ao quadrado, penaliza erros grandes de forma mais severa, o que incentiva o controlador a eliminá-los rapidamente.

$$ISE = \int_0^{\infty} e^2(t) dt \quad (26)$$

- ITAE (*Integral of the Time-weighted Absolute Error*): está descrito pela equação 27 e multiplica o erro absoluto pelo tempo antes de somá-lo. Isso dá mais peso a erros que persistem por um longo tempo, sendo a escolha ideal para garantir um erro final em regime estacionário muito pequeno.

$$ITAE = \int_0^{\infty} t|e(t)| dt \quad (27)$$

- ITSE (*Integral of the Time-weighted Squared Error*): está descrito pela equação 28 e combina as características do ISE e do ITAE, multiplicando o erro ao quadrado pelo tempo. É o índice que penaliza de forma mais acentuada erros grandes e persistentes, resultando em uma resposta que se estabiliza rapidamente e com um erro final mínimo.

$$ITSE = \int_0^{\infty} te^2(t) dt \quad (28)$$

- IAU (*Integral of Absolute Control Action*): mede o esforço de controle aplicado ao sistema. Quanto maior o IAU, maior o esforço de controle aplicado, ou seja, um IAU alto pode indicar que o controlador faz grandes variações no sinal de controle, o que pode ser indesejado em sistemas com restrições de atuação ou consumo de energia e um IAU baixo sugere que o controle é mais econômico, mas pode resultar em respostas mais lentas. Este índice é calculado a partir da equação 29, em que $u(t)$ representa o sinal de controle ao longo do tempo.

$$IAU = \int_0^{\infty} |u(t)| dt \quad (29)$$

2.6 ESTADO DA ARTE

Nesta seção, é feita uma exposição sobre a aplicação das redes neurais artificiais na academia para a modelagem e controle de sistemas dinâmicos não lineares além dos controladores PID e *Fuzzy*. São discutidas as implementações e abordagens utilizadas nesse contexto de pesquisa.

Em (Li *et al.*, 2023), é proposta uma estrutura de aprendizado por reforço para o problema de controle da pressão do vapor de saída de um gerador de vapor de passagem único (OTSG). O controlador de camada dupla utiliza o algoritmo *Proximal Policy Optimization* (PPO) na estrutura de controle do OTSG. O algoritmo PPO permite treinar redes neurais continuamente por meio da interação com o ambiente, possibilitando um controle aprimorado do OTSG. Para lidar com a dificuldade da aplicação do aprendizado por reforço em sistemas

reais, é proposta uma abordagem inovadora de pré-treinamento. O modelo *Long Short-Term Memory* (LSTM) é adotado como ambiente de treinamento para reduzir o tempo de aprendizado e aumentar a eficiência. Os resultados experimentais demonstram que essa abordagem permite o autoajuste dos parâmetros de controle em diversas condições operacionais, apresentando vantagens como baixo sobressinal, rápida estabilização e alta adaptabilidade.

Em (Bannloud; Santos; Martins, 2021), as RNAs foram utilizadas em conjunto com o controle preditivo para controlar processos de fermentação e destilação. Durante a simulação, foi avaliada a relação entre a saturação das RNAs e o desempenho do controle. Os resultados demonstraram que as RNAs foram capazes de representar adequadamente os processos, garantindo um controle robusto e sem *offset*. A saturação das RNAs não impactou negativamente o desempenho do controlador, apresentando valores tipicamente inferiores a 0,4 para funções de ativação lineares e superiores a 0,6 para sigmóides. Além disso, observou-se que o IAE tende a diminuir para redes com menor erro médio quadrático (MSE), embora isso não assegure um controle ideal, sendo necessário testar diferentes estruturas. As simulações confirmaram a eficiência das RNAs em conjunto com o controle preditivo, assegurando estabilidade no controle, com baixo *overshoot* e mínimas oscilações.

Em (Wang; Pan; Lee, 2021), é proposto um *framework* de otimização de controle para usinas termelétricas, combinando aprendizado profundo e técnicas de otimização. Um classificador Gaussiano Naïve Bayes e o método de erro de previsão extraem dados para modelagem, enquanto redes LSTM aprendem a dinâmica do sistema. Para otimizar o controle, um observador de estado estendido (ESO) melhora a rejeição de perturbações, e o algoritmo PSO ajusta os parâmetros do controlador PID. Aplicado ao controle de temperatura do vapor superaquecido (SST) em uma usina de 330 MW, o método reduziu flutuações e aprimorou o desempenho, destacando-se como uma abordagem promissora para usinas inteligentes.

Em (Hussein *et al.*, 2024), é proposto um modelo inteligente para o gerenciamento energético de geradores de vapor, focado no controle da razão ar-combustível ao longo da curva de operação e em modos transitórios. A razão ar-combustível é crucial para a eficiência da combustão, emissões reduzidas e desempenho otimizado. O sistema é simulado no *Matlab/Simulink*, avaliando diferentes condições de carga e geração. As RNAs são implementadas em vários pontos do sistema, atuando no controle do fluxo de ar, combustível e velocidade do inversor. Os testes demonstraram que a abordagem melhora a eficiência, reduz o consumo de combustível e contribui para a minimização do impacto ambiental.

Em (Kurkin *et al.*, 2023), é proposta uma arquitetura híbrida paralela de redes neurais combinando circuitos quânticos parametrizados e redes neurais clássicas para prever o fluxo de

massa de vapor em usinas termelétricas com 15 minutos de antecedência. Os resultados mostram que o modelo híbrido supera redes puramente clássicas e quânticas, reduzindo o erro médio quadrático em mais de 5,7 e 4,9 vezes, respectivamente. Além disso, apresenta erros relativos até duas vezes menores em comparação com o modelo clássico. Esses achados demonstram o potencial do aprendizado de máquina quântica para otimizar operações no setor energético e abrir novas possibilidades para aplicações industriais.

Em (Büttenbender, 2021), são investigadas diferentes estratégias de simulação para otimizar a eficiência do subsistema de moagem e geração de vapor em usinas termelétricas, utilizando redes neurais. Compararam-se abordagens com RNAs únicas para o sistema completo e modelos acoplados com RNAs individuais para cada moinho e o gerador de vapor. Os resultados indicam que a modelagem com volume de controle único apresenta menor erro. Apesar do uso de *Random Forest* e Máquinas de Vetor de Suporte (SVM) mostrar leve melhora, a acurácia ainda foi comprometida. Esses achados ressaltam a necessidade de aprimorar estratégias de acoplamento de modelos baseados em dados para representar com maior precisão a eficiência da geração de vapor em usinas termelétricas.

Em (Oliveira junior, 2009), explora-se a modelagem de componentes do circuito de vapor em centrais termelétricas a vapor, focando na caldeira de circulação natural e na conversão de energia mecânica em elétrica. A análise das respostas dinâmicas dos componentes é essencial para a avaliação de estabilidade e para o desenvolvimento do sistema de controle. O estudo aborda a utilização de modelos simples, mas eficazes, que podem ser aplicados em simulações dinâmicas do processo para testar estratégias de controle ou treinar operadores. A crescente preocupação com a escassez de água doce e as mudanças climáticas destaca a importância das termelétricas, que se beneficiam pela flexibilidade no uso de diversos combustíveis e pelo custo competitivo de geração. Os modelos discutidos são baseados em dados físicos ou em informações de plantas existentes, proporcionando uma referência importante para a criação de simuladores usados no treinamento de operadores e na implementação de malhas de controle no processo de geração de energia elétrica.

Em (Guimarães, 2023), é desenvolvido um modelo físico, matemático e computacional para o processo de geração de vapor em caldeiras aquatubulares alimentadas por biomassa, visando superar as dificuldades de testar caldeiras reais devido a questões de segurança e restrições operacionais. O modelo é implementado no ambiente *Matlab/Simulink*, utilizando a biblioteca *User-Defined Functions*, permitindo a análise detalhada das variáveis envolvidas no processo de geração de vapor. Com o apoio da empresa Aimirim, que trabalha com soluções de controle avançado e monitoramento em tempo real, o simulador é validado por meio de

comparações com dados reais de operação de caldeiras industriais. O trabalho também inclui a implementação de malhas de controle tradicionais, análise de estabilidade e resposta dinâmica, e busca aprimorar as ferramentas de controle e otimização na indústria 4.0.

Em (Jibril; Tadese; Tadese, 2020), é realizada a concepção e análise de um sistema de controle do processo de produção de cimento utilizando o *Matlab/Simulink*, com a implementação dos controladores *Nonlinear AutoRegressive Moving Average model of order 2* (NARMA-L2), modelo de referência e preditivo baseado em redes neurais. O objetivo é controlar a taxa de fluxo de vapor de saída, ajustando a válvula de fluxo de vapor, com o sistema projetado para seguir os sinais de entrada de vapor desejados, como degrau e onda senoidal. A performance dos controladores propostos é analisada por meio de simulações com e sem ruído de sensor. Os resultados mostram que, embora todos os controladores apresentem tempos de subida semelhantes, o controlador de modelo de referência tem um tempo de assentamento e sobrecarga percentual menores, tanto na presença quanto na ausência de ruído de sensor.

O estudo em (Wright; Marwala, 2008) analisou diferentes métodos de Inteligência Artificial (IA) para prever variáveis contínuas de um gerador de vapor, incluindo RNA, SVM, Sistemas de Inferência Neuro-Fuzzy Adaptativa (ANFIS), técnicas Bayesianas e comitês. Os resultados mostraram que todos os métodos foram eficazes, mas o ANFIS obteve a melhor precisão geral, além de ser mais fácil de implementar e rápido no treinamento e execução. O SVM e as redes neurais bayesianas tiveram desempenho semelhante, enquanto as redes neurais padrão tiveram mais dificuldades de ajuste. O uso de comitês melhorou ligeiramente a precisão. No geral, o ANFIS foi considerado o método mais eficiente para prever os dados do gerador de vapor.

Em (Štefek *et al.*, 2021) é proposto um processo completo de otimização de um Controlador de Lógica Fuzzy (FLC) para o controle de movimento de um Robô Móvel com Tração Diferencial (DDWMR), visando a minimização do consumo de energia e a extensão do tempo de operação. Diferente de abordagens anteriores focadas no planejamento do caminho mais curto, esta metodologia utiliza um Algoritmo Genético (GA) para otimizar o controlador, ajustando automaticamente todos os parâmetros das funções de pertinência (MFs) do FLC, buscando minimizar a aceleração do robô durante o movimento ponto a ponto. Os resultados das simulações mostraram que o controlador otimizado (GA_FLC) proporcionou uma redução de cerca de 110% no consumo de energia em comparação com o melhor dos seis controladores alternativos testados. O controlador proposto demonstrou ser superior em termos de economia de energia, resolvendo também problemas de vibração e solavancos observados nos outros controladores, mantendo um bom desempenho em diversas trajetórias.

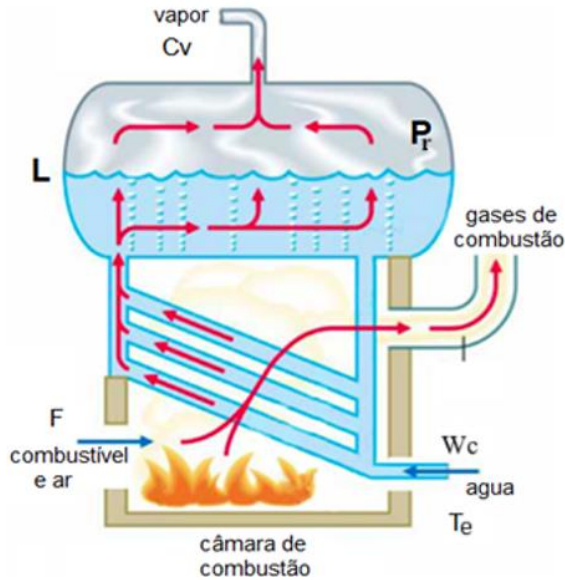
3 CONTROLE

Esta seção tem como objetivo principal apresentar o caso de estudo do gerador de vapor utilizado no trabalho e o seu modelo dinâmico. Adicionalmente, serão detalhadas as características e princípios de operação dos seguintes controladores: RNA, PID e *Fuzzy*.

3.1 CASO DE ESTUDO

Como caso de estudo neste trabalho se propõe um gerador de vapor multivariável do tipo aquatubular, representado na Figura 13, cuja dinâmica é descrita pelas equações 30, 31 e 32 (Cervantes, 2002).

Figura 13 - Gerador de Vapor Aquatubular.



Fonte: BOILER, 2020

$$\frac{dP_r}{dt} = (-0,193) \left(P_r^{\frac{1}{8}} \right) S + (0,014524)F - (0,000736)W_c + (0,00121)L + (0,000176)T_e \quad (30)$$

$$\frac{dS}{dt} = (0,1) \left(c_v \left(P_r^{\frac{1}{2}} \right) \right) - (0,785716)S \quad (31)$$

$$\frac{dL}{dt} = (0,00863)(W_c) + (0,002)F + (0,463)c_v - (10^{-6})P_r^2 - (0,00914)L - 8,2(10^{-5})L^2 - (0,7328)S \quad (32)$$

em que, S é o fluxo de vapor, uma variável que não se controla diretamente. As variáveis controladas (saídas) são a pressão no interior do gerador (P_r) e o nível de água no gerador (L) e as variáveis manipuladas (entradas) são o fluxo de combustível (F) e fluxo de entrada de água (W_c). Como distúrbios têm a demanda de vapor (C_v) e a temperatura de entrada da água (T_e). A Tabela 1 apresenta os parâmetros nominais do gerador de vapor.

