Uniube

CONTROLADOR INTELIGENTE NEURO-FUZZY(ANFIS) PARA CALDEIRA DE DESTILARIA

Acadêmico do Curso da Computação: Nelson de Paiva Dagrava
Acadêmico do Curso da Computação: Rodrigo dos Reis Silva Filho
Docente Orientador do Curso de Engenharia da Computação: Marcelo Lucas

Universidade de Uberaba – Uniube, Uberaba – MG, Brasil

nelsonpaiva2002@outlook.com

RESUMO

Este trabalho propõe o desenvolvimento de um controlador neurofuzzy (ANFIS) para otimizar o controle de temperatura em uma caldeira de destilação, combinando redes neurais e lógica fuzzy para criar um sistema adaptativo. Segundo **Jang, Sun e Mizutani (1997)**, a arquitetura ANFIS é ideal para sistemas dinâmicos, pois une o aprendizado das redes neurais à lógica fuzzy, permitindo maior precisão e adaptabilidade. A metodologia inclui a modelagem matemática da caldeira, coleta de dados no Google Colab e treinamento do modelo no MATLAB, utilizando o ANFIS Editor. Dados pré-processados foram utilizados, com variáveis de entrada como erro (SP - PV), processo (PV) e derivada, e uma saída manipulada (MV). Funções de pertinência gaussianas e o treinamento híbrido com 1.000 épocas garantiram um desempenho robusto. A configuração mais enxuta (três entradas e uma saída) demonstrou equilíbrio entre simplicidade e precisão, validando a eficácia do modelo em condições dinâmicas e não lineares. O projeto prevê a implementação prática do controlador em uma plataforma Arduino para validação em condições reais. Essa etapa é fundamental para demonstrar a aplicabilidade do controlador no ambiente industrial, assegurando maior eficiência energética, menor intervenção manual e maior estabilidade. Os resultados confirmam o potencial dos sistemas neurofuzzy na automação industrial, como sugerido por **Zimmermann (2001)**, destacando sua relevância para modernizar processos em setores como química, petroquímica e alimentício.

Palavras-chave: Controle neuro-fuzzy; ANFIS; Automação industrial; Controle de temperatura; Sistemas inteligentes; Modelagem de sistemas; Controle adaptativo; Otimização de processos; Arduino.

Uniube

ABSTRACT

This work proposes the development of a neuro-fuzzy controller (ANFIS) to optimize temperature control in a distillation boiler, combining neural networks and fuzzy logic to create an adaptive system. According to Jang, Sun, and Mizutani (1997), the ANFIS architecture is ideal for dynamic systems, as it integrates the learning capabilities of neural networks with the interpretive flexibility of fuzzy logic, enabling greater precision and adaptability. The methodology includes the mathematical modeling of the boiler, data collection using Google Colab, and model training in MATLAB with the ANFIS Editor. Pre-processed data were utilized, incorporating input variables such as error (SP - PV), process (PV), and derivative, as well as a manipulated variable (MV) as output. Gaussian membership functions and hybrid training with 1,000 epochs ensured robust performance. A simplified configuration (three inputs and one output) demonstrated an optimal balance between simplicity and precision, validating the model's effectiveness under dynamic and nonlinear conditions. The project also includes the practical implementation of the controller on an Arduino platform to validate its performance in real-world conditions. This step is essential to demonstrate the controller's applicability in industrial environments, ensuring higher energy efficiency, reduced manual intervention, and greater stability. The results confirm the potential of neuro-fuzzy systems in industrial automation, as highlighted by Zimmermann (2001), emphasizing their relevance in modernizing processes across sectors such as chemical, petrochemical, and food industries.

Keywords: Neuro-fuzzy control; ANFIS; Industrial automation; Temperature control; Intelligent systems; System modeling; Adaptive control; Process optimization; Arduino.

1. INTRODUÇÃO

O controle de processos industriais é essencial para garantir eficiência, qualidade e sustentabilidade em setores como destilarias, petroquímica, farmacêutica e alimentícia. Em ambientes complexos e dinâmicos, o ajuste preciso de variáveis como temperatura, pressão e vazão é crucial para o funcionamento adequado dos processos e a redução de custos operacionais. No entanto, sistemas de controle convencionais, baseados em modelos fixos, frequentemente não conseguem lidar com a não-linearidade e as rápidas mudanças inerentes aos processos industriais. Nesse contexto, soluções baseadas em inteligência artificial (IA) têm emergido como abordagens promissoras para superar essas limitações. Entre as técnicas de IA, o Sistema Neuro-fuzzy adaptativo (ANFIS - Adaptive Neuro-Fuzzy Inference System) destaca-se por combinar o aprendizado das redes neurais com o raciocínio aproximado da lógica fuzzy, permitindo modelar sistemas dinâmicos e não-lineares com alta precisão. Como destacado por **Jang, Sun e Mizutani (1997)**, o ANFIS é capaz de ajustar automaticamente seus parâmetros com base em

Uniube

dados históricos, tornando-se especialmente útil em aplicações como o controle de temperatura de caldeiras em destilarias. Pequenas variações térmicas nesse processo podem comprometer a qualidade do produto final e aumentar o consumo energético, ressaltando a necessidade de sistemas inteligentes e adaptativos. Além disso, a preparação de dados desempenha um papel fundamental no sucesso de modelos baseados em IA. Ferramentas como o Google Colab permitem realizar o pré-tratamento dos dados, eliminando ruídos, inconsistências e valores ausentes, enquanto técnicas de análise exploratória ajudam a identificar padrões e tendências. Dados de qualidade são essenciais para garantir o treinamento eficaz do ANFIS e melhorar a capacidade do modelo em responder às variações operacionais. Este trabalho propõe o desenvolvimento de um controlador neuro-fuzzy adaptativo (ANFIS) para o controle de temperatura em caldeiras de destilarias, utilizando dados simulados e reais. A metodologia inclui o pré-tratamento de dados no Google Colab, o treinamento e ajuste do modelo no MATLAB, e a validação prática do controlador em uma plataforma Arduino. A implementação busca demonstrar a aplicabilidade do ANFIS em condições reais, contribuindo para a modernização dos processos industriais. Ao final, este estudo pretende evidenciar os benefícios dos sistemas inteligentes no aumento da eficiência, sustentabilidade e qualidade em setores estratégicos. O sistema investigado neste trabalho é uma caldeira de destilação, cujo controle de temperatura é realizado por meio de um controlador inteligente baseado no modelo **ANFIS (Sistema Neuro-Fuzzy Adaptativo)**. Esse modelo combina as vantagens de duas tecnologias de inteligência artificial: as redes neurais artificiais (RNAs), conhecidas por sua capacidade de aprendizado automático, e a lógica fuzzy, que é eficaz para lidar com a incerteza e a não-linearidade dos processos. Essa integração permite que o controlador proporcione um ajuste adaptativo e preciso da temperatura, mesmo diante de variações significativas no processo de destilação. O objetivo principal é manter a temperatura da caldeira dentro do intervalo ideal de **49°C a 51°C**, garantindo estabilidade operacional e eficiência energética. A definição desse intervalo de temperatura como setpoint é fundamental para o processo de destilação, pois a estabilidade térmica influencia diretamente a separação adequada dos componentes do fluido, a eficiência do sistema e a qualidade do produto final. Oscilações fora desse intervalo podem comprometer o rendimento, aumentar o consumo energético ou até mesmo causar falhas operacionais. O sistema desenvolvido busca eliminar essas oscilações, otimizando o funcionamento da caldeira com o mínimo de intervenção manual, um benefício crucial em ambientes industriais que demandam alta precisão. Um dos desafios enfrentados neste trabalho foi a dificuldade em obter dados reais de operação, devido à negativa de fornecimento de informações por parte das empresas do setor. Para superar essa limitação, foi utilizado um simulador desenvolvido pela equipe da Universidade federal de minas gerais, que replicou as condições típicas de operação de uma caldeira de destilação. Esse simulador gerou os dados necessários para o treinamento e a validação do modelo ANFIS, permitindo que o controlador aprendesse padrões relevantes e representativos do processo, mesmo sem acesso a dados reais. O simulador foi configurado para incluir variações dinâmicas comuns, como mudanças no fluxo de calor,

