

**Emulação de Uma Lei de Controle Preditivo via Arquiteturas
de Redes Neurais**

Emanuel Benício de Almeida Cajueiro

Trabalho de Conclusão de Curso

MBA em Inteligência Artificial e Big Data

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Emulação de uma Lei de Controle
Preditivo via Arquiteturas de Redes
Neurais

Emanuel Benício de Almeida Cajueiro

Emanuel Benício de Almeida Cajueiro

Emulação de uma Lei de Controle Preditivo via Arquiteturas de Redes Neurais

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Dr. João Paulo Papa

Dr. Márcio André Fernandes Martins

Dr. Raony Maia Fontes

USP - São Carlos

2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

B467e Benício de Almeida Cajueiro, Emanuel
Emulação de uma Lei de Controle Preditivo via
Arquiteturas de Redes Neurais / Emanuel Benício de
Almeida Cajueiro; orientador João Paulo Papa;
coorientador Márcio André Fernandes Martins. -- São
Carlos, 2024.
57 p.

Trabalho de conclusão de curso (MBA em
Inteligência Artificial e Big Data) -- Instituto de
Ciências Matemáticas e de Computação, Universidade
de São Paulo, 2024.

1. controle preditivo baseado em modelo. 2.
redes neurais artificiais. 3. torre de destilação.
4. eficiência computacional. I. Papa, João Paulo,
orient. II. André Fernandes Martins, Márcio,
coorient. III. Título.

DEDICATÓRIA

*A Deus, à Soane, minha esposa, e ao
Eli, meu filho, pelo apoio, paciência
e amor.*

AGRADECIMENTOS

Agradeço a Deus, por Sua infinita misericórdia, pois até aqui Ele tem me ajudado.

Ao Prof. João Paulo Papa por aceitar ser meu orientador e por todo o auxílio prestado.

Aos Profs. Márcio Martins e Raony Fontes, meus coorientadores, que, com humildade, paciência e dedicação, têm me ajudado de modo inestimável.

À Profa. Solange, por toda a sua dedicação a este curso de especialização.

À Soane, minha esposa, e ao Eli, meu filho, pelo apoio e paciência.

Aos meus pais, Edvaldo Benício e Reinalice, por sempre orarem e desejarem o melhor pra mim.

Aos meus irmãos, Nazareno, Agno, Sanmaria e Santana, por toda a torcida e apoio.

RESUMO

BENÍCIO DE ALMEIDA CAJUEIRO, E. **Emulação de uma Lei de Controle Preditivo via Arquiteturas de Redes Neurais.** 2024. 56 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Neste trabalho, uma metodologia foi desenvolvida com o objetivo de substituir a lei de controle do controlador preditivo baseado em modelo (MPC) linear incremental por uma estrutura de inteligência artificial, mais especificamente, uma rede neural *feedforward*. A hipótese foi que redes neurais artificiais (RNAs) poderiam substituir o MPC em questão, proporcionando respostas similares nas saídas da planta escolhida, além de oferecer uma solução mais eficiente em termos de custo computacional, permitindo, em aplicações futuras, a implementação do controlador em sistemas embarcados com recursos computacionais limitados, ao mesmo tempo mantendo ou até melhorando a estabilidade do sistema devido à otimização do tempo de resposta do controlador. Para validar essa abordagem, foram utilizados dados simulados de um subsistema de uma torre de destilação, onde as entradas do sistema eram a taxa de fluxo de vapor no refeedor (u_1 , em ton/h) e a taxa de fluxo de refluxo (u_2 , em m³/d), enquanto as saídas monitoradas foram o nível de líquido no vaso superior (y_1 , em %) e a temperatura no prato 68 (y_2 , em °C). Duas simulações foram realizadas: a primeira considerou a inserção de perturbações não medidas sem variação no setpoint, enquanto a segunda incluiu tanto perturbações não medidas quanto a variação do setpoint. Em ambas as simulações, os resultados foram satisfatórios. A rede neural foi treinada com base no sinal de erro ao longo do horizonte de predição, sendo que a saída da rede representava o incremento de controle. Os resultados demonstraram que a rede neural foi capaz de emular de forma eficaz o comportamento do MPC, proporcionando respostas rápidas e estáveis, similares às encontradas com o uso do MPC, mesmo na presença de perturbações e mudanças no setpoint. Além disso, a redução no custo computacional indicou que a abordagem proposta pode ser viável para controle em sistemas embarcados com recursos computacionais limitados, oferecendo um desempenho equivalente ao MPC convencional.