Tabela 1 - Parâmetros do Gerador de Vapor Aquatubular

Variável	Valores
Pr	175,8 (Kg/cm ²)
L	64 cm
F	20 a 45 Kg/s
W _c	100 a 320 Kg/s
C _v	0,8
T _e	310 K

Fonte: Cervantes, 2002

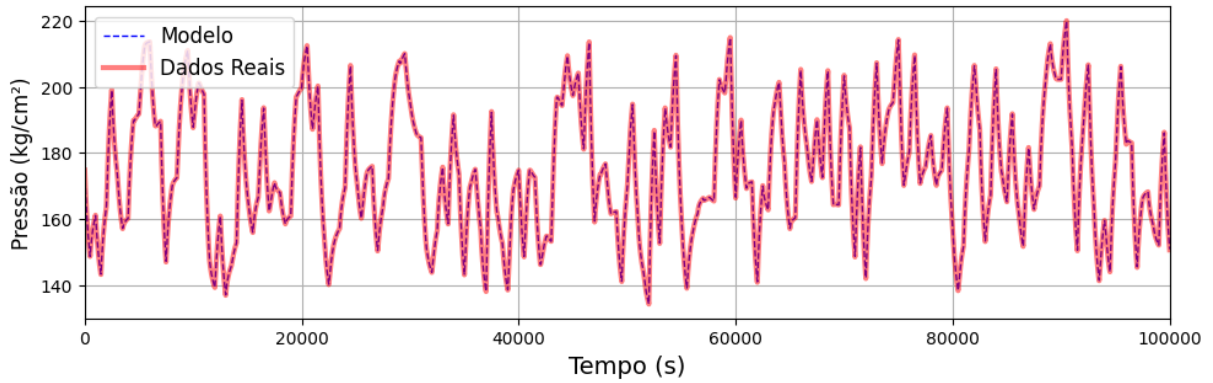
A partir das equações diferenciais 30, 31 e 32 obtém-se um modelo matemático em espaço de estados do gerador de vapor desenvolvido em Python, o qual é obtido por meio da função *stateSpaceModel*, que representa matematicamente a dinâmica do sistema em termos de suas variáveis de estado: pressão, fluxo de vapor e nível de água. Essa função recebe como entrada o vetor de estado atual e o tempo, e retorna as derivadas de cada variável com base nas equações diferenciais previamente definidas e transformadas em funções numéricas com *lambdify* (da biblioteca *SymPy*). Dentro do ciclo de cálculo, os valores das entradas de controle (fluxo de água e de combustível) são atualizados continuamente por interpolação linear via *numpy.interp* a partir de uma lista de valores desejados (dados experimentais). Essa abordagem permite aplicar o método de integração numérica *odeint*, da biblioteca *SciPy*, que utiliza a função *stateSpaceModel* para calcular a trajetória temporal das variáveis de estado com base nas condições iniciais e nas entradas fornecidas. A equação 33 mostra o modelo em espaço de estados obtido.

$$\begin{bmatrix} \frac{dP_r}{dt} \\ \frac{dS}{dt} \\ \frac{dL}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -0,193P_r^{1/8} & 0,00121 \\ 0 & -0,785716 & 0 \\ -6(10^{-6})P_r & -0,7328 & -0,00914 - 8,2(10^{-5})L \end{bmatrix} \begin{bmatrix} P_r \\ S \\ L \end{bmatrix} + \begin{bmatrix} 0,014524 & -0,000736 & 0 & 0,000176 \\ 0 & 0 & 0,1P_r^{1/2} & 0 \\ 0,002 & 0,00863 & 0,463 & 0 \end{bmatrix} \begin{bmatrix} F \\ W_c \\ C_v \\ T_e \end{bmatrix} \quad (33)$$

Nas Figuras 14 e 15 se mostram os gráficos da validação do modelo em espaço de estados e os dados experimentais do gerador de vapor (processo), no qual os MSEs para pressão e nível da água são: 0,06 e 0,1, respectivamente. O conjunto de dados experimentais é

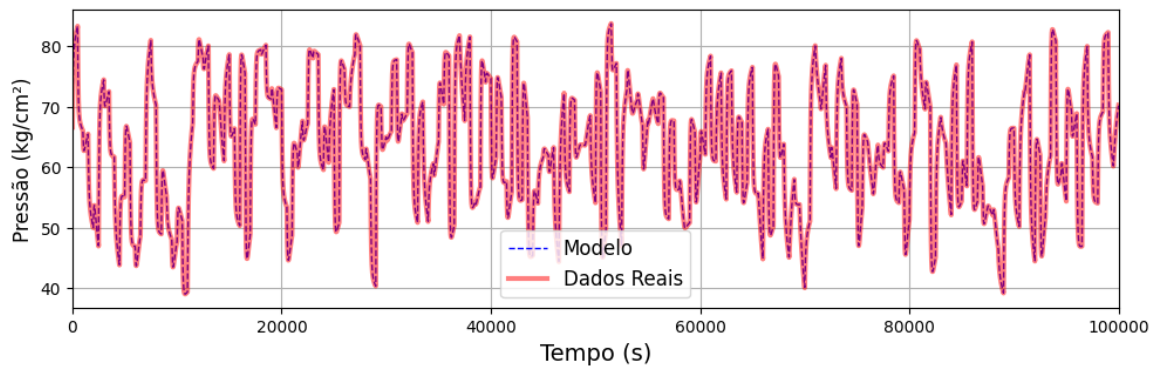
estruturado em quatro séries temporais distintas, tendo um total de 10 mil dados ao todo, representando as duas variáveis de entrada e as duas variáveis de saída do processo (Cervantes, 2002).

Figura 14 - Validação entre modelo e o processo para a saída de pressão.



Fonte: elaborado pelo próprio autor.

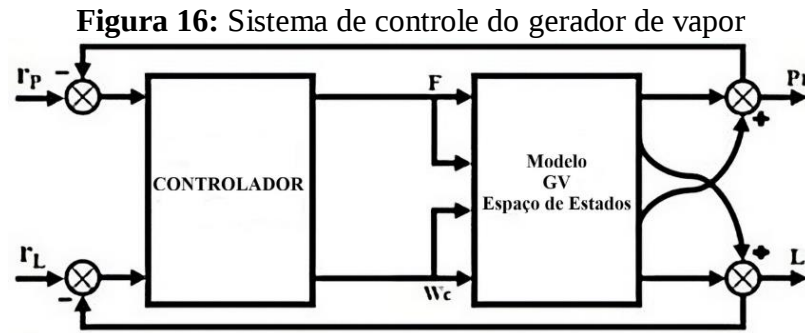
Figura 15 - Validação entre modelo e o processo para a saída de nível de água.



Fonte: elaborado pelo próprio autor.

3.2 CONTROLE DO GERADOR DE VAPOR

O diagrama do sistema de controle do gerador de vapor é apresentado na Figura 16, no qual é utilizado o modelo matemático do sistema em espaço de estados. Nesta seção se descreve a implementação dos controladores por RNA, PSO PID, AG PID e *Fuzzy* no gerador de vapor com base nesse diagrama mediante programação em *Python*.



Fonte: elaborado pelo próprio autor.

3.2.1. Controle com RNA

O controle por RNA é desenvolvido utilizando redes do tipo MLP com o método de aprendizado por reforço, empregando o algoritmo SAC. Para isso, utilizando a linguagem de programação Python foi criado o ambiente CaldeiraEnv (CE), uma simulação do comportamento dinâmico de uma caldeira industrial, implementado com a biblioteca Gymnasium. Esse ambiente permite que um agente de aprendizado por reforço interaja com o sistema, ajustando variáveis operacionais para otimizar o desempenho da caldeira. A seguir a configuração do ambiente é detalhada.

Configuração do Ambiente: o ambiente CaldeiraEnv modela o comportamento da caldeira com base nas equações 30, 31 e 32. O sistema recebe como entradas o fluxo de combustível injetado no queimador e o fluxo de água, enquanto a saída consiste nos valores observáveis do estado da caldeira, incluindo erros de controle, e as variações de pressão e nível de água e todas as variáveis são normalizadas no intervalo que vai de -1 a 1. A configuração do ambiente define:

- **Espaço de Ações:** intervalo contínuo de valores normalizados para entrada de combustível e entrada de água, permitindo ajustes dinâmicos pelo agente. O fluxo de combustível tem um limite inferior de 20 kg/s e superior de 45 kg/s. Já o fluxo de água tem um limite inferior de 100 kg/s e superior de 320 kg/s.
- **Espaço de Observações:** vetor de estados contendo informações sobre os erros de controle e variações da pressão e nível de água. As taxas de variações da pressão e do nível de água possuem um limite inferior de $-0,316 \text{ kg/cm}^2 \cdot \text{s}^{-1}$ e $-0,86 \text{ cm} \cdot \text{s}^{-1}$, respectivamente. Já os limites superiores são de $0,23 \text{ kg/cm}^2 \cdot \text{s}^{-1}$ e $1,17 \text{ cm} \cdot \text{s}^{-1}$, respectivamente.
- **Parâmetros Físicos:** definição de constantes como limites de pressão e nível de água,

coeficientes de troca térmica e características da válvula de entrada. A pressão tem um limite inferior de 102 kg/cm² e superior de 202 kg/cm². Já o nível de água tem um limite inferior de 40 cm e superior de 80 cm.

Dinâmica do Treinamento: durante a interação, o agente recebe uma observação do estado da caldeira e decide quais ações tomar. As ações são aplicadas ao sistema e, com base em equações diferenciais resolvidas numericamente, a nova condição da caldeira é calculada.

A função de recompensa avalia o desempenho do agente considerando o erro em relação aos alvos de pressão e nível de água, além de penalizações para variações abruptas nas ações de controle. A função de recompensa pode ser expressa matematicamente como a soma de quatro componentes: recompensa de modelagem (*shaping*), recompensa direta, penalidade de variação de ação e recompensa de objetivo. A recompensa total (R_{total}) no instante de tempo t é dada por a Equação 34:

$$R_{total,t} = R_{shaping,t} + R_{direct,t} + R_{\Delta action,t} + R_{goal,t} \quad (34)$$

Recompensa de modelagem (*Reward Shaping*): a função de modelagem (S_t) no instante de tempo t (Equação 35) é uma combinação ponderada das distâncias euclidianas dos erros e das variações de estado. A recompensa de modelagem ($R_{shaping,t}$) é a diferença entre a função de modelagem no instante de tempo t e no instante de tempo anterior $t-1$. Para o primeiro passo de tempo ($t=0$), a recompensa é zero.

$$S_t = -\omega_{err} \sqrt{E_p^2 + E_L^2} - \omega_{var} \sqrt{\Delta P^2 + \Delta L^2} \quad (35)$$

em que:

$\omega_{err} \rightarrow$ Coeficiente de peso para o termo de erro;

$\omega_{var} \rightarrow$ Coeficiente de peso para o termo de variação;

$E_p^2 \rightarrow$ Erro da pressão;

$E_L^2 \rightarrow$ Erro do nível;

$\Delta P \rightarrow$ Variação da pressão;

$\Delta L \rightarrow$ Variação do nível.

Recompensa direta (*Reward Direct*) ($R_{direct,t}$): é uma penalidade aplicada se o agente estabilizar longe do alvo e se calcula através da Equação 36. Essa recompensa só ocorre se o agente tiver uma variação na pressão ou no nível menor que a de um limite estabelecido fora do alvo.

$$R_{\text{direct},t} = -\omega_{\text{dir}} \sqrt{E_P^2 + E_L^2} \quad (36)$$

sendo, ω_{dir} um fator de peso.

Penalidade de variação da ação (*Reward Action Variation*) ($R_{\Delta\text{action},t}$): esta é uma penalidade que dissuade o agente de fazer grandes variações nas suas ações entre os passos de tempo e se calcula através da Equação 37.

$$R_{\Delta\text{action},t} = -\omega_{fc} |\Delta F| - \omega_{fa} |\Delta Wc| \quad (37)$$

em que, ω_{fc} e ω_{fa} são pesos associados a variação das ações de controle.

Termo de Recompensa de Objetivo (*Reward Goal*) ($R_{\text{goal},t}$): é uma recompensa positiva concedida quando o agente atinge o alvo, calculada pela Equação 38.