Uniube

condições ambientais e comportamento do sistema, garantindo que o treinamento fosse robusto. A arquitetura geral do sistema combina sensores de temperatura de alta precisão, atuadores responsáveis pelo controle do fornecimento de energia e um microcontrolador Arduino, no qual o modelo ANFIS é implementado. O controlador neuro fuzzy é treinado para correlacionar dados de entrada, como temperatura inicial e fluxo de calor, com os ajustes necessários para alcançar e manter o setpoint desejado. Uma vez treinado, o modelo pode ajustar automaticamente os parâmetros do sistema em tempo real, minimizando desvios e otimizando o desempenho da caldeira. Além de controlar a temperatura, o sistema também foi projetado para operar com alta eficiência energética. Ajustando continuamente o fornecimento de energia com base nas condições atuais do processo, o controlador reduz o desperdício de energia e, consequentemente, os custos operacionais. Essa abordagem também está alinhada com as demandas modernas de sustentabilidade industrial, reduzindo o impacto ambiental e promovendo o uso responsável de recursos energéticos. Portanto, o sistema descrito neste trabalho representa uma aplicação prática do modelo ANFIS em cenários industriais, mostrando como soluções baseadas em inteligência artificial podem superar as limitações de métodos de controle convencionais. Mesmo em condições desafiadoras, como a ausência de dados reais, a abordagem baseada em simulação demonstrou ser uma alternativa viável para o desenvolvimento de sistemas inteligentes, contribuindo para a inovação em processos industriais críticos, como a destilação.

2. METODOLOGIA

Os materiais utilizados neste projeto incluem:

- **Google Colab:** Uma plataforma de computação em nuvem que desempenhou um papel central no pré-tratamento e análise dos dados utilizados no projeto. Por meio do uso de bibliotecas poderosas como Pandas, NumPy e Matplotlib, o Colab permitiu a manipulação e visualização de dados de forma eficiente, além de oferecer um ambiente colaborativo para execução de scripts Python otimizados para machine learning e data science. Neste projeto, o Colab foi utilizado para realizar a limpeza dos dados, removendo valores inconsistentes ou ausentes que poderiam comprometer o treinamento do modelo ANFIS, além da normalização e padronização das variáveis de entrada, garantindo escalas compatíveis e otimizando o desempenho do modelo. A plataforma também foi essencial para a análise exploratória, possibilitando a identificação de padrões, correlações e anomalias nos dados, o que auxiliou na criação de regras fuzzy mais representativas. Por fim, os dados pré-tratados foram exportados em formatos compatíveis para integração com o MATLAB, garantindo uma transição fluida para as etapas de simulação e treinamento do modelo.

Uniube

- **MATLAB**:Principal ferramenta utilizada para o treinamento e validação do modelo ANFIS, destacando-se por sua interface intuitiva e recursos avançados, especialmente por meio do **ANFIS Editor (anfisedit)**. Essa ferramenta gráfica permitiu configurar o número de regras fuzzy, ajustar funções de pertencimento e realizar simulações detalhadas para avaliar o desempenho do sistema. Além disso, o MATLAB possibilitou a realização de análises de desempenho, como o cálculo de métricas de erro, incluindo o erro médio absoluto (MAE) e o erro quadrático médio (RMSE), garantindo a eficácia do modelo em diferentes cenários. O software também foi fundamental para a visualização dos resultados por meio de gráficos tridimensionais, que ilustraram o comportamento do sistema em relação às entradas e saídas simuladas. Assim, o MATLAB não apenas otimizou o treinamento e a validação do controlador neurofuzzy, mas também permitiu a simulação de condições variadas, preparando o sistema para sua implementação prática.

- **Plataforma embarcada**: Após o treinamento e validação no MATLAB, o modelo foi implementado em um Arduino, que serviu como microcontrolador principal do sistema. Ele é responsável por executar as regras fuzzy geradas pelo modelo ANFIS e ajustar o fornecimento de energia à caldeira com base nas leituras de temperatura.

- **Sensor de temperatura**: Esses sensores são utilizados para medir a temperatura em tempo real na caldeira. As leituras obtidas são fundamentais para a retroalimentação do sistema, garantindo que os ajustes realizados pelo controlador mantenham a temperatura dentro do intervalo de 49°C a 51°C.

- **Computador**: Equipamento utilizado para todas as etapas do projeto, incluindo o pré-tratamento dos dados no Google Colab, o treinamento e simulação no MATLAB e a programação do Arduino. Ele também foi usado para monitorar o desempenho do sistema durante os testes práticos.

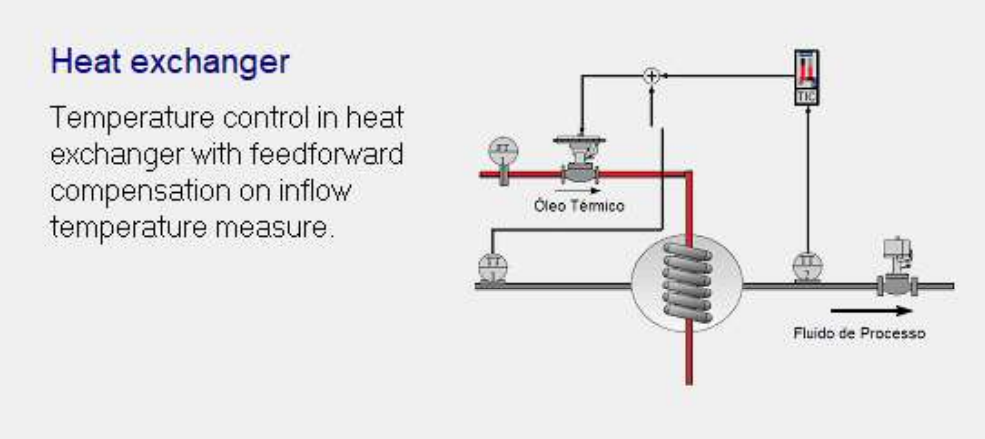
O desenvolvimento do controlador neuro-fuzzy será realizado em três métodos principais:

- **Coleta e Tratamento de Dados**: A primeira etapa deste trabalho envolveu a coleta e o tratamento dos dados, fundamentais para o desenvolvimento do modelo neuro fuzzy. Dado que o acesso a dados reais de processos industriais não foi possível, utilizou-se um simulador baseado no sistema de controle representado na figura, que descreve um trocador de calor com controle de temperatura e compensação feedforward. Esse simulador foi projetado para reproduzir condições típicas de operação industrial, incluindo a interação entre a temperatura de entrada e saída do fluido de processo, o fluxo de óleo térmico utilizado para troca de calor e perturbações externas que poderiam influenciar o

Uniube

sistema. Durante as simulações, os dados foram registrados automaticamente em arquivos de texto no formato .txt, organizados em colunas, com cada linha representando registros temporais dos parâmetros monitorados no processo. Entre as variáveis registradas, incluem-se a temperatura de entrada e saída do fluido, o fluxo de óleo térmico e variações nos parâmetros de controle, o que proporcionou um conjunto de dados abrangente e representativo. Após a coleta, os dados foram submetidos à etapa de tratamento no ambiente de computação em nuvem Google Colab. O arquivo .txt foi carregado para o Colab, e a biblioteca Pandas foi utilizada para estruturar os dados em um dataframe, permitindo manipulação e análise eficiente. Durante o tratamento, os registros foram submetidos a uma limpeza rigorosa, com a remoção de entradas incompletas ou inconsistentes que poderiam comprometer a eficácia do modelo. Além disso, as variáveis foram padronizadas e normalizadas, garantindo que estivessem em escalas compatíveis, o que é essencial para o bom desempenho de modelos baseados em aprendizado de máquina como o ANFIS. Para assegurar a compatibilidade com ferramentas subsequentes, como o MATLAB, o dataframe final foi exportado para o formato .csv. Este formato foi escolhido devido à sua ampla aceitação e facilidade de integração com as ferramentas utilizadas para o treinamento e validação do modelo. O tratamento de dados no Colab não apenas garantiu a integridade e consistência do conjunto de dados, mas também facilitou a integração com as demais etapas do projeto. A utilização de uma plataforma baseada em nuvem permitiu processamento eficiente e colaborativo, com acesso a recursos computacionais avançados. O arquivo CSV resultante foi posteriormente importado para o MATLAB, onde foi utilizado como base para o treinamento, ajuste e validação do modelo ANFIS, representando uma transição crucial entre a preparação dos dados e o desenvolvimento do controlador inteligente.

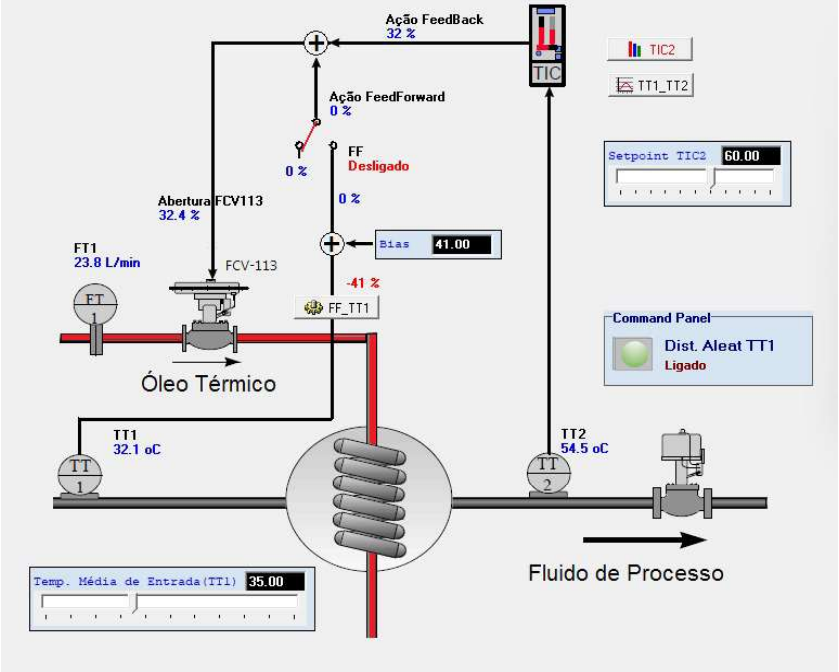
Figura 1 – Plataforma de coleta de dados



Fonte: Acervo dos autores (2024).

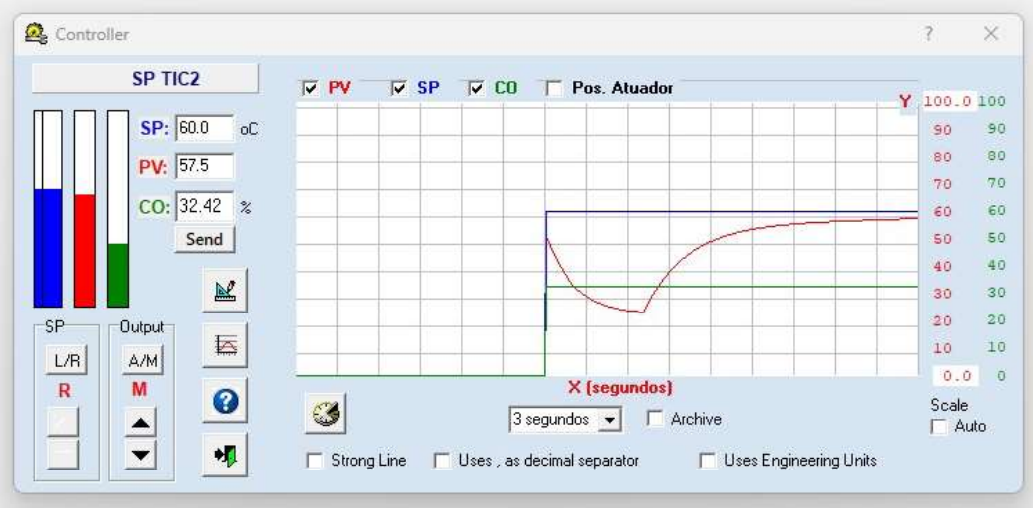
Uniube

Figura 2 – Plataforma de coleta de dados



Fonte: Acervo dos autores (2024).

Figura 3 – Controle da plataforma de coleta de dados



Fonte: Acervo dos autores (2024).