Palavras-chave: controle preditivo baseado em modelo (MPC); redes neurais artificiais; torre de destilação; eficiência computacional.

ABSTRACT

BENÍCIO DE ALMEIDA CAJUEIRO, E. **Emulation of a Predictive Control Law via Neural Network Architectures.** 2024. 56 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

In this work, a methodology was developed with the objective of replacing the control law of the linear incremental model predictive controller (MPC) with an artificial intelligence structure, more specifically a feedforward neural network. The hypothesis was that artificial neural networks (ANNs) could replace the MPC in question, providing similar responses in the outputs of the chosen plant, in addition to offering a more efficient solution in terms of computational cost, enabling, in future applications, the implementation of the controller in embedded systems with limited computational resources, while maintaining or even improving system stability due to the optimization of the controller's response time. To validate this approach, simulated data from a subsystem of a distillation column were used, where the system inputs were the vapor flow rate in the reboiler (u_1 , in ton/h) and the reflux flow rate (u_2 , in m³/d), while the monitored outputs were the liquid level in the top vessel (y_1 , in %) and the temperature at tray 68 (y_2 , in °C). Two simulations were performed: the first considered the insertion of unmeasured disturbances without setpoint variation, while the second included both unmeasured disturbances and setpoint variation. In both simulations, the results were satisfactory. The neural network was trained based on the error signal over the prediction horizon, with the network's output representing the control increment. The results showed that the neural network was able to effectively emulate the behavior of the MPC, providing fast and stable responses, similar to those obtained using the MPC, even in the presence of disturbances and setpoint changes. Additionally, the reduction in computational cost indicated that the proposed approach may be viable for control in embedded systems with limited computational resources, offering performance equivalent to that of the conventional MPC.

Keywords: model predictive control (MPC); artificial neural networks; distillation column; computational efficiency.

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	Hipótese de Pesquisa	16
1.2	Estrutura do Trabalho	17
2	REVISÃO BIBLIOGRÁFICA	18
3	FUNDAMENTAÇÃO TEÓRICA	23
3.1	Controle Preditivo Baseado em Modelo (MPC).....	23
3.1.1	Modelo em Espaço de Estado Aumentado	26
3.1.2	Controle Preditivo de Horizonte Recuado	27
3.2	Rede Neural <i>Feedforward</i>	28
3.3	Função de Correlação Cruzada.....	30
3.4	Teste	31
4	METODOLOGIA.....	32
4.1	Simulação da Planta com o MPC	32
4.2	Armazenamento e Pré-Processamento dos Dados.....	38
4.3	Simulação com Modelo de Rede Neural	43
5	RESULTADOS E DISCUSSÃO	45
5.1	Emulação do MPC linear incremental via RNA <i>Feedforward</i>	45
6	CONCLUSÃO.....	54
	REFERÊNCIAS	55

1 INTRODUÇÃO

O controle preditivo baseado em modelo (do inglês, *Model Predictive Control* - MPC) envolve a solução de problemas de otimização a cada período de amostragem. Neste intervalo de tempo se faz necessário gerar uma sequência de ações de controle com base nas informações mais recentes disponíveis (Camacho e Bordons, 1995). Essa necessidade de otimização *online*, historicamente, tem implicado para a indústria a adição de mais um nível na pirâmide de automação industrial devido ao elevado esforço computacional neste tipo de tarefa (de Prada, 2021).

Na camada de otimização da pirâmide de automação é comum o uso de hardware especializado (de Prada, 2021), isso porque atrasos significativos na resolução da otimização podem impactar negativamente o desempenho do MPC, uma vez que o controle pode não ser ajustado em tempo hábil para mudanças rápidas no sistema (Richalet e O'donovan, 2009). Além disso, muitos algoritmos de otimização utilizados no MPC são iterativos, o que pode exigir estratégias eficientes para inicialização e convergência.