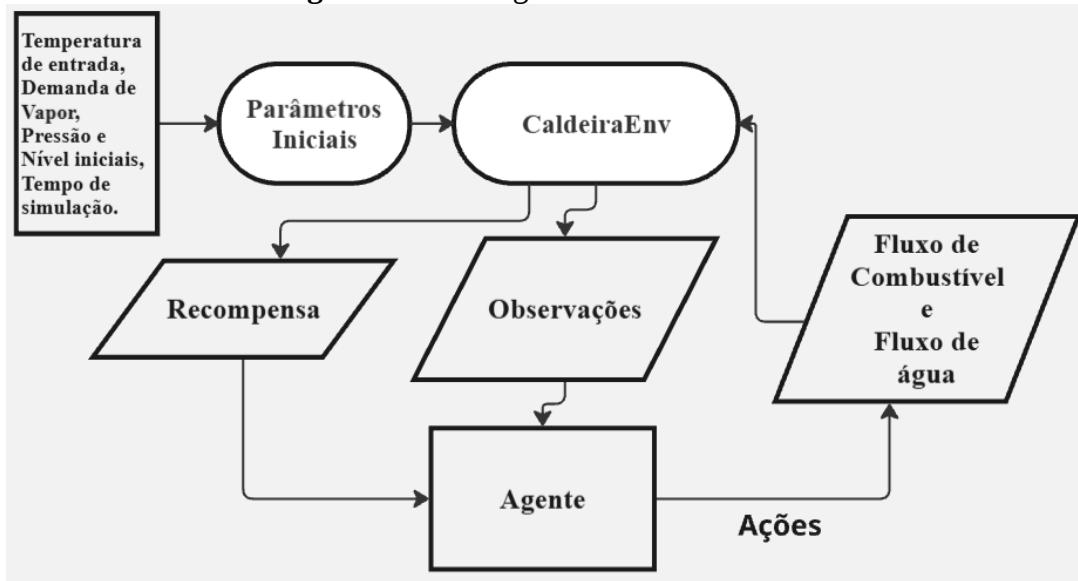
$$R_{\text{goal},t} = \begin{cases} R & \text{se } E_P \text{ e } E_L < 0.1 \\ 0 & \text{caso contrário} \end{cases} \quad (38)$$

em que $R \in \mathbb{R}$.

Se a caldeira operar dentro dos parâmetros desejados, definidos nas funções de recompensa, o agente recebe uma recompensa positiva, incentivando ajustes que levem à estabilidade do sistema.

Critérios de parada e reinicialização: o episódio de treinamento termina quando o agente atinge um número máximo de iterações (7991 iterações para um passo de 1s) ou quando a caldeira ultrapassa limites críticos de segurança. Em seguida, o ambiente é resetado, reiniciando a simulação com novos valores-alvo de pressão e nível de água para garantir generalização no aprendizado. O fluxograma do ambiente CaldeiraEnv se mostra na Figura 17.

Figura 17 - Fluxograma do ambiente CE.



Fonte: elaborado pelo próprio autor.

Durante a simulação, a caldeira tenta atingir e manter níveis alvo de pressão e nível de água, que podem mudar em determinados momentos do episódio. O ambiente calcula a recompensa com base na proximidade dos valores atuais ao alvo e se os ajustes nas entradas (combustível e água) estão conduzindo o sistema para a direção correta. A observação retornada pelo ambiente inclui informações como o erro da pressão e do nível de água, e os valores da variação da pressão e do nível de água.

O episódio termina quando o número máximo de passos é alcançado, ou seja, em 7991. Além disso, informações adicionais sobre as ações tomadas, como a quantidade de combustível e água injetados, são fornecidas para facilitar a análise do desempenho do agente. A precisão da simulação depende do passo de tempo utilizado na integração das equações diferenciais, nesse caso, foi utilizado um passo de tempo de 1s. Passos menores proporcionam uma maior precisão na modelagem do comportamento da caldeira, mas aumentam a demanda computacional, tornando a simulação mais lenta. Portanto, há um compromisso entre a precisão da simulação e a eficiência computacional: passos de tempo menores resultam em um modelo mais detalhado, enquanto passos maiores tornam a execução do código mais rápida.

Foram também realizados testes comparativos entre os algoritmos PPO e SAC. Em termos de eficiência computacional, o PPO demonstrou uma velocidade de processamento por episódio superior ao SAC; contudo, o tempo total necessário para a convergência do treinamento foi muito mais extenso devido à sua acentuada instabilidade. Essa característica é inerente à natureza *on-policy* do PPO que, por não utilizar um *Replay Buffer* para reutilizar experiências passadas em *batches*, apresenta uma menor eficiência de amostragem e maior

oscilação na aprendizagem quando comparado ao SAC.

Em uma etapa experimental inicial, os algoritmos foram avaliados quanto à composição do vetor de observação e à lógica de recompensa. Inicialmente, o espaço de observação das redes neurais incluía o erro, sua integral, o *setpoint* e o valor atual das variáveis. A função de recompensa utilizada nessa fase baseava-se em uma lógica por partes com limiares fixos: o agente recebia uma recompensa definida por uma parábola invertida enquanto o erro permanecia dentro de uma zona de tolerância específica, sendo penalizado linearmente caso o erro excedesse esse limite. Contudo, os testes demonstraram que a nova implementação, baseada na técnica de *Reward Shaping* e penalidades por variação de ação, apresentou resultados superiores.

3.2.2. Controle PID

Na Tabela 2 mostra-se a sintonia dos controladores PID, a qual é realizada utilizando dois algoritmos de otimização: PSO e GA.

Tabela 2 - Ganhos dos controladores PID

Tipo	Pressão			Nível		
	Kp	Ki	Kd	Kp	Ki	Kd
PSO PID	54,908312	0	67,859728	142,8399649	0,1642161	44,9966923
AG PID	70,527774	0,00012	22,748133	101,550131	6,3855784	44,4060812

Fonte: elaborado pelo próprio autor.

Os parâmetros para o algoritmo de otimização PSO estão dispostos na Tabela 3, os quais estão descritos em (Roncaglio, 2020), e foram escolhidos de forma empírica através das revisões bibliográficas onde esses valores são comuns. A dimensão das partículas é de 6, que corresponde aos 6 ganhos do PID e a função de custo utilizada tanto para o PSO quanto para o AG foi o ISE (função de custo que obteve os melhores resultados).

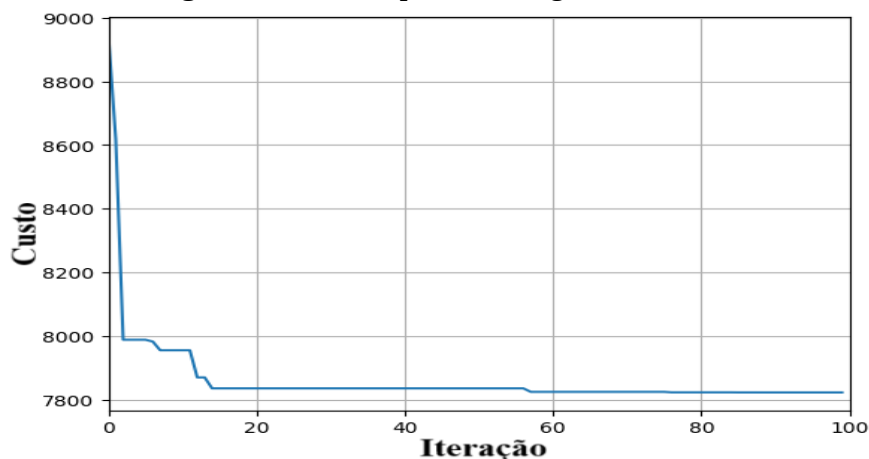
Tabela 3 - Parâmetros do Algoritmo PSO.

Parâmetros	C ₁	C ₂	max	min	Iterações	Partículas
Valores	2,05	2,05	0,9	0,1	100	100

Fonte: Roncaglio, 2020

A Figura 18 ilustra o desempenho do algoritmo PSO ao longo das iterações, mostrando a evolução da função de custo. Observa-se uma rápida redução do custo nas primeiras 20 iterações, indicando que o algoritmo encontrou rapidamente uma solução promissora. Posteriormente, a curva estabiliza, mostrando que o algoritmo converge para um valor mínimo de custo (aproximadamente 7800) e não há mais melhorias significativas após 60 iterações. Isso demonstra que o PSO foi eficaz na busca por uma solução ótima para o problema de otimização.

Figura 18 - Desempenho do algoritmo PSO.



Fonte: elaborado pelo próprio autor.

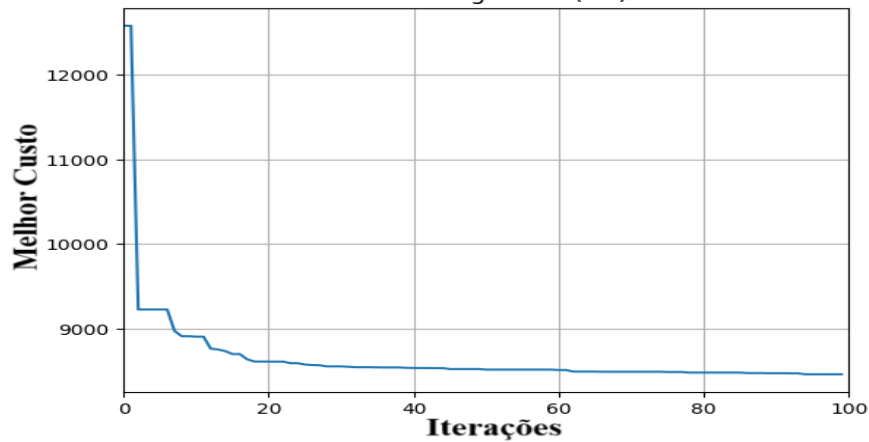
Os parâmetros para o AG mostram-se na Tabela 4, e foram escolhidos de forma empírica através das revisões bibliográficas onde esses valores são comuns. A Figura 19 mostra o desempenho do AG ao longo das 100 iterações. Cada pessoa dentro da população tem 6 genes, semelhante às dimensões do algoritmo PSO.

Tabela 4 - Parâmetros do AG.

Parâmetros	β	γ	δ	σ	Iterações	População
Valores	1	0,1	0,1	0,01	100	100

Fonte: elaborado pelo próprio autor.

Figura 19 - Desempenho do AG
Genetic Algorithm (GA)



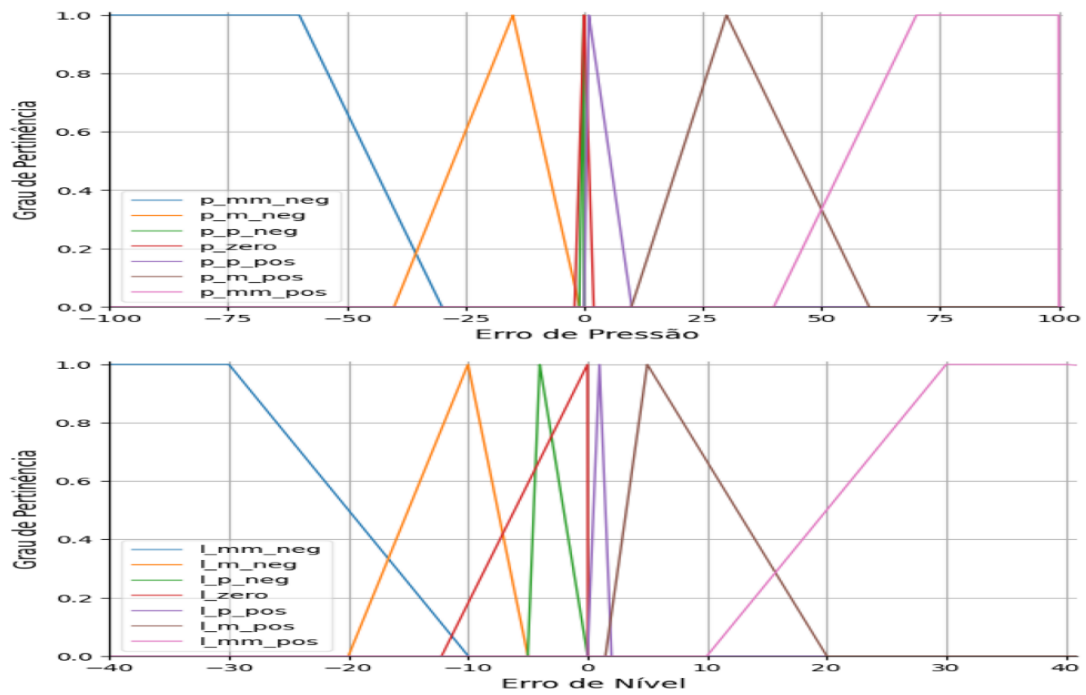
Fonte: elaborado pelo próprio autor.

Analisando a Figura 19, observar-se a trajetória de otimização do AG. A curva de custo decresce rapidamente, mas a estabilização ocorre mais lentamente em comparação ao PSO. O AG converge para um custo final acima de 8400, um valor superior ao obtido com o PSO (aproximadamente 7800), o que sugere um desempenho de otimização ligeiramente inferior para este problema específico.