Uniube

- **Modelagem e Simulação**: A modelagem e a simulação constituem etapas fundamentais para o desenvolvimento do sistema de controle proposto. A primeira parte do processo envolve a construção de uma modelagem matemática simplificada do sistema de controle de temperatura da caldeira, levando em consideração suas principais variáveis e características dinâmicas. Este modelo foi implementado de forma a representar as interações entre o fluxo de óleo térmico, a temperatura do fluido de processo na entrada e saída, e a influência de perturbações externas, garantindo que a simulação capturasse as dinâmicas mais relevantes do sistema. Com base nessa modelagem, o MATLAB foi utilizado como ambiente principal de simulação devido à sua robustez e ferramentas especializadas para sistemas neuro-fuzzy. A ferramenta **ANFIS Editor (anfisedit)** desempenhou um papel central nesta etapa, permitindo a configuração, o treinamento e a validação do modelo neurofuzzy. Inicialmente, diferentes arquiteturas de sistema foram testadas, variando a quantidade de entradas de 3 a 5 variáveis. Essas entradas incluíram combinações como erro calculado entre o (set point - process variable), set point cujo é a temperatura desejada e variável de processo. O número de saídas foi fixado para representar o sinal de controle necessário para ajustar o fluxo de calor fornecido ao sistema, garantindo a manutenção da temperatura no setpoint desejado (49°C a 51°C). A configuração do ANFIS envolveu o ajuste das funções de pertinência para cada entrada. Foram testados diferentes tipos de funções, como triangular, trapezoidal e gaussiana, com o objetivo de identificar a configuração que melhor representasse as relações não-lineares do sistema. O número de funções de pertinência por entrada também foi ajustado durante os experimentos, variando de 2 a 5 funções por variável de entrada, o que influenciou diretamente o número de regras fuzzy geradas pelo sistema. No treinamento do modelo, diferentes métodos de aprendizado foram aplicados, incluindo o híbrido (combinação de backpropagation e método dos mínimos quadrados) e o método exclusivamente baseado em gradiente descendente. Cada abordagem foi testada com diferentes números de épocas de treinamento, variando de 10 a 1000, para avaliar o impacto na precisão do modelo. A variação no número de épocas permitiu identificar o ponto de saturação, onde o aumento das iterações não resultava em melhorias significativas no desempenho do sistema. Além disso, métricas como o erro médio absoluto (MAE) e o erro quadrático médio (RMSE) foram calculadas em cada configuração para avaliar a capacidade do modelo de generalizar os padrões do sistema e responder corretamente a condições não vistas durante o treinamento. As simulações também foram utilizadas para gerar um conjunto de dados suplementares, simulando cenários com perturbações externas e variações nos parâmetros operacionais, como mudanças na temperatura de entrada do fluido ou no fluxo de óleo térmico. Esses dados foram usados tanto para validar o modelo quanto para melhorar sua robustez, garantindo que ele fosse capaz de se adaptar a condições operacionais dinâmicas e não-lineares. Por fim, os melhores

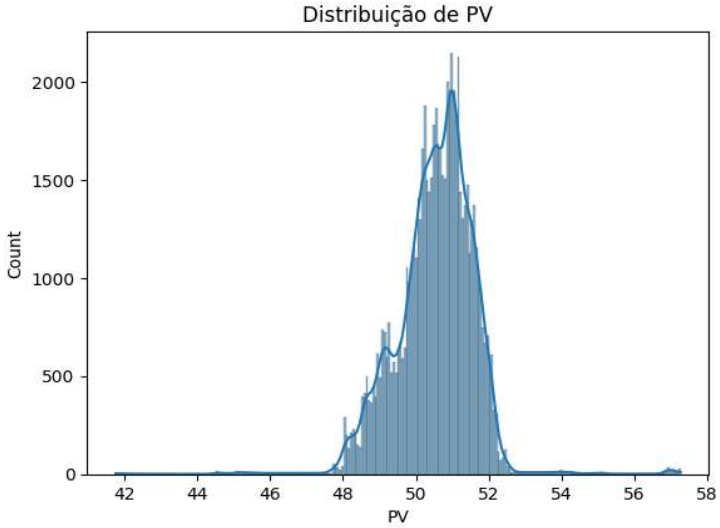
Uniube

resultados obtidos durante a simulação foram analisados em relação ao tempo de resposta do controlador e à estabilidade do sistema. Com base nas configurações otimizadas, o modelo ANFIS foi exportado para implementação prática no Arduino, garantindo que ele refletisse com precisão as dinâmicas reais do processo de controle da caldeira. Essa abordagem iterativa de modelagem, simulação e ajuste assegurou um desenvolvimento robusto e adaptado às necessidades do sistema de controle proposto, explorada para configurar as funções de pertinência fuzzy e ajustar os parâmetros da rede neural.

3. RESULTADOS E DISCUSSÃO

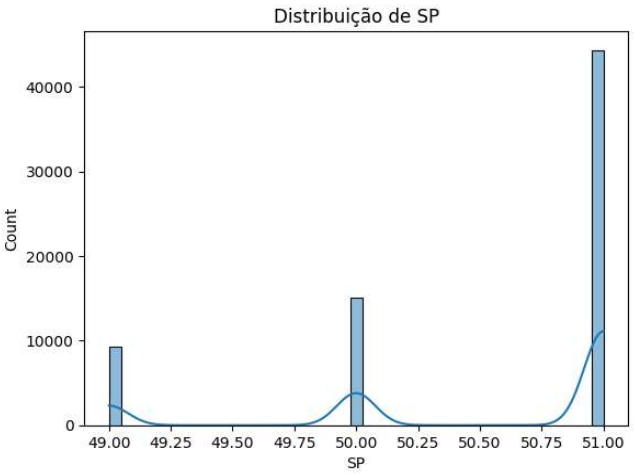
Nesta seção, apresentamos os resultados obtidos durante a etapa de tratamento de dados, que foi essencial para preparar o conjunto de informações utilizadas no treinamento do modelo ANFIS. A análise dos dados coletados no simulador e o uso de ferramentas como o Google Colab para o pré-processamento garantiram a qualidade e a representatividade das informações. A seguir, detalhamos os principais pontos observados a partir das distribuições das variáveis SP (Set Point), PV (Process Variable) e MV (Manipulated Variable), bem como os ajustes realizados no conjunto de dados.

Figura 4 – Gráfico de distribuição do PV



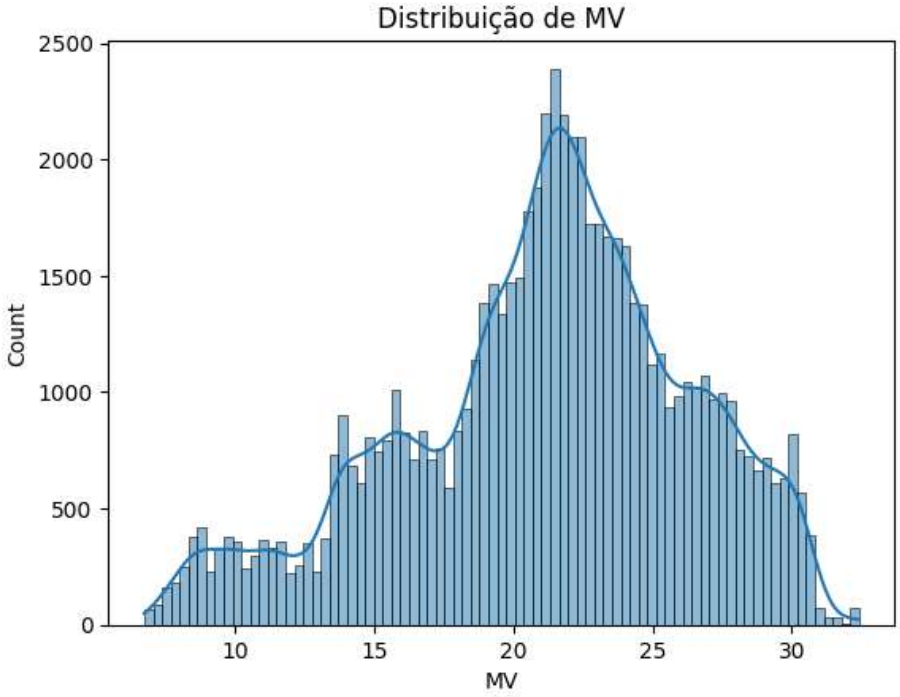
Fonte: Acervo dos autores (2024).