Para que a otimização em tempo real seja possível, deseja-se que as leis de MPC, resolvidas por meio de técnicas de inteligência artificial (IA), alcancem melhor desempenho do que por métodos convencionais de solução numérica (como exemplo, a programação quadrática). Por isso, a justificativa para o desenvolvimento do presente trabalho consiste na necessidade, ainda corrente, de se investigar métodos de sintetização de leis de MPC, para que a indústria possa implementá-los em controladores industriais (ex.: o controlador lógico programável) e sistemas embarcados, os quais, muitas vezes, possuem menor poder computacional.

Trabalhos têm sido desenvolvidos buscando uma implementação eficiente de métodos de MPC em sistemas embarcados. Santana et al. (2023) propõem um método prático e bem-sucedido para embarcar duas diferentes estratégias de MPC: o MPC de Horizonte Infinito Robusto (do inglês, *Robust Infinite Horizon MPC* - RIHMPC) e o Controle Preditivo Não Linear (do inglês, *Nonlinear Model Predictive Control* - NMPC), em um sistema microcontrolado.

O objetivo deste trabalho é sintetizar estruturas de inteligência artificial (IA) com o propósito de emular uma lei de controle preditivo baseado em modelo (MPC), mais especificamente, o MPC linear na forma incremental. Como aplicação para extração dos sinais que serão usados no treinamento de uma Rede Neural Artificial (RNA), serão considerados dois

subsistemas de uma torre de destilação. Os subsistemas da referida torre de destilação são descritos no artigo de Alvarez *et al.* (2009).

As torres de destilação são de grande importância na indústria química e petroquímica, pois desempenham um papel de separação de misturas líquidas em seus componentes individuais. Esse processo é útil para a produção de substâncias puras e para a obtenção de produtos específicos. A destilação é amplamente utilizada para separar hidrocarbonetos no refino de petróleo, bem como na produção de solventes, produtos farmacêuticos e bebidas alcoólicas. As torres operam com base na diferença de volatilidade entre os componentes de uma mistura, utilizando o aquecimento e a condensação seletiva para fracionar as substâncias (Luyben, 2013).

Além de sua relevância para a separação de produtos, a eficiência das torres de destilação afeta diretamente a rentabilidade e a sustentabilidade dos processos industriais. Projetos otimizados de torres, que maximizam a recuperação de produtos e minimizam o consumo de energia, são essenciais para a redução dos custos operacionais. Ademais, a automação e o controle ótimo das malhas de controle da torre garantem que o processo de destilação ocorra de forma eficiente, segura e em conformidade com as normas ambientais (Luyben, 2013).

Os resultados esperados tendo como aplicação a torre de destilação podem ser expandidos para as demais áreas que envolvem o controle automático de sistemas dinâmicos lineares.

1.1 Hipótese de Pesquisa

A hipótese deste estudo é que o controle preditivo baseado em RNA pode ser uma alternativa mais eficaz, do ponto de vista prático, em comparação ao MPC linear na forma incremental para o controle de processos em tempo real. Ao utilizar RNAs para substituir a resolução de problemas de otimização em tempo real, além de a RNA conseguir fazer o papel do MPC em questão no controle ótimo de processos, obtém-se uma redução significativa no custo computacional. Isso pode permitir a implementação desse controlador em computadores industriais, como CLPs, instrumentos inteligentes e sistemas embarcados mais acessíveis, mantendo ou até melhorando a estabilidade do sistema devido à otimização do tempo de resposta do controlador.

1.2 Estrutura do Trabalho

No capítulo 2 é realizada uma breve revisão bibliográfica sobre métodos de inteligência artificial para emulação de MPC tradicionais e sua aplicação em subsistemas de uma torre de destilação. Em seguida, no capítulo 3, é abordada a fundamentação teórica, a qual trata brevemente sobre o MPC, RNA feedforward, função de correlação cruzada e a validação via equação do ajuste. A metodologia necessária para substituição de uma lei de MPC via RNA é tratada no capítulo 4. No capítulo 5 são apresentados os resultados. Por fim, no capítulo 6, conclui-se o trabalho com uma avaliação crítica dos resultados e sugestões para futuras pesquisas.