3.2.3. Controle *Fuzzy*

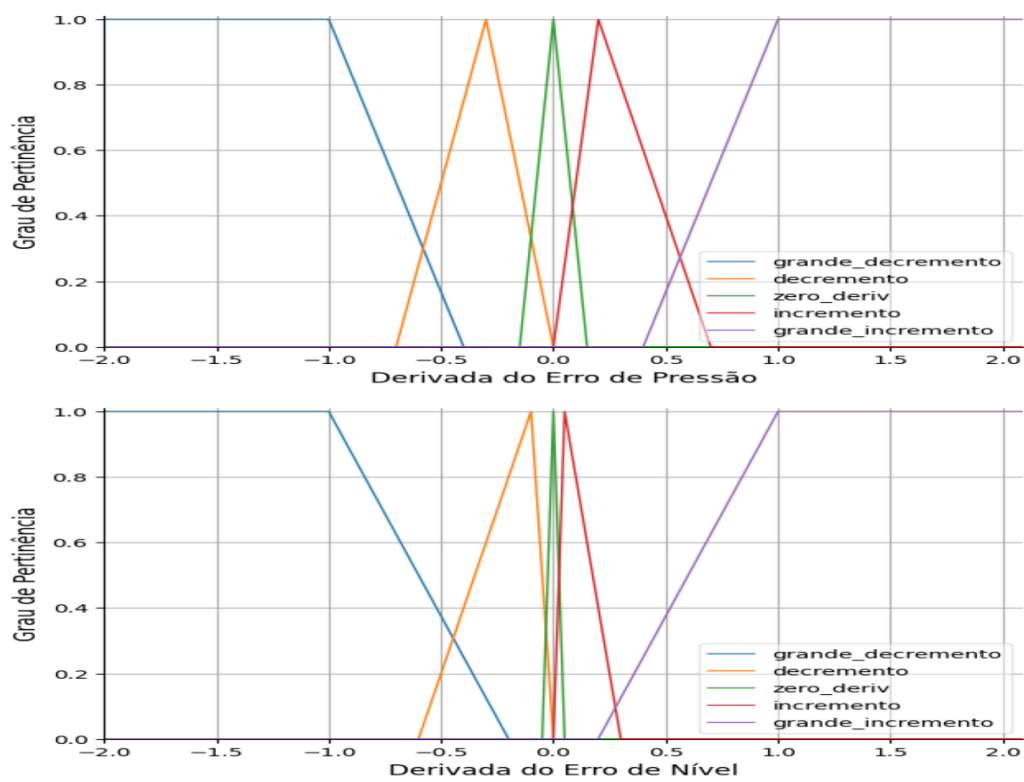
A implementação do controlador *Fuzzy* foi desenvolvida utilizando o método de inferência de Mandami cujas entradas (antecedentes) do controlador são o erro (e) e a derivada do erro (de). As saídas (consequente) do controlador, chamadas de ação de controle (u) são o fluxo de combustível e o fluxo de água. Para a fuzzificação dessas entradas, foram empregadas funções de pertinência trapezoidais e triangulares, as quais são mostradas nas Figuras 20 e 21.

Figura 20 - Função de pertinência da entrada “erro” – pressão e nível



Fonte: elaborado pelo próprio autor.

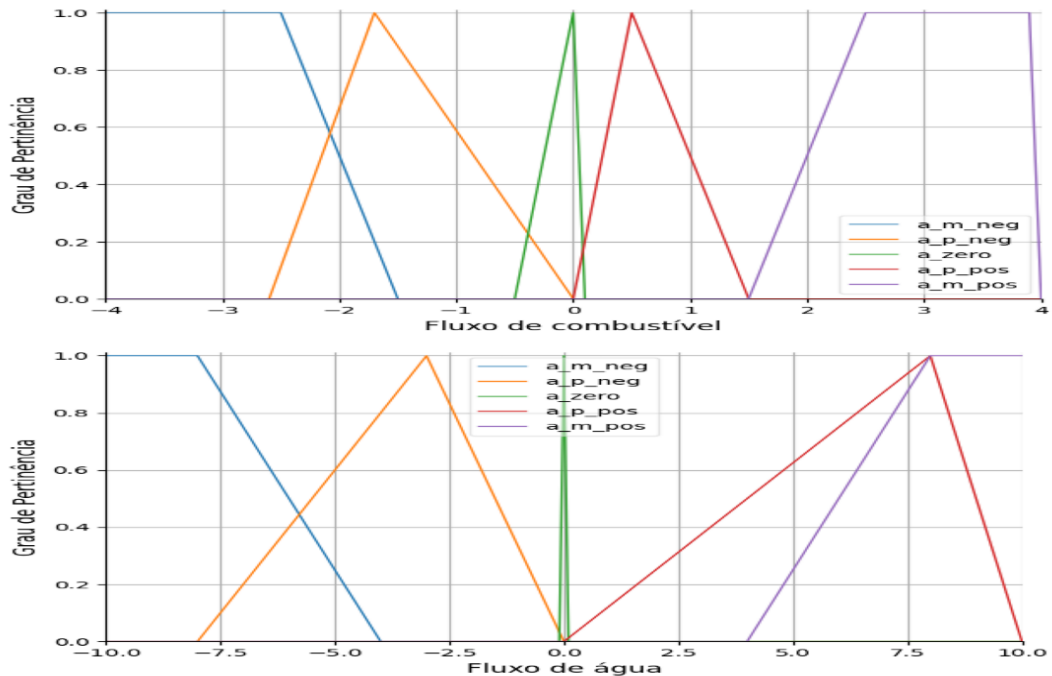
Figura 21 - Função de pertinência da entrada “derivada do erro” – pressão e nível



Fonte: elaborado pelo próprio autor.

As funções de pertinência das ações de controle fluxo de combustível e fluxo de água são do tipo triangulares e trapezoidais e mostram-se na Figura 22.

Figura 22 - Função de pertinência da ação de controle “fluxo de combustível e fluxo de água”.



Fonte: elaborado pelo próprio autor.

As funções de pertinência associadas a cada rótulo, define o grau em que um determinado valor numérico pertence a um conceito linguístico, sendo a base para a lógica *Fuzzy*. Os termos como *p_mm_neg* (Pressão Muito, Muito Negativa) e *grande_incremento* são os rótulos linguísticos que definem os conjuntos *Fuzzy*. Eles transformam valores numéricos contínuos (entradas) em conceitos que podem ser processados por regras, como, por exemplo, o rótulo *am_pos* (Tabela 5), que significa Ação Muito Positiva, indicando um comando de correção para aumentar a variável controlada. A criação das regras de inferência está baseada nos estudos do funcionamento do gerador de vapor. De acordo com o exposto na seção 2.4 a lista de regras é definida da forma: IF-THEN (SE-ENTÃO). As Tabelas 5 e 6 representam as matrizes de regras de inferência para o controle de pressão e nível, respectivamente.

Quadro 1 - Matriz de regras para o controle de pressão.

		Erro						
		pmm_neg	pm_neg	pp_neg	pzero	pp_pos	pm_pos	pmm_pos
Derivada do Erro (deP)	grande decremento	am_pos	am_pos	am_pos	ap_pos	am_neg	am_neg	am_neg
	decremento	am_pos	am_pos	am_pos	a_zero	am_neg	am_neg	am_neg
	zero derivada	am_pos	am_pos	ap_pos	a_zero	am_neg	am_neg	am_neg
	incremento	am_pos	am_pos	ap_neg	a_zero	am_neg	am_neg	am_neg
	grande incremento	am_pos	am_pos	azero	ap_neg	am_neg	am_neg	am_neg

Fonte: elaborado pelo próprio autor.

Quadro 2 - Matriz de regras para o controle de nível.

		Erro						
		lmm_neg	lm_neg	lp_neg	l_zero	lp_pos	lm_pos	lmm_pos
Derivada do Erro (deL)	grande decremento	am_pos	am_pos	am_pos	ap_pos	am_pos	am_neg	am_neg
	decremento	am_pos	am_pos	ap_pos	a_zero	am_pos	am_neg	am_neg
	zero derivada	am_pos	am_pos	ap_pos	a_zero	ap_neg	am_neg	am_neg
	incremento	am_pos	am_pos	am_neg	a_zero	am_neg	am_neg	am_neg
	grande incremento	am_pos	am_pos	am_neg	ap_neg	am_neg	am_neg	am_neg

Fonte: elaborado pelo próprio autor.

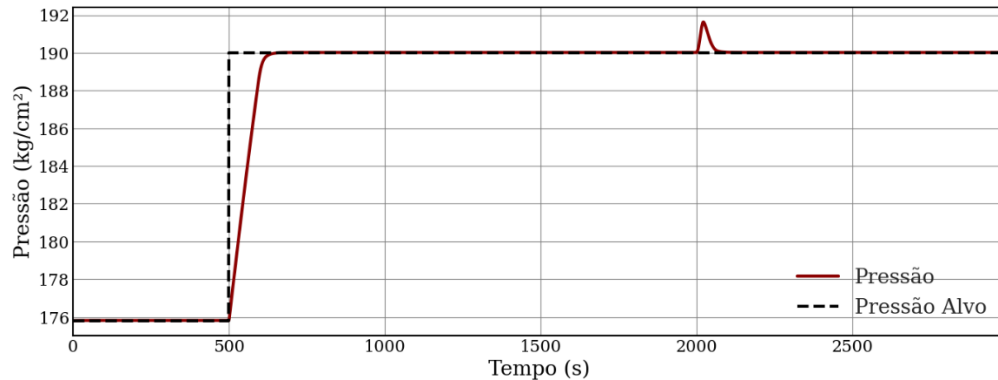
4 RESULTADOS

O capítulo apresenta os resultados de controle para a pressão e o nível do gerador de vapor, obtidos com os controladores RNA, PID e *Fuzzy*. Além, mostra-se uma análise comparativa do desempenho dos controladores utilizados.

4.1 RESULTADO DO CONTROLE COM RNA

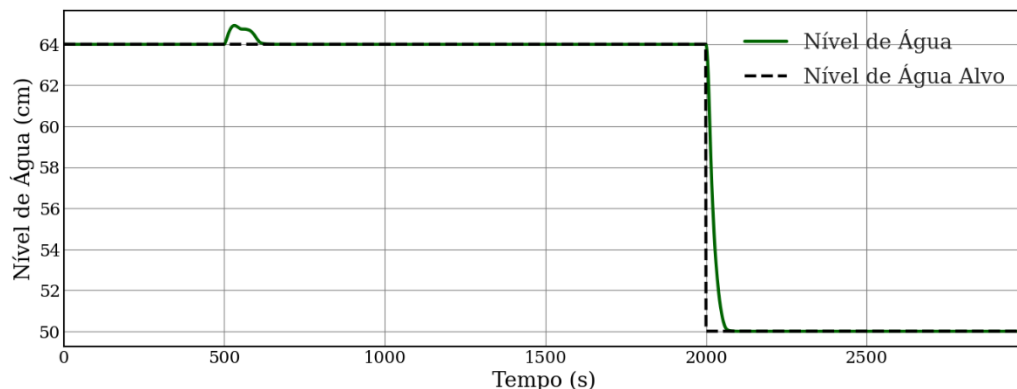
Nesta seção mostram-se os resultados das simulações do controle por RNA descrito na seção 3.2.1. Nas Figuras 23 e 24, apresenta-se a resposta da pressão de vapor e do nível de água ao longo do tempo, considerando duas mudanças na referência. A primeira mudança ocorre aos 500s, quando a referência de pressão passa de 175,8 para 190. A segunda mudança acontece aos 2000s, quando a referência para o nível de água é reduzida de 64 para 50. Nessas Figuras, observa-se o que o controlador neural consegue seguir as variações das referências em ambas saídas, assim como evidencia-se o nível de interação presentes no gerador de vapor multivariável, em que uma mudança de referência na pressão tem impacto na saída de nível e vice-versa.

Figura 23 - Saída de pressão de vapor para mudança na referência com controle RNA.



Fonte: elaborado pelo próprio autor.

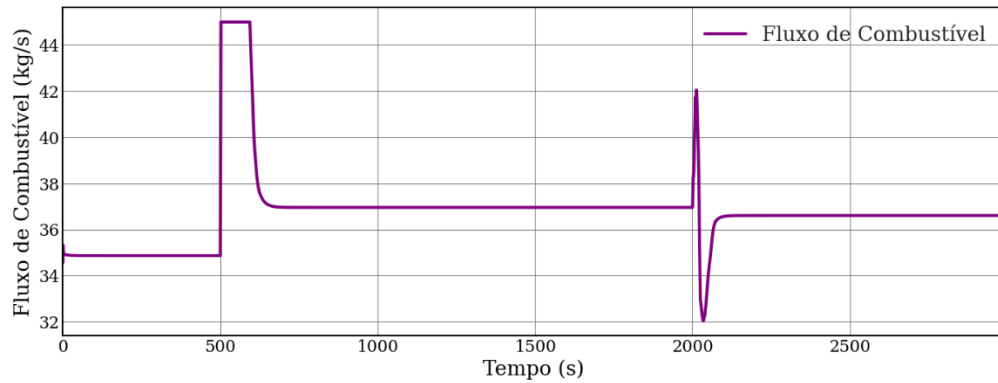
Figura 24 - Saída de nível de água para mudança na referência com controle RNA.



Fonte: elaborado pelo próprio autor.

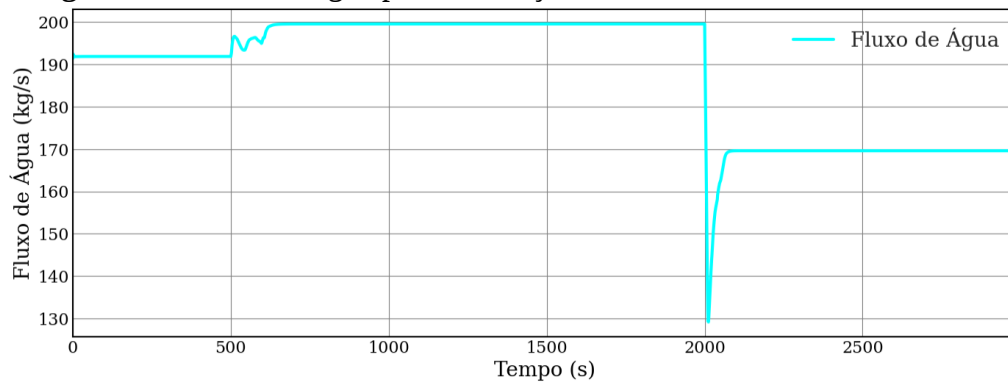
As Figuras 25 e 26 mostram o fluxo de combustível e o fluxo de água para as mudanças na referência com controle RNA.