Figura 5 – Gráfico de distribuição do SP



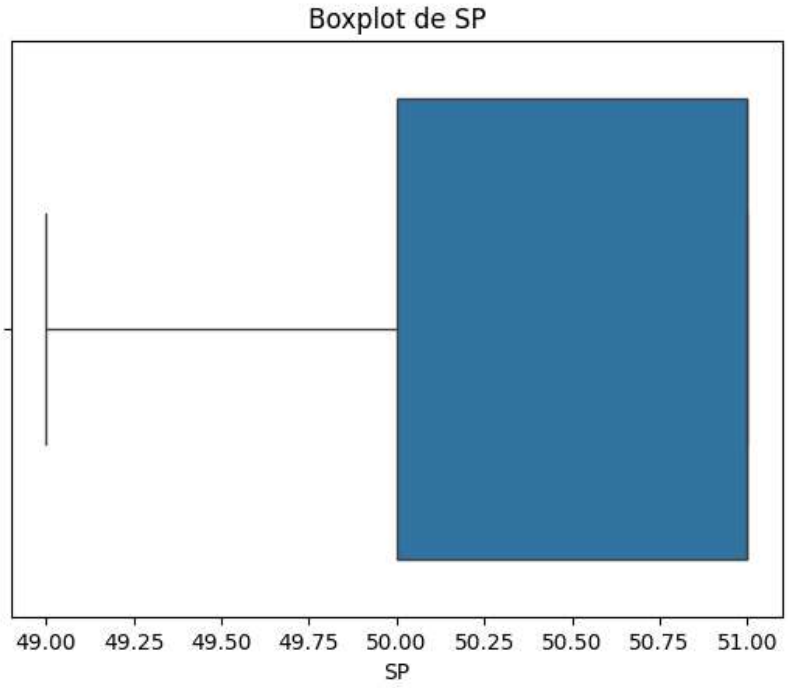
Fonte: Acervo dos autores (2024).

Figura 6 – Gráfico de distribuição do MV



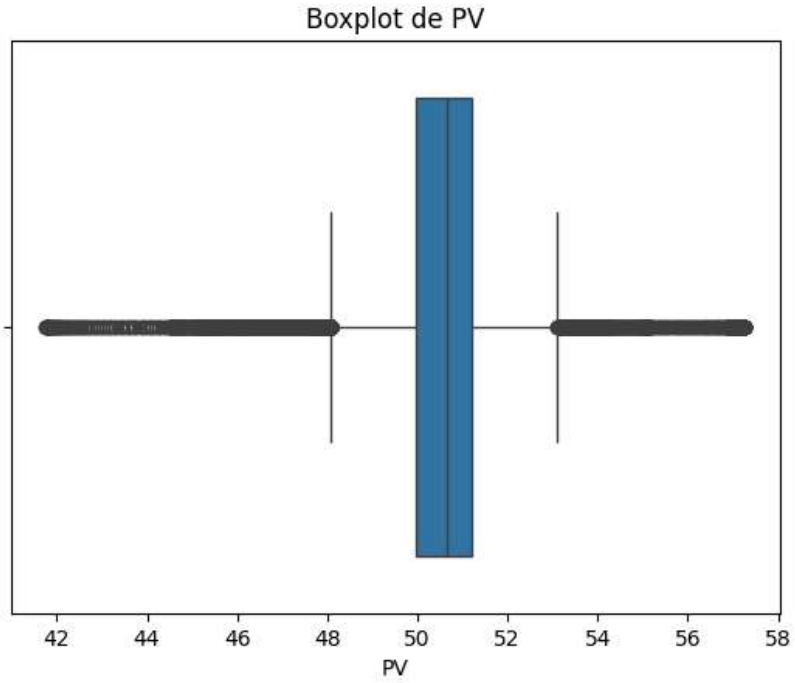
Fonte: Acervo dos autores (2024).

Figura 7 – Boxplots de distribuição do SP



Fonte: Acervo dos autores (2024).

Figura 8 – Boxplots de distribuição do PV



Fonte: Acervo dos autores (2024).

A distribuição dos dados de **SP (Set Point)** revelou que os valores estavam majoritariamente concentrados em torno de 50, com algumas variações