Figura 25 - Fluxo de combustível para mudança na referência com controle RNA.



Fonte: elaborado pelo próprio autor.

Figura 26 - Fluxo de água para mudança na referência com controle RNA.

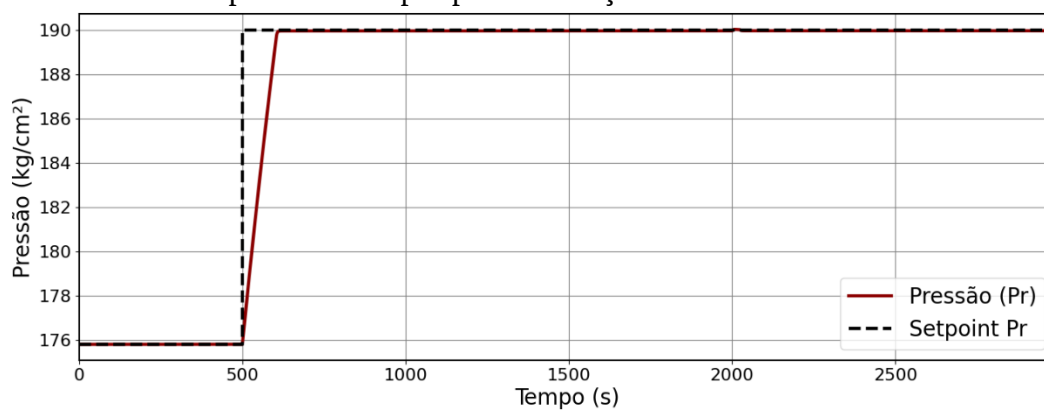


Fonte: elaborado pelo próprio autor.

4.2 RESULTADOS DO CONTROLE PID

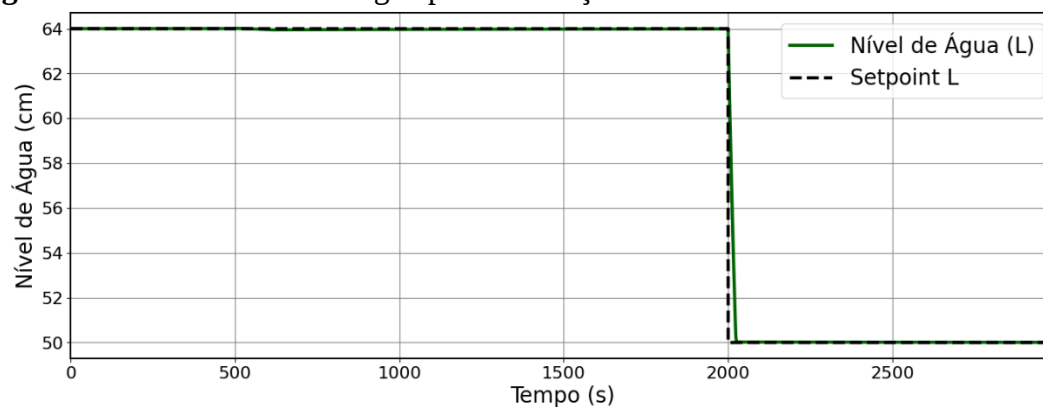
Nesta seção mostram-se os resultados das simulações do controle PID descrito na seção 3.2.2. Primeiramente apresenta-se nas Figuras 27 a 30 os resultados do controlador PID otimizado pelo algoritmo PSO. As Figuras 27 e 28 apresentam as respostas da pressão de vapor e do nível de água ao longo do tempo, considerando as mesmas duas mudanças nas referências utilizadas para o controlador RNA. Já nas Figuras 29 e 30, são exibidas as ações de controle, que são os fluxos de combustível e de água. Nessas Figuras, observa-se o seguimento nas referências em ambas saídas e uma diminuição do nível de interação presentes no gerador de vapor multivariável.

Figura 27 - Saída de pressão de vapor para mudança na referência com controle PSO PID.



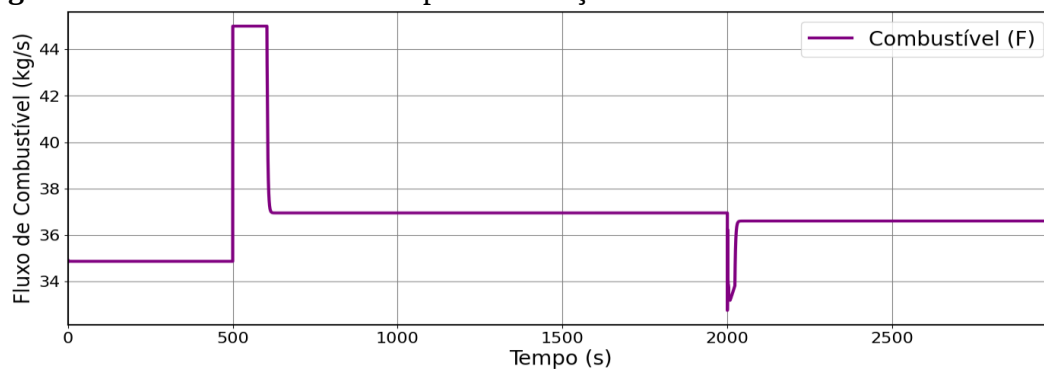
Fonte: elaborado pelo próprio autor.

Figura 28 - Saída de nível de água para mudança na referência com controle PSO PID.



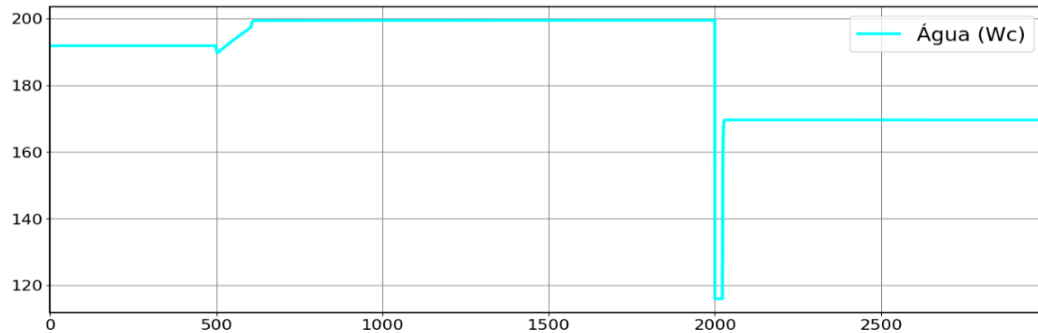
Fonte: elaborado pelo próprio autor.

Figura 29 - Fluxo de combustível para mudança na referência com controle PSO PID.



Fonte: elaborado pelo próprio autor.

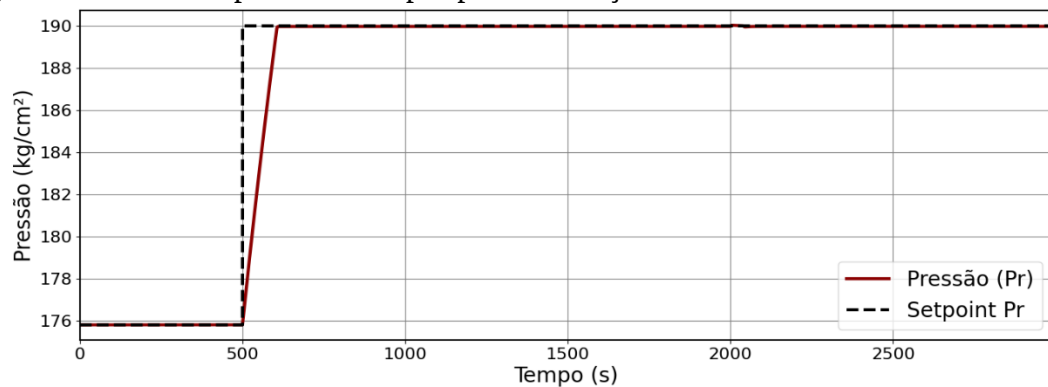
Figura 30 - Fluxo de água para mudança na referência com controle PSO PID.



Fonte: elaborado pelo próprio autor.

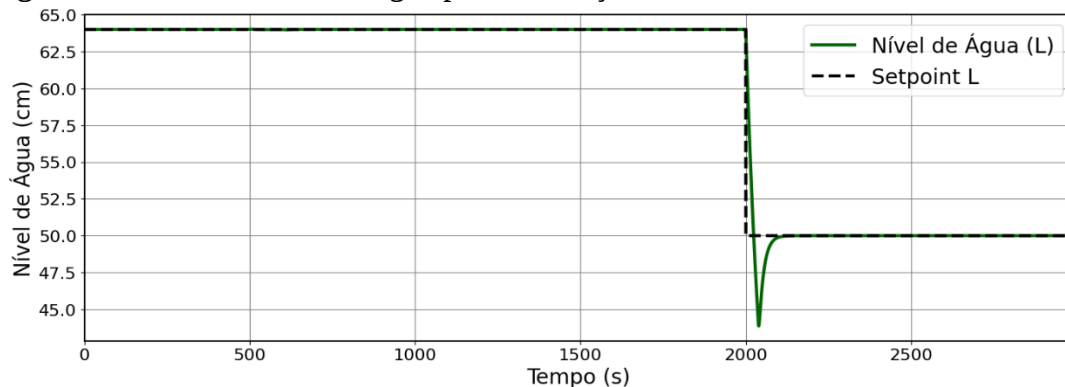
As Figuras 31 a 34 apresentam os resultados do controlador PID otimizado por AG. De forma análoga aos gráficos anteriores, as Figuras 31 e 32 mostram as respostas da pressão de vapor e do nível de água, enquanto as Figuras 33 e 34 exibem as respectivas ações de controle (fluxos de combustível e de água), observando-se seguimento nas referências em ambas saídas e uma diminuição do nível de interação.

Figura 31 - Saída de pressão de vapor para mudança na referência com controle AG PID.



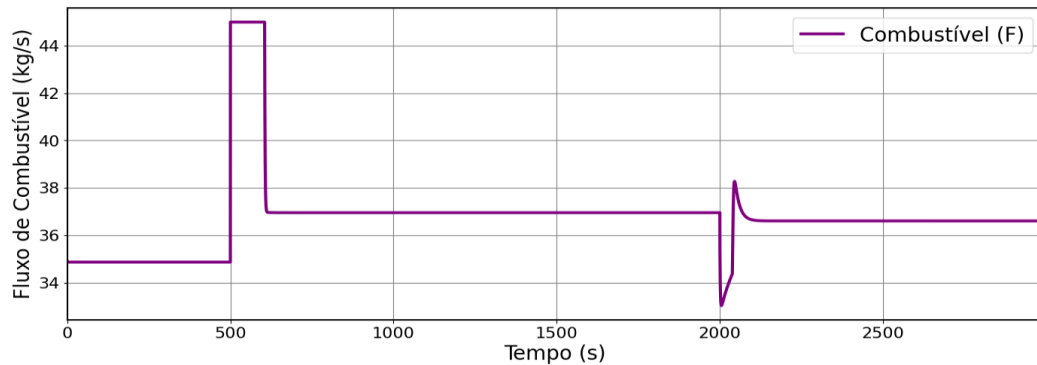
Fonte: elaborado pelo próprio autor.

Figura 32 - Saída de nível de água para mudança na referência com controle AG PID.



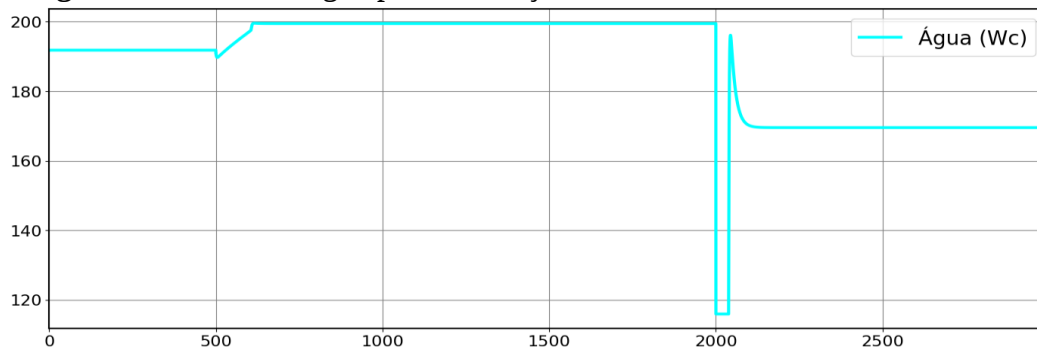
Fonte: elaborado pelo próprio autor.

Figura 33 - Fluxo de combustível para mudança na referência com controle AG PID.



Fonte: elaborado pelo próprio autor.

Figura 34 - Fluxo de água para mudança na referência com controle AG PID.

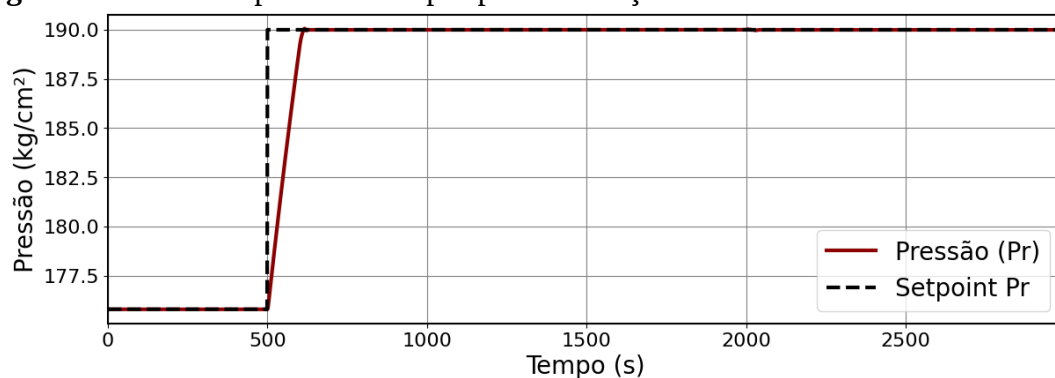


Fonte: elaborado pelo próprio autor.

4.3 RESULTADOS DO CONTROLADOR *FUZZY*

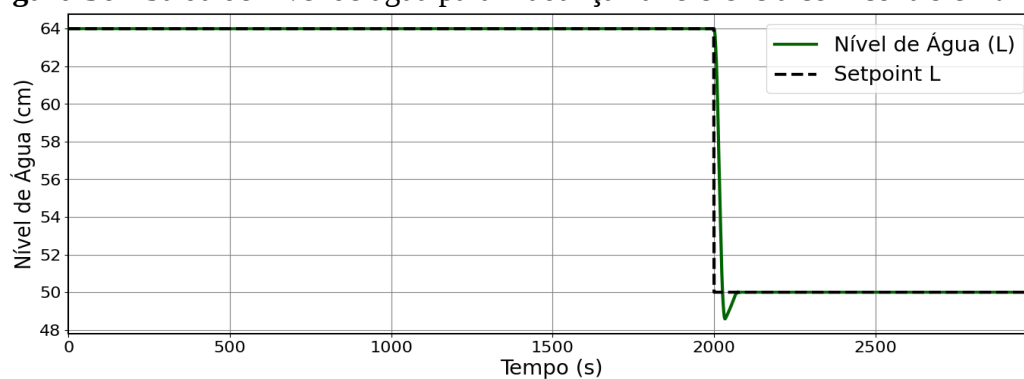
Nesta seção mostram-se os resultados das simulações do controle *Fuzzy* descrito na seção 3.2.3. As Figuras 35 e 36 apresentam as respostas da pressão de vapor e do nível de água considerando as mesmas duas mudanças nas referências utilizadas para os controladores anteriores, observando-se seguimento nas referências em ambas saídas e um baixo nível de interação entre as variáveis. Já nas Figuras 37 e 38, são exibidas as ações de controle, que são os fluxos de combustível e de água.

Figura 35 - Saída de pressão de vapor para mudança na referência com controle *Fuzzy*.



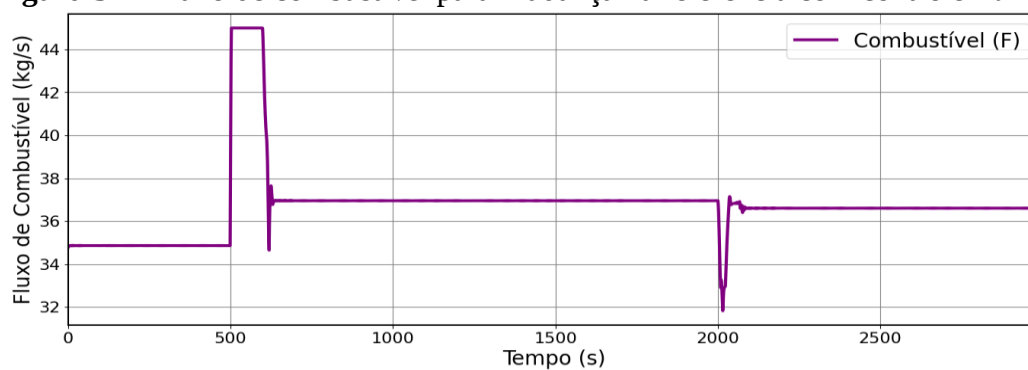
Fonte: elaborado pelo próprio autor.

Figura 36 - Saída de nível de água para mudança na referência com controle *Fuzzy*.



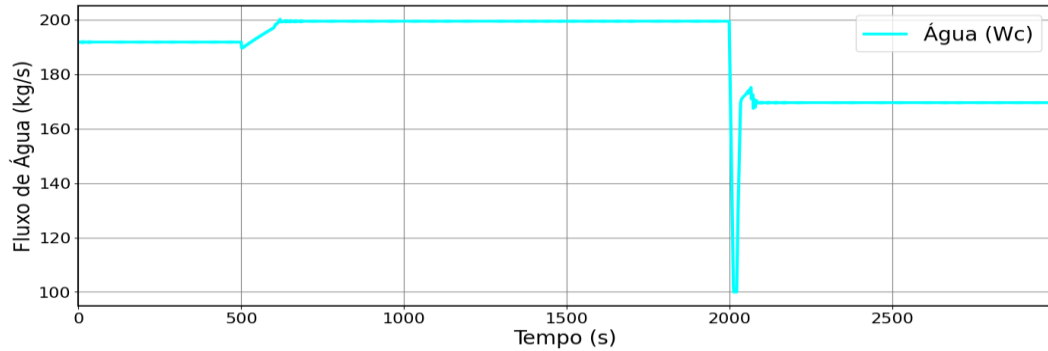
Fonte: elaborado pelo próprio autor.

Figura 37 - Fluxo de combustível para mudança na referência com controle *Fuzzy*.



Fonte: elaborado pelo próprio autor.

Figura 38 - Fluxo de água para mudança na referência com controle *Fuzzy*.

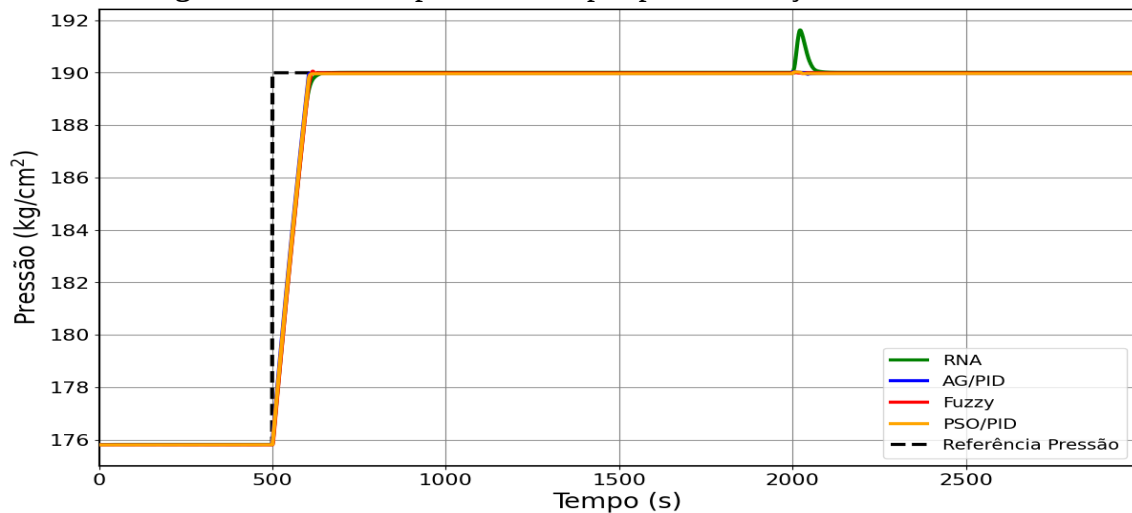


Fonte: elaborado pelo próprio autor.

4.4 ANÁLISE DO DESEMPENHO DOS CONTROLADORES

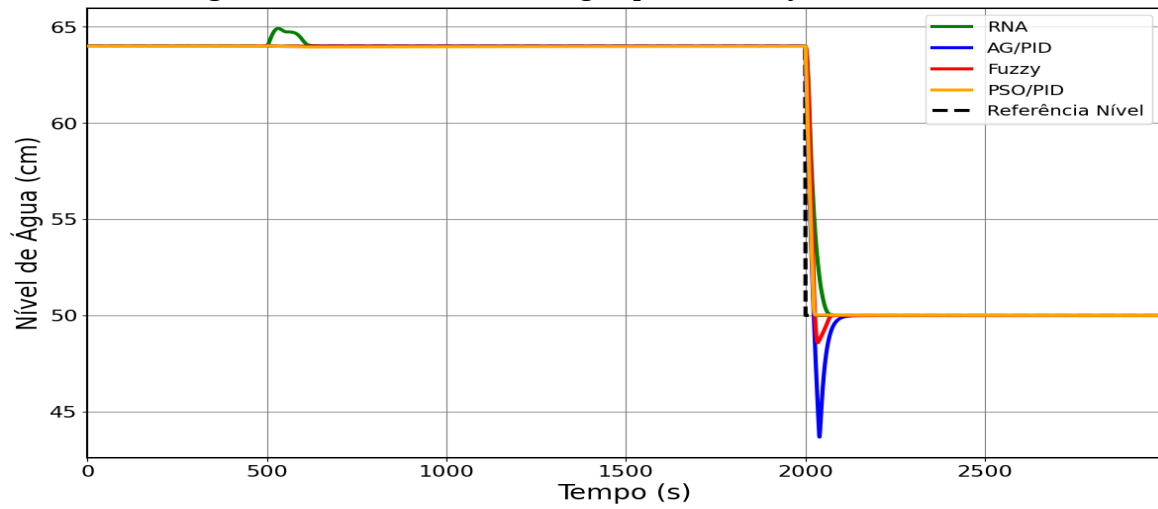
A seguir mostram-se o desempenho dos controladores: RNA, *Fuzzy*, e PID sintonizados por PSO e AG. Nas Figuras 39 e 42, apresenta-se a resposta da pressão de vapor e do nível de água ao longo do tempo, considerando duas mudanças na referência, uma aos 500 s e outra aos 2000 s. Nessas Figuras, observa-se o seguimento nas referências em ambas saídas e o nível de interação entre as variáveis, em que uma mudança de referência na pressão tem impacto na saída de nível e vice-versa.

Figura 39- Saída de pressão de vapor para mudança na referência.



Fonte: elaborado pelo próprio autor.

Figura 40 - Saída de nível de água para mudança na referência.



As Figuras 41 e 42 mostram o fluxo de combustível e o fluxo de água para as mudanças na referência estabelecidas.

Figura 41 - Fluxo de combustível para mudança na referência.

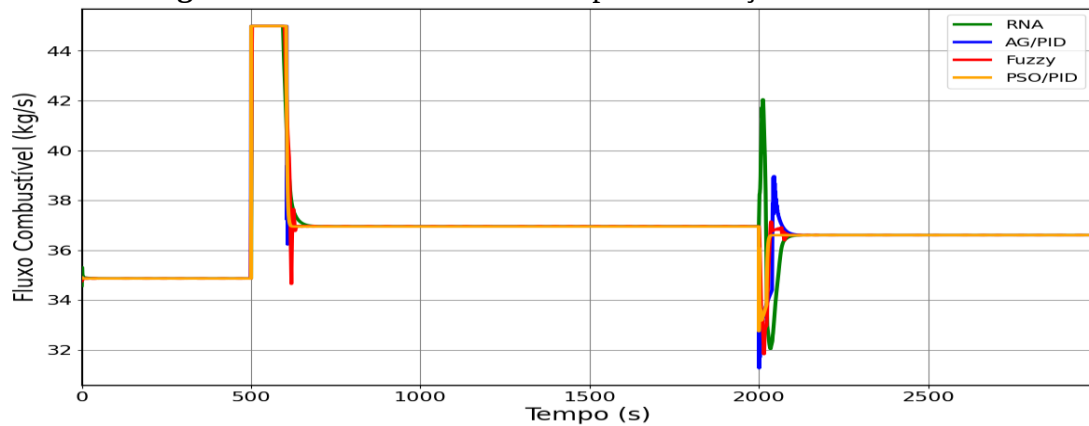
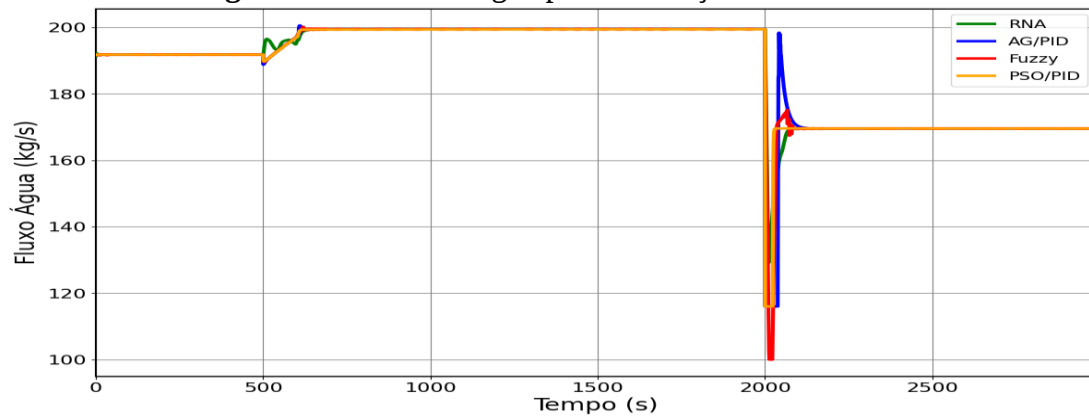


Figura 42 - Fluxo de água para mudança na referência.