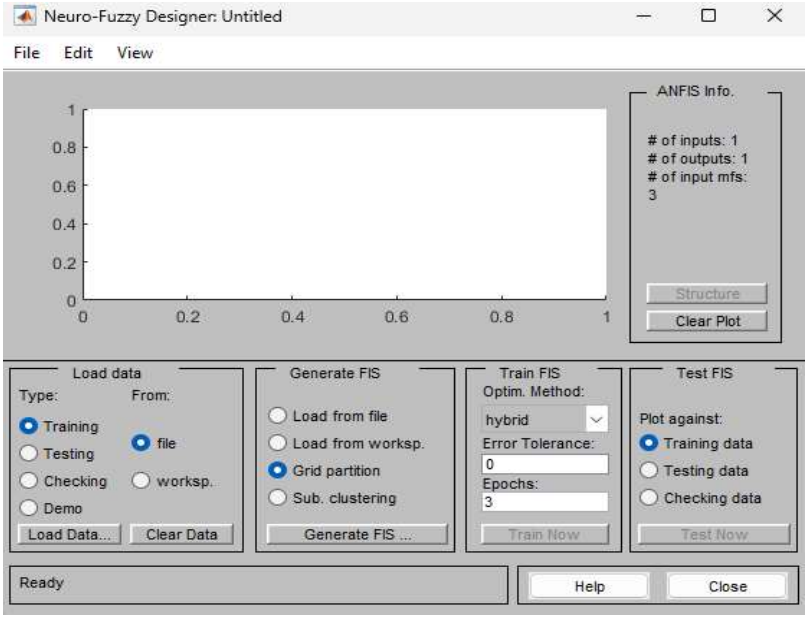
Uniube

pontuais entre 49 e 51. Essa constância no valor do setpoint é ideal para a análise, pois permitiu que ele fosse tratado como uma referência fixa para calcular o erro entre o setpoint e a variável de processo (SP - PV). Essa abordagem simplifica a identificação de desvios e facilita o ajuste do modelo durante o treinamento. Já a **distribuição de PV (Process Variable)** apresentou maior variabilidade, com valores oscilando entre 42 e 58. A densidade mais elevada foi observada no intervalo de 48 a 52, o que reflete uma boa proximidade com o setpoint. No entanto, o boxplot da variável PV revelou a presença de outliers fora do intervalo esperado, especificamente abaixo de 46 e acima de 54. Esses valores foram considerados como possíveis desvios extremos e, por isso, avaliados para remoção durante o tratamento de dados, dependendo do comportamento esperado do sistema. A variável **MV (Manipulated Variable)**, por sua vez, apresentou uma ampla distribuição com valores variando de 10 a 30. Diferentemente de PV, a distribuição de MV não mostrou discrepâncias extremas, indicando que os valores estão bem distribuídos e refletem o comportamento esperado do controlador. Isso é fundamental para garantir que o modelo ANFIS receba informações consistentes sobre o ajuste necessário para manter o sistema estável. Com base na análise inicial dos dados, algumas etapas de pré-processamento foram realizadas. Primeiramente, foi adicionada uma nova coluna para calcular o **erro (SP - PV)**, que foi utilizada como uma das entradas no treinamento do modelo ANFIS. Em seguida, os outliers identificados na variável PV foram filtrados, restringindo os valores ao intervalo de 46 a 54, com o objetivo de remover ruídos e focar nas condições esperadas do sistema. Após a limpeza, os dados ajustados foram exportados para um arquivo .csv utilizando o Google Colab, garantindo compatibilidade com o MATLAB para as etapas seguintes de modelagem e simulação. Os gráficos anexados evidenciam essas distribuições e os ajustes realizados, destacando a importância do pré-tratamento para garantir um conjunto de dados robusto e representativo. Esse processo permitiu que as entradas do modelo fossem adequadamente calibradas, otimizando o treinamento do sistema ANFIS e aumentando sua capacidade de generalização e resposta às variações operacionais. Os dados filtrados foram então utilizados para configurar e validar o modelo, marcando uma transição crucial entre as etapas de coleta e análise para o desenvolvimento do controlador inteligente. Após a etapa de tratamento de dados realizada no Google Colab, os dados foram organizados e preparados no MATLAB para posterior treinamento e validação no ANFIS Editor (**anfisedit**). O pós-tratamento foi uma etapa crucial para garantir a separação adequada dos dados em entradas e saídas, bem como para configurar corretamente o modelo neurofuzzy. Primeiramente, os dados exportados em formato .csv foram importados para o MATLAB e organizados em colunas específicas para representar as variáveis de entrada e saída. As entradas escolhidas incluíram o erro (SP - PV), a variável PV (Process Variable) e outra variável derivada, como a taxa de variação de PV, totalizando **três entradas**. A saída foi configurada como a variável **MV (Manipulated Variable)**, representando o ajuste necessário no sistema para estabilizar a temperatura no setpoint desejado. Para o treinamento do modelo, os dados foram divididos em dois subconjuntos: **70% dos dados foram utilizados para treinamento** do modelo ANFIS, enquanto

Uniube

30% foram reservados para teste. Essa separação seguiu a técnica padrão de validação cruzada, garantindo que o modelo fosse treinado com uma amostra robusta, mas também avaliado com dados independentes para verificar sua capacidade de generalização. A divisão foi feita aleatoriamente no MATLAB, utilizando comandos para embaralhar os dados antes de separá-los em conjuntos de treino e teste. Com os dados organizados, o treinamento foi realizado no **ANFIS Editor**. Após testes preliminares com diferentes números de entradas (3, 4 e 5), constatou-se que o modelo configurado com **3 entradas** apresentou o melhor desempenho. Modelos com menos entradas demonstraram maior eficiência devido à sua **menor complexidade**, o que reduziu o risco de overfitting e aumentou a capacidade de ajuste aos padrões presentes nos dados. Em contrapartida, o aumento no número de entradas (para 4 ou 5) elevou a dimensionalidade do problema, tornando o modelo mais suscetível a introduzir redundâncias ou ruídos, além de exigir um volume maior de dados para evitar perdas de desempenho. A configuração com três entradas foi, portanto, a mais eficiente, capturando os padrões essenciais sem comprometer a simplicidade. Além disso, os testes indicaram que modelos com menos saídas tendem a apresentar melhores resultados, pois a redução da dimensionalidade das saídas minimiza a necessidade de ajustes complexos e evita a introdução de inconsistências nos dados. Esse comportamento foi particularmente evidente neste caso, onde a saída foi mantida como única (MV), alinhando-se ao objetivo do controlador de ajustar o sistema com precisão e eficiência.

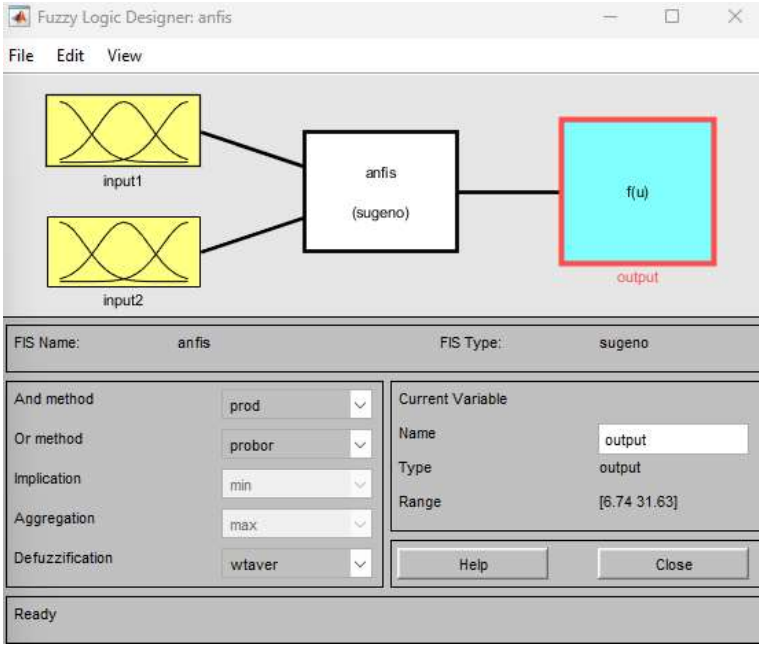
Figura 9 – Simulador do anfis (ANFISEDIT)



Fonte: Acervo dos autores (2024).

Uniube

Figura 10 – Regras do modelo Anfis

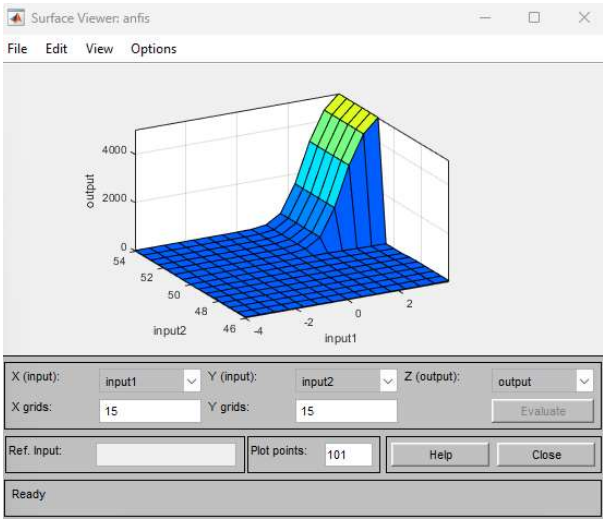


Fonte: Acervo dos autores (2024).

Na configuração das funções de pertinência no **anfisedit**, o modelo **gaussiano** foi identificado como o mais preciso, superando outras opções como as funções trapezoidais e triangulares. A suavidade da função gaussiana permitiu uma transição mais gradual entre os estados fuzzy, melhorando a precisão do modelo ao lidar com as variações contínuas do sistema. O treinamento do modelo foi realizado utilizando o **algoritmo híbrido**, que combina o método de backpropagation para ajustar os parâmetros das funções de pertinência com o método dos mínimos quadrados para otimizar os coeficientes das regras fuzzy. Para garantir uma convergência adequada, foram configuradas **1.000 épocas** no treinamento, resultando em um modelo altamente ajustado às dinâmicas do sistema.