Nas Tabelas 5 e 6 mostram-se os índices de desempenho para cada saída. Como observa-se, os controladores PID sintonizados por PSO e AG têm uma superioridade em relação ao controle por RNA, com valores menores de índices de desempenho. O controlador *Fuzzy* demonstrou um desempenho equilibrado e aceitável em ambas as saídas, com tempos de resposta (t_s e t_r) comparáveis aos dos PID.

Tabela 5 - Índices de desempenho para saída de pressão para mudança na referência.

Índices de desempenho	Pressão RNA	Pressão PSO PID	Pressão GA PID	Pressão <i>Fuzzy</i>
Tempo de acomodação (t_s)	114s	104,2s	104,18s	107s
Tempo de subida (t_r)	85s	85,12s	85,14s	86s
Máximo sobressinal (M_p)	0%	0,26%	0,2%	0,41%
Erro estacionário (e_{ss})	0,0170	0	-0,022	-0,0037
ISE	7082,42	6758,68	6757,46	7116,53
IAE	853,81	810,71	789,40	765,6538
ITAE	581461,31	533388	496435	420722,08
ITSE	3809015,39	3556472	3554353	3741032,37
IAU	109901,18	251379,82	109872,21	109881,23

Fonte: elaborado pelo próprio autor.

Tabela 6 - Índices de desempenho para saída de nível de água para mudança na referência.

Índices de desempenho	Nível de Água RNA	Nível de Água PSO PID	Nível de Água GA PID	Nível de Água Fuzzy
Tempo de acomodação (t_s)	57s	23,48s	85,89s	63s
Tempo de subida (t_r)	37s	19,01s	19,01s	17s
Máximo sobressinal (M_p)	0%	0,08%	43,81%	10,16%
Erro estacionário (e_{ss})	0,0046	0	0	0,00028534
ISE	2905,2	1392,02	1946,37	2361,18
IAE	394,84	204,74	306,86	251,70
ITAE	690260,45	381324,02	619808,81	506138,12
ITSE	5768706,82	2790644,19	3923856,77	4739216,24
IAU	5,61(10^5)	1,49(10^5)	5,61(10^5)	5,61(10^5)

Fonte: elaborado pelo próprio autor.

A seguir, algumas observações sobre esse análises comparativo entre controladores:

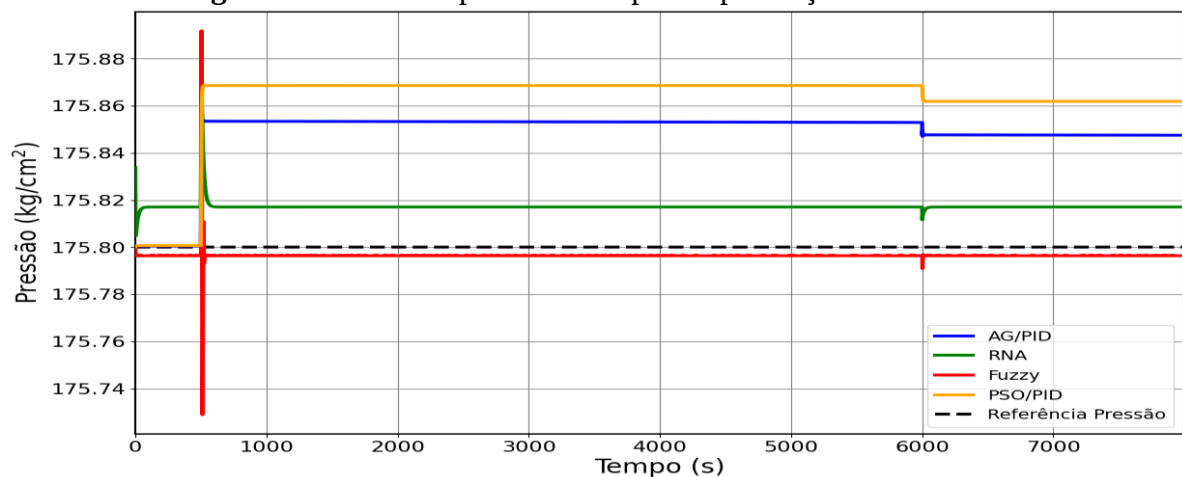
- Todas as estratégias de controle apresentam um seguimento na referência com erros relativos menores do que 0,02%.
- Os valores de ITAE e ITSE são elevados, indicando um desempenho menos satisfatório na soma dos erros quadráticos e de erro absoluto acumulados ao longo do tempo.
- PID + Otimizadores (AG/PSO): o AG e o PSO encontraram os ganhos (K_p , K_i , T_d) ideais para esse ponto de operação específico. Eles "linearizam" o controle de forma ótima para a região no qual o sistema está trabalhando. O PID resultante é um controlador robusto e eficiente para essa condição.
- Uma característica crucial a ser considerada é a função de recompensa no controle por RNA, pois ela fornece o *feedback* necessário para guiar a política do agente durante o treinamento por reforço. A recompensa foi projetada para ser máxima quando o erro das variáveis controladas é inferior a 0,1, sem uma punição ou recompensa adicional por valores ainda menores. Apesar da ausência de um incentivo direto para atingir um erro nulo, o controlador neuronal demonstrou um desempenho superior ao esperado, conseguindo reduzir o erro para valores abaixo de 0,02.
- Pode ser revisada a arquitetura da rede: a escolha do número de camadas, neurônios,

funções de ativação e algoritmo de treinamento é crucial para o desempenho da RNA.

- Resultados de velocidade e erro acumulado: O controle *Fuzzy* alcançou tempos de acomodação e subida competitivos com os controladores PID.
- Máximo sobressinal: Um ponto de atenção é o máximo sobressinal de 0,41% e 10,16% nas respectivas saídas usando o controle *Fuzzy*. Isso sugere que as regras e conjuntos nebulosos podem ter sido ajustados de forma mais agressiva ou menos otimizada para o regime transitório, resultando em uma resposta ligeiramente mais oscilatória na busca pela referência.

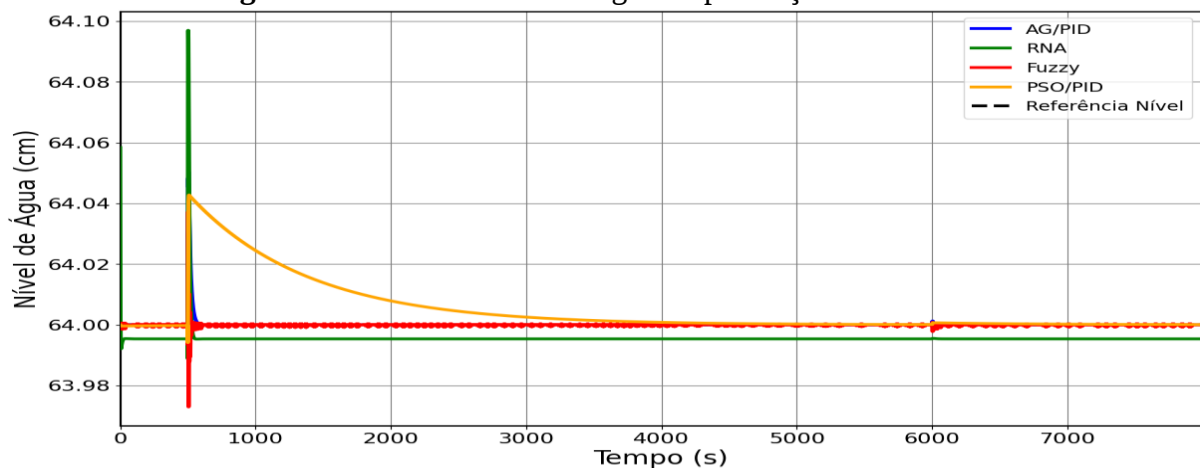
As Figuras 43 e 44 mostram o desempenho dos controladores frente a um distúrbio de -10% nas variáveis C_v e T_e . O distúrbio em C_v (0,72) ocorre aos 500s e em T_e (279 °C) aos 6000s. Todos os controladores rejeitam o distúrbio de forma adequada.

Figura 43 - Saída de pressão de vapor na presença de distúrbios.



Fonte: elaborado pelo próprio autor.

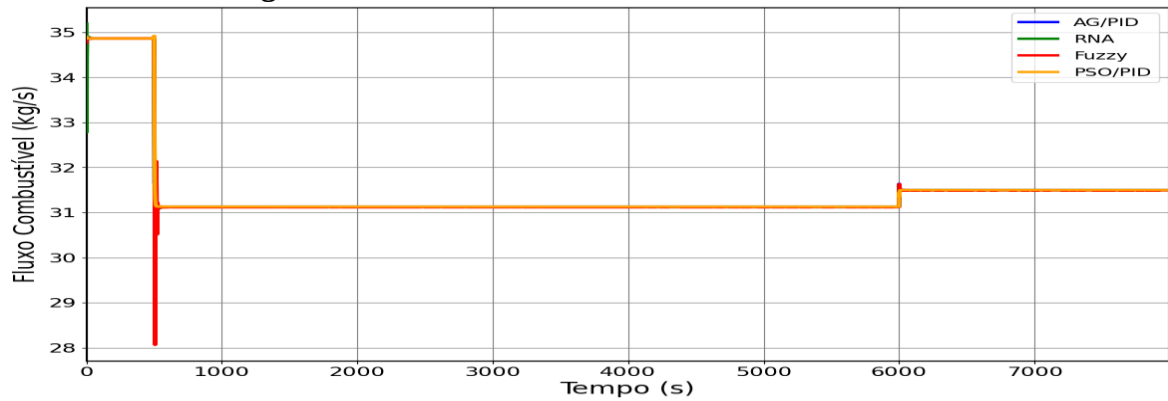
Figura 44 - Saída de nível de água na presença de distúrbios.



Fonte: elaborado pelo próprio autor.

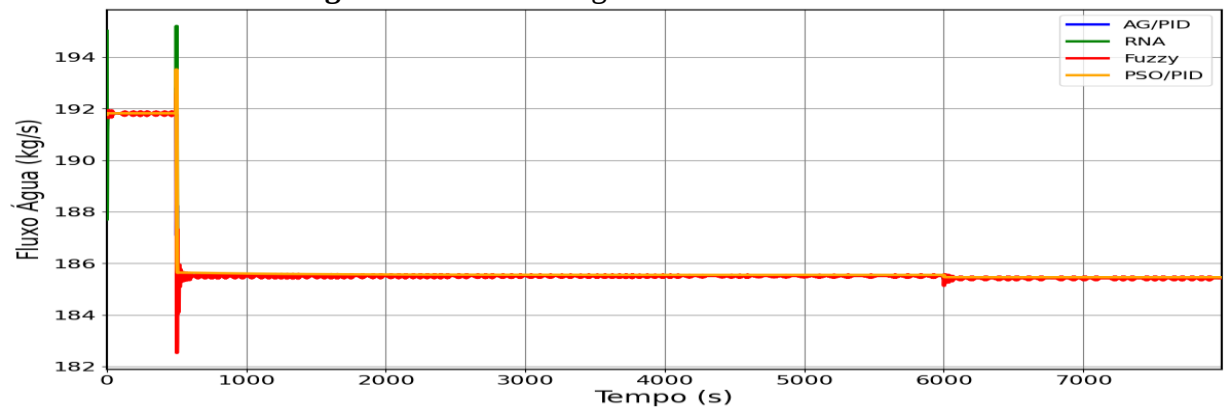
As Figuras 45 e 46 mostram o fluxo de combustível e do fluxo de água para os diferentes controladores frente ao distúrbio. Nas Tabelas 9 e 10 mostram-se os índices de desempenho para cada saída.

Figura 45 - Fluxo de combustível frente aos distúrbios.



Fonte: elaborado pelo próprio autor.

Figura 46 - Fluxo de água frente aos distúrbios.



Fonte: elaborado pelo próprio autor.

Tabela 7 - Índices de desempenho para saída de pressão frente ao distúrbio.

Índices de desempenho	Pressão RNA	Pressão PSO PID	Pressão AG PID	Pressão Fuzzy
ISE	2,38	33,4960	20,0544	0,16
IAE	136,85	500,66	387,39	29,93
ITAE	543636,04	2088410	1611070	116713,89
ITSE	9281,53	137490	81898	452,83
IAU	251290,61	251379,82	251331,16	251271,31
Erro Estacionário (e_{ss})	0,02	0,07	0,05	0,00

Fonte: elaborado pelo próprio autor.

Tabela 8 - Índices de desempenho para saída de nível de água frente ao distúrbio.

Índices de desempenho	Nível de Água RNA	Nível de Água PSO PID	Nível de Água AG PID	Nível de Água Fuzzy
ISE	0,23	0,8162	0,03	0,0052
IAE	37,86	38,62	1,02	1,78
ITAE	147988,9	55834,56	598,91	5708,48
ITSE	712,9	772,41	14,78	4,26
IAU	1,49(10 ⁶)	1,49(10 ⁶)	1,49(10 ⁶)	1,49(10 ⁶)
Erro Estacionário	0,0046	0	0	0

Fonte: elaborado pelo próprio autor.

Com base nas Tabelas 7 e 8, que apresentam os índices de desempenho para a saída de pressão e nível de água frente a um distúrbio, observa-se que o controlador *Fuzzy* se destaca como o mais eficiente neste cenário com os menores valores para todos os índices integrais e o $e_{ss} = 0$, com exceção do ITAE e IAE para a saída de nível, indicando nesse caso um desempenho menos satisfatório nos erros absolutos acumulados ao longo do tempo.