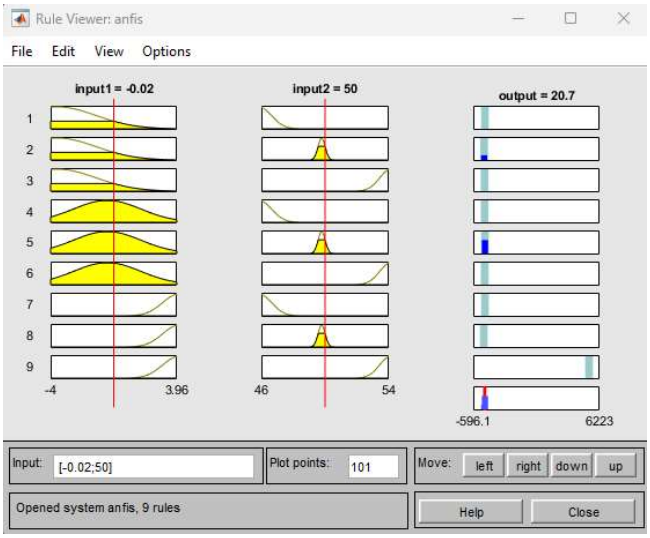
Uniube

Figura 11 – Gráfico representando a relação entrada x saída



Fonte: Acervo dos autores (2024).

Figura 12 – Gráfico representando regras do modelo

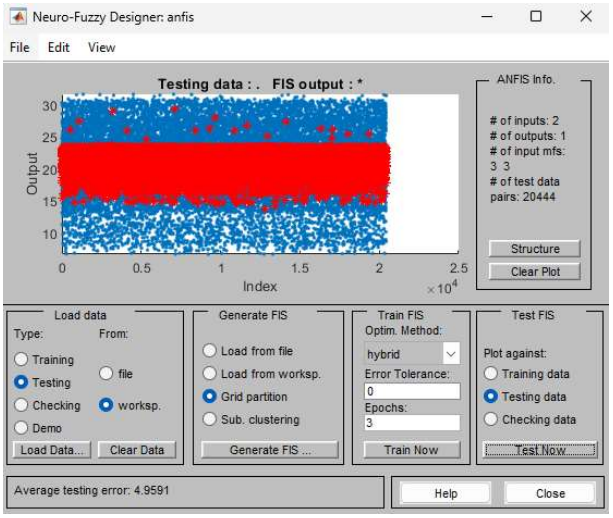


Fonte: Acervo dos autores (2024).

Os resultados preliminares das simulações com o modelo treinado mostraram que a configuração escolhida proporcionou uma excelente precisão na previsão dos ajustes necessários no sistema. O modelo final apresentou uma alta capacidade de generalização, respondendo de maneira eficaz às condições de teste e demonstrando ser uma solução robusta e eficiente para o controle de temperatura da caldeira.

Uniube

Figura 13 – Treinamento do modelo



Fonte: Acervo dos autores (2024).

Os resultados preliminares das simulações com o modelo treinado mostraram que a configuração escolhida proporcionou uma excelente precisão na previsão dos ajustes necessários no sistema. O modelo final apresentou uma alta capacidade de generalização, respondendo de maneira eficaz às condições de teste e demonstrando ser uma solução robusta e eficiente para o controle de temperatura da caldeira. **Figura 10 -** Representa o modelo treinado O treinamento do modelo ANFIS foi uma das etapas mais importantes no desenvolvimento do sistema de controle de temperatura, e sua execução foi realizada no MATLAB utilizando a ferramenta **ANFIS Editor (anfisedit)**. Após o pré-tratamento dos dados no Google Colab e a exportação em formato `.csv`, o conjunto de dados foi organizado no MATLAB, separando-se as colunas de entrada e saída. As entradas foram definidas como a variável **PV (Process Variable)** e o **Erro (SP - PV)**, enquanto a saída foi configurada como a variável manipulada **MV (Manipulated Variable)**, responsável por ajustar o fornecimento de energia ao sistema. Essa organização garantiu que os dados fossem coerentes e compatíveis com a estrutura requerida pelo ANFIS, estabelecendo a base para o treinamento. Os dados foram então divididos em dois subconjuntos, com **70% para o treinamento** e **30% para o teste**, seguindo a técnica padrão de validação cruzada. Essa separação foi realizada de forma aleatória para garantir que os dados fossem distribuídos de maneira equilibrada entre os dois conjuntos, evitando vieses que pudessem comprometer a avaliação do modelo. O conjunto de treinamento foi utilizado para ajustar os parâmetros do modelo, enquanto o conjunto de teste foi reservado para validar sua capacidade de generalização. Essa abordagem é fundamental para garantir que o modelo não apenas se ajuste aos dados vistos durante o treinamento, mas também consiga prever com precisão o comportamento do sistema em condições desconhecidas. O processo de treinamento foi realizado no ANFIS Editor, com uma configuração inicial que utilizava **3 funções de pertinência por entrada**. Diferentes tipos de funções foram testados, incluindo triangular, trapezoidal e

Uniube

gaussiana. Após várias iterações, constatou-se que o modelo utilizando funções **gaussianas** apresentou o melhor desempenho. Esse tipo de função oferece uma transição suave entre os estados fuzzy, permitindo capturar com maior precisão as variações contínuas do sistema, especialmente em condições dinâmicas e com alta variabilidade dos dados. A suavidade da função gaussiana foi essencial para reduzir o erro durante o treinamento, além de melhorar a estabilidade do modelo. Outro aspecto crítico foi a escolha do **método híbrido de treinamento**, que combina **backpropagation** para ajustar os parâmetros das funções de pertinência com o **método dos mínimos quadrados** para otimizar os coeficientes das regras fuzzy. Essa combinação garantiu que o modelo fosse treinado de maneira eficiente, minimizando o erro residual e ajustando as regras fuzzy de forma consistente com os padrões presentes nos dados. O modelo foi configurado para realizar **1.000 épocas de treinamento**, garantindo uma convergência adequada dos parâmetros. O número elevado de épocas foi necessário para ajustar de forma precisa as funções de pertinência e as regras fuzzy, proporcionando ao modelo a capacidade de aprender as relações complexas entre as variáveis de entrada e saída. Durante o treinamento, diferentes configurações de entradas foram testadas, variando entre 3, 4 e 5 variáveis de entrada. Os resultados indicaram que o modelo com **3 entradas** apresentou o melhor desempenho. Modelos com maior número de entradas, como 4 ou 5, aumentaram significativamente a complexidade do sistema, resultando em maior risco de **overfitting** e menor capacidade de generalização. Além disso, o aumento no número de entradas introduziu redundâncias nos dados, tornando o treinamento mais sensível a ruídos e exigindo maior volume de dados para manter a estabilidade. Por outro lado, o modelo com 3 entradas conseguiu capturar as principais dinâmicas do sistema sem sacrificar a simplicidade, demonstrando que uma abordagem mais enxuta e bem calibrada pode oferecer resultados superiores em sistemas com alta correlação entre variáveis. Os resultados do treinamento indicaram que o modelo ajustado apresentou uma excelente correspondência entre as entradas e a saída, com um **erro médio durante o teste de 4,9591**, conforme reportado no ANFIS Editor. Essa métrica evidencia a capacidade do modelo de prever os ajustes necessários no sistema com alta precisão, mesmo em condições não vistas durante o treinamento. A correspondência visual entre os dados de teste e as previsões do modelo, mostrada no gráfico do ANFIS Editor, reforça essa conclusão, com os pontos azuis (dados de teste) e vermelhos (previsões do modelo) alinhados de forma consistente. Assim, o treinamento no MATLAB, utilizando o ANFIS Editor, foi concluído com sucesso, resultando em um modelo robusto, eficiente e com alta capacidade de generalização. A escolha cuidadosa das configurações do modelo, como o número de entradas, tipo de funções de pertinência, método de treinamento e número de épocas, foi fundamental para garantir o desempenho final. Com esse modelo treinado, o sistema está preparado para ser implementado na plataforma embarcada, validando sua eficácia em um ambiente prático e dinâmico. Com o treinamento do modelo ANFIS concluído e os parâmetros otimizados no MATLAB, o sistema está agora pronto para ser implementado em uma plataforma embarcada. O próximo passo será realizado por outra equipe, que dará continuidade ao projeto ao integrar o

Uniube

modelo treinado em um microcontrolador **Arduino**, responsável por gerenciar as leituras em tempo real e aplicar os ajustes necessários no sistema de controle. Essa implementação prática permitirá validar o desempenho do modelo em condições reais, testando sua capacidade de adaptar-se dinamicamente às variações do processo e garantir o controle preciso da temperatura na caldeira. A transição para a plataforma embarcada representa uma etapa crucial para transformar o trabalho desenvolvido em uma solução funcional e aplicável no ambiente industrial.

4. CONCLUSÕES

O desenvolvimento do controlador neuro-fuzzy para o sistema de controle de temperatura de uma caldeira demonstrou ser uma solução eficaz e adaptativa, atendendo aos objetivos de estabilidade e eficiência estabelecidos no início do projeto. A combinação de redes neurais e lógica fuzzy no modelo ANFIS provou ser uma abordagem poderosa para lidar com a não-linearidade e a variabilidade inerente a sistemas industriais. Durante as etapas de simulação e treinamento, o controlador foi capaz de aprender os padrões do sistema e responder de forma precisa e rápida às variações nas condições operacionais, garantindo um controle estável mesmo sob flutuações externas. A análise comparativa entre diferentes configurações de entradas e saídas reforçou a importância de um design equilibrado, com o modelo configurado para três entradas e uma saída apresentando o melhor desempenho. Esse design mais enxuto reduziu a complexidade do sistema, minimiza riscos de overfitting e otimizou o uso dos dados disponíveis, comprovando que soluções simples e bem estruturadas podem ser mais eficientes em processos industriais. O uso de funções de pertinência gaussianas, aliado ao treinamento híbrido com 1.000 épocas, foi essencial para alcançar um ajuste preciso e confiável do modelo. O tratamento prévio dos dados no Google Colab, seguido pelo treinamento no MATLAB, garantiu um fluxo de trabalho eficiente e integrado. As ferramentas utilizadas permitiram a organização, limpeza e normalização dos dados, enquanto o ANFIS Editor facilitou o ajuste fino dos parâmetros do modelo. O erro médio obtido nos testes validou a eficácia do treinamento, evidenciando que o controlador é capaz de generalizar seu aprendizado e operar com alto nível de precisão. A implementação prática do controlador em uma plataforma embarcada Arduino será realizada por uma equipe sucessora, marcando o próximo estágio do projeto. A escolha do Arduino oferece uma solução viável e escalável, permitindo a integração do modelo com sensores e atuadores para validar sua aplicação em tempo real. Essa etapa será crucial para avaliar o desempenho do controlador em cenários reais, testando sua capacidade de adaptar-se dinamicamente e manter o controle eficiente da temperatura. Como resultado, este estudo contribui não apenas para a aplicação prática de sistemas neuro-fuzzy, mas também para o avanço das tecnologias de automação industrial. A implementação de um controlador adaptativo como este demonstra ser uma alternativa viável para reduzir intervenções manuais, otimizar o consumo energético e melhorar a estabilidade dos processos industriais. Além

Uniube

disso, os resultados obtidos neste trabalho reforçam a flexibilidade e a escalabilidade dessa abordagem, abrindo possibilidades para sua aplicação em uma variedade de sistemas industriais além das caldeiras de destilação. Em conclusão, o trabalho atingiu seus objetivos ao desenvolver e validar um controlador neuro-fuzzy eficiente, mostrando que a integração de inteligência artificial com controle automático tem grande potencial para transformar processos industriais. Essa tecnologia oferece não apenas ganhos em eficiência e sustentabilidade, mas também uma base para inovações futuras na automação de sistemas complexos. Complementando a conclusão, podemos observar que o sucesso na implementação do controlador neuro-fuzzy para o sistema de controle de temperatura da caldeira reforça a viabilidade e o potencial das abordagens híbridas em processos industriais. A combinação de inteligência artificial e técnicas de controle adaptativo proporciona uma solução robusta que não só melhora a performance do sistema, mas também oferece vantagens em termos de flexibilidade e adaptabilidade a condições operacionais variáveis. Além disso, o processo de validação realizado com dados reais e a comparação entre diferentes configurações de entradas e saídas demonstram a importância de um design otimizado e equilibrado. O desempenho alcançado pelo modelo, tanto nas simulações quanto no treinamento, confirma que a inteligência artificial pode efetivamente contribuir para a melhoria de processos industriais, reduzindo a dependência de intervenção humana e aumentando a precisão no controle de variáveis cruciais como a temperatura. A implementação do controlador no Arduino abre novas possibilidades de aplicação para essa tecnologia em outros setores industriais, principalmente aqueles que exigem controle preciso e adaptável, como os setores químico, petroquímico e alimentício. A escalabilidade do sistema, aliado à sua flexibilidade, permite que ele seja facilmente adaptado a diferentes processos e necessidades, representando um grande avanço na automação industrial. O próximo passo da implementação prática será essencial para testar a eficácia do controlador em condições reais de operação. A integração do modelo neuro-fuzzy com sensores e atuadores no Arduino possibilitará a validação do modelo, além de possibilitar ajustes finos que podem otimizar ainda mais a eficiência do sistema. A experiência adquirida com a implementação prática fornecerá insights valiosos para futuras melhorias e adaptações do sistema, ampliando seu potencial de aplicação. Por fim, este estudo demonstra que as tecnologias baseadas em inteligência artificial, como o controlador neuro-fuzzy, estão posicionadas como soluções inovadoras e altamente eficientes para a automação de processos industriais complexos. Com os resultados obtidos, o trabalho não só contribui para o avanço da automação, mas também abre caminho para novas pesquisas que possam explorar e expandir ainda mais as capacidades dessa abordagem híbrida em sistemas industriais de grande porte.

Uniube

REFERÊNCIAS

ZIMMERMANN, H. J. *Fuzzy set theory and its applications*. Boston: Kluwer Academic Publishers, 2001.

JANG, J. S. R.; SUN, C. T.; MIZUTANI, E. *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Englewood Cliffs: Prentice Hall, 1997.

BEALE, M. H.; HAGAN, M. T.; DEMUTH, H. B. *Neural network toolbox user's guide*. Natick: MathWorks, 2019.

PASSINO, K. M.; YURKOVICH, S. *Fuzzy control*. Menlo Park: Addison-Wesley Longman, 1998.