5 CONCLUSÕES E TRABALHOS FUTUROS

No presente trabalho foram estudadas as características das redes neurais, da lógica *Fuzzy* e dos controladores PID, assim como os conceitos básicos dos geradores de vapor. Seguidamente se realiza o estudo da modelagem do sistema dinâmico, obtendo um modelo em espaço de estados que se mostrou satisfatório para representar o comportamento dinâmico do gerador. Com base nesse modelo, foram desenhadas e implementadas via simulação cinco estratégias de controle: Redes Neurais Artificiais, Controladores PID otimizados por Otimização por Enxame de Partículas e Algoritmos Genéticos e Lógica *Fuzzy*. Finalmente apresenta-se um estudo comparativo aprofundado das estratégias de controle aplicadas ao gerador, evidenciando-se que todos os controladores apresentam um seguimento na referência com erros relativos menores do que 0,02%. Os controladores PID sintonizados por PSO e AG têm uma superioridade sobre os outros métodos de controle utilizados no seguimento da referência, enquanto para rejeição ao distúrbio, o controle *Fuzzy* apresenta um melhor desempenho.

A principal dificuldade encontrada no decorrer da pesquisa foi a complexidade inerente ao projeto, treinamento e ajuste fino do controlador por RNA. Diferentemente das abordagens de otimização direta (como o AG PID e PSO PID), a implementação da RNA exigiu o uso de aprendizagem por reforço para otimizar a política de controle. Esta metodologia, baseada em tentativa e erro para maximizar recompensas a longo prazo, impôs desafios significativos na definição da função de recompensa, na seleção da arquitetura neural ideal e no volume de interações (episódios) necessárias para convergir para uma política estável e eficiente. O custo computacional e o tempo dedicado à robustez do treinamento por RL foram notavelmente elevados.

Por fim, sugerem-se como trabalhos futuros a validação experimental dos controladores em uma bancada real de gerador de vapor, a fim de investigar os desafios da implementação em tempo real. Adicionalmente, recomenda-se explorar outros tipos de controladores como o Controle Preditivo Não Linear. Sugere-se também uma modelagem do sistema utilizando árvores de decisão ou outros tipos de sintonia para outros controladores como a sintonia por enxame de formigas ou os algoritmos inspirados nos comportamentos físicos.

REFERÊNCIAS

ANNARATONE, Donatello. **Steam generators: description and design**. Berlin: Springer, 2008.

BANNLOUD, Mohamad Al; SANTOS, Brunno Ferreira dos; MARTINS, Tiago Dias. Controle preditivo por modelo de rede neural aplicado a processos de fermentação e destilação. **Revista de Engenharia e Tecnologia**, v. 13, n. 3, p. 168, set. 2021. Disponível em: <https://revistas.uepg.br/index.php/ret/article/view/18480>. Acesso em: 15 mar. 2025.

BERTSEKAS, D. P. **Dynamic programming and optimal control**. 4. ed. Belmont: Athena Scientific, 2012. v. 1.

BOBÁL, V. *et al.* **Digital self-tuning controllers: algorithms, implementation and applications**. [s. l.]: Springer Science & Business Media, 2006. Disponível em: <https://www.uiam.sk/pc05/data/books/bobal05/contents.pdf>. Acesso em: 2 abr. 2026.
BOILER: definition, principle & working, properties, classification & component. [s. l.], 2020. Disponível em: <https://mechanicalnotes.com/boiler-definition-working-properties-component/>. Acesso em: 14 mar. 2024.

BONALA, Sath Yan. **A study on neural network based system identification with application to heating, ventilating and air conditioning (HVAC) system**. 2009. Dissertation (Master in Technology) – National Institute of Technology, Rourkela, 2009. Disponível em: <https://scispace.com/pdf/a-study-on-neural-network-based-system-identification-with-29pupa94yv.pdf>. Acesso em: 2 abr. 2026.

BÜTTENBENDER, Felipe Brunetto. **Previsão da eficiência de geração de vapor por meio de redes neurais artificiais com diferentes acoplamentos**. 2021. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2021. Disponível em: <https://lume.ufrgs.br/handle/10183/223610>. Acesso em: 15 mar. 2025.

CERVANTES, Lussón Ania. **Control Predictivo baseado em Modelos com estrutura Wiener-LAT**. 2002. Tese (Doutorado em Controle de Sistemas) – Universidad Nacional del Sur, Bahía Blanca, 2002.

CHEN, G.; YIN, D.; CHEN, F. Collaborative multi-objective optimization of combustion and emissions in circulating fluidized bed boilers. **Sustainability**, v. 17, n. 11, 2025. Disponível em: <https://doi.org/10.3390/su17115225>. Acesso em: 2 abr. 2026.

DORF, Richard C.; BISHOP, Robert H. **Modern control systems**. 13. ed. Boston: Pearson, 2016.

FIGUEIREDO, Jonathas Marcelo Pereira; REJAILI, Rodrigo Pereira Abou. **Aplicação de algoritmos de aprendizagem por reforço para controle de navios em águas restritas**. 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia Naval) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2018. Disponível em: <https://bdta.abcd.usp.br/directbitstream/039f9a0c-ee2e-4822-92af-6492a2594c34/Rodrigo%20Rejaili%20%20-%20Jonathas%20Figueiredo%20-%20monografia.pdf>. Acesso em: 2 abr. 2026.

FINOCCHIO, Marco Antonio Ferreira. **Noções de redes neurais artificiais**. 2014. Apostila – Cornélio Procópio: UTFPR, 2014. Disponível em: <http://paginapessoal.utfpr.edu.br/mafinocchio/labsi-laboratorio-de-seguranca-e-iluminacao/redes-neurais-artificiais/NOCaO%20DE%20REDES%20NEURAS%20ARTIFICIAIS.pdf>. Acesso em: 2 abr. 2026.

FLECK, Leandro *et al.* Redes neurais artificiais: princípios básicos. **Revista de Engenharia e Ciência da Computação**, v. 7, n. 15, 2016. Disponível em: <https://periodicos.utfpr.edu.br/recit/article/view/4330>. Acesso em: 15 mar. 2025.
FURTADO, Inês Vasconcellos. **Redes neurais artificiais: uma abordagem para sala de aula**. Ponta Grossa: Atena Editora, 2019.

GUIMARÃES, Breno Henrique Corrêa. **Modelagem e simulação dinâmica de uma caldeira industrial aquatubular utilizando o Simulink**. 2023. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Uberlândia, Uberlândia, 2023. Disponível em: <https://repositorio.ufu.br/handle/123456789/33183>. Acesso em: 15 mar. 2025.

HAARNOJ, T. *et al.* Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML)*, 35., 2018. **Anais [...]**. [s. l.]: PMLR, 2018.

HUSSEIN, Ahmed S. *et al.* Intelligent energy management of steam generators. **arXiv preprint**, [s. l.], 3 jun. 2024. Disponível em: <https://doi.org/10.48550/arXiv.2406.16904>. Acesso em: 15 mar. 2025.

JIBRIL, Mustefa; TADESE, Messay; TADESE, Eliyas Alemayehu. Design and control of steam flow in cement production process using neural network based controllers. **Researcher**, v. 12, n. 5, p. 76-84, 2020. Disponível em: https://www.sciencepub.net/researcher/rsj120520/09_36482rsj120520_76_84.pdf. Acesso em: 15 mar. 2025.

KRIDI, Douglas *et al.* Desenvolvimento de uma biblioteca fuzzy para o controle autônomo de um robô móvel em ambiente desconhecido. *In: MOSTRA NACIONAL DE ROBÓTICA (MNR)*, 1., 2011, São João Del-Rei. **Anais [...]**. São João Del-Rei: UFSJ, 2011. v. 1. Disponível em: https://www.researchgate.net/publication/259310399_Desenvolvimento_de_Uma_Biblioteca_Fuzzy_para_o_Controle_Autonomo_de_Um_Robo_Movel_em_Ambiente_Desconhecido. Acesso em: 2 abr. 2026.

KULLBACK, S.; LEIBLER, R. A. On Information and Sufficiency. **The Annals of Mathematical Statistics**, [s. l.], v. 22, n. 1, p. 79-86, 1951. Disponível em: <https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694.full>. Acesso em: 2 abr. 2026.

KURKIN, Andrii *et al.* Forecasting steam mass flow in power plants using the parallel hybrid network. **Applied Soft Computing**, v. 160, p. 111075, 2023. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0952197625019141>. Acesso em: 15 mar. 2025.

LI, Cheng *et al.* Reinforcement learning-based control with application to the once-through steam generator system. **Nuclear Engineering and Technology**, v. 55, n. 10, p. 3515-3524, out. 2023. Disponível em: <https://doi.org/10.1016/j.net.2023.06.001>. Acesso em: 15 mar. 2025.

LIU, Jie *et al.* Application of a Neural Network on Steam Drum Pressure Control in a Fixed-Bed Reactor for Exothermic Reactions. **Applied Sciences**, v. 14, n. 7, p. 3052, 5 abr. 2024. Disponível em: <https://doi.org/10.3390/app14073052>. Acesso em: 9 set. 2025.

MARQUES, F. A. P. **Modelagem e controle de nível do tubulão de uma caldeira de vapor aquatubular de uma refinaria de petróleo**. 2005. Dissertação (Mestrado em Engenharia Química) – Universidade Federal de Minas Gerais, Belo Horizonte, 2005. Disponível em: <https://electra.pee.ufrj.br/pee/index.php/pt/producao-academica/dissertacoes-de-mestrado/2005-1/2005040801-2005040801/file>. Acesso em: 2 abr. 2026.

OLIVEIRA JUNIOR, Valter Barbosa de. **Modelagem de central termelétrica a vapor para simulação dinâmica**. 2009. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Espírito Santo, Vitória, 2009. Disponível em: <https://abmproceedings.com.br/pt-br/article/modelagem-de-central-termeltrica-vapor-para-simulao-dinmica>. Acesso em: 15 mar. 2025.

QUEIROZ, T. L. **Implementação de um sistema de automação, controle e supervisão da combustão em uma caldeira**. 2017. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade Federal do Rio Grande do Norte, Natal, 2017. Disponível em: <https://revista.unifacear.edu.br/rem/article/download/427/420/842>. Acesso em: 2 abr. 2026.

RIZZO, Igor Vilela; CANATO, Robson Leandro Carvalho. Inteligência artificial: funções de ativação. **Prospectus**, v. 2, n. 2, 2020. Disponível em: <https://prospectus.fatecitapira.edu.br/index.php/pst/article/view/37>. Acesso em: 15 mar. 2025.

ROMANOV, Anton A.; FILIPPOV, Aleksey A.; YARUSHKINA, Nadezhda G. An Approach to Generating Fuzzy Rules for a Fuzzy Controller Based on the Decision Tree Interpretation. **Axioms**, v. 14, n. 3, p. 196, mar. 2025. Disponível em: <https://doi.org/10.3390/axioms14030196>. Acesso em: 30 nov. 2025.

RONCAGLIO, Mateus Maruzka. **Otimização por enxame de partículas aplicado em controladores PI**. 2020. Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica) – Universidade Tecnológica Federal do Paraná, Toledo, 2020. Disponível em: <https://portaldeinformacao.utfpr.edu.br/Record/riut-1-35279>. Acesso em: 2 abr. 2026.

SCHWERTNER, Anderson Ervino. **O método de Levenberg-Marquardt para problemas de otimização de menor valor ordenado**. 2019. Dissertação (Mestrado em Matemática) – Universidade Estadual de Maringá, Maringá, 2019. Disponível em: http://repositorio.uem.br:8080/jspui/bitstream/1/5746/1/Anderson%20Ervino%20Schwertner_2019.pdf. Acesso em: 2 abr. 2026.

ŠTEFEK, Alexandr *et al.* Optimization of Fuzzy Logic Controller Used for a Differential Drive Wheeled Mobile Robot. **Applied Sciences**, v. 11, n. 13, p. 6023, jun. 2021. Disponível em: <https://www.mdpi.com/2076-3417/11/13/6023>. Acesso em: 30 nov. 2025.

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: an introduction**. 2. ed. Cambridge: MIT Press, 2018.

TAVARES, P. A. N. **Análise de diferentes tipos de ajuste de um controlador PID aplicado a um motor CC sem escovas**. 2019. Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica) – Universidade de Brasília, Brasília, 2019. Disponível em: https://bdm.unb.br/bitstream/10483/26043/1/2019_PericlesAlvesNery_tcc.pdf. Acesso em: 2 abr. 2026.

WANG, Q.; PAN, L.; LEE, K. Y. Deep-learning modeling and control optimization framework for intelligent thermal power plants: a practice on superheated steam temperature. **Korean Journal of Chemical Engineering**, v. 38, p. 1983-2002, 2021. Disponível em: <https://doi.org/10.1007/s11814-021-0865-6>. Acesso em: 15 mar. 2025.

WINDMANN, Alexander *et al.* Artificial intelligence in Industry 4.0: a review of integration challenges for industrial systems. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN), 22., 2024. **Proceedings** [...]. [s. l.]: IEEE, 2024. p. 1-8.

WRIGHT, Sarah; MARWALA, Tshilidzi. Artificial intelligence techniques for steam generator modelling. **arXiv preprint**, [s. l.], 13 nov. 2008. Disponível em: <https://doi.org/10.48550/arXiv.0811.1711>. Acesso em: 15 mar. 2025.