2 REVISÃO BIBLIOGRÁFICA

Implementar o MPC em sistemas com dinâmica rápida ainda é desafiador, e essa dificuldade aumenta se o modelo do processo for não-linear (Kather *et al.*, 2023; Chan *et al.*, 2021). Sistemas não-lineares acrescentam complexidade ao problema de otimização do MPC, pois requerem algoritmos de otimização mais sofisticados que são computacionalmente mais intensos. Como o tempo de cálculo do otimizador já é limitado pela rapidez da dinâmica do sistema, pode não ser viável encontrar a solução ótima da lei de MPC dentro do tempo disponível, exigindo abordagens de otimização mais rápidas, simplificações do modelo ou abordagens que envolvam a sintetização de uma IA capaz de reproduzir o comportamento de um MPC.

As RNAs são consideradas aproximadores universais devido a capacidade de, partindo de uma dada arquitetura de rede previamente selecionada, aprenderem diretamente a partir dos sinais de entrada e saída de sistemas complexos e com forte não-linearidades (Haykin, 2009). De acordo com Hecht-Nielsen (1989), uma RNA *feedforward* de três camadas, treinada com algoritmo *backpropagation*, é teoricamente capaz de modelar qualquer função contínua que possua energia limitada (funções no espaço L_2).

Li *et al.* (1991) propuseram um algoritmo de RNAs para uma lei de MPC. Neste trabalho, as RNAs desempenham um papel duplo no controle preditivo. Enquanto uma RNA atua como um preditor de um passo à frente, prevendo a próxima saída da planta com base na informação atual e passada. A outra RNA atua como um controlador preditivo, que utiliza as previsões do preditor para tomar decisões de controle no sistema. Eles usaram uma RNA *feedforward* com múltiplas camadas, com função de ativação sigmoide, e a treinaram via algoritmo *backpropagation* (Haykin, 2009). De acordo com os autores, os resultados obtidos mostram que o sistema de controle baseado em RNAs consegue alcançar um desempenho superior em comparação com métodos tradicionais de MPC, especialmente em sistemas com dinâmicas complexas e não-lineares.

Wang e Boyd (2010) abordam métodos *online* para reduzir o tempo de cálculo do MPC para sistemas com dinâmica rápida. Os autores ressaltam que uma forma comum de lidar com o MPC para sistemas de dinâmica rápida é calcular a solução da programação quadrática (QP) de forma explícita, porém *offline*. Nesse caso, implementa-se a ação de controle *online*, na forma de uma tabela de pesquisa, o que torna este método conveniente para sistemas em que o número de estados considerados não é maior do que cerca de cinco. Isso ocorre, pois o número

de entradas na tabela tende a crescer exponencialmente com o incremento do número de estados, restrições, entradas e horizonte de predição.

A estratégia adotada por Wang e Boyd (2010) consiste em explorar a estrutura da QP reordenando-a de modo que um problema com $O(T^3(n + m)^3)$ operações por passo, via método de ponto interior, se torne $O(T(n + m)^3)$. Em que n é a dimensão do estado, m é a dimensão da entrada e T é o horizonte de controle. Assim, o artigo em questão faz uma análise detalhada de um conjunto de métodos de otimização *online* que exploram a estrutura específica do problema de QP na solução de MPC para acelerar o cálculo da ação de controle. Estes métodos são variações de um método chamado “Método de Barreira Primal”.

De acordo com Wang e Boyd (2010), ajustes (variações) nesses métodos permitem a execução do MPC com esforço computacional significativamente menor (na ordem de 100 vezes mais rápido) em comparação com a solução via otimizadores genéricos. Os autores destacam a necessidade de análises formais de garantia da estabilidade para as soluções propostas. Os autores disponibilizam um código escrito em linguagem C (ver http://www.stanford.edu/~boyd/fast_mpc.html) para *download* da implementação do método desenvolvido. A dificuldade da implementação em microcontrolador, do método tratado por estes autores, reside na necessidade ainda presente do uso de um algoritmo de otimização para solução da lei de controle.

Williams *et al.* (2017) apresentaram um algoritmo de MPC para modelos baseados em RL capaz de lidar com dinâmicas complexas e não-lineares. RNAs multicamadas são usadas como modelos de dinâmica no algoritmo MPC, facilitando a resolução de tarefas de RL baseada em modelo. Os autores implementaram o método em um modelo simulado de um pêndulo invertido, além de um sistema físico (um veículo de rally, para controle de trajetória). Eles ressaltam que o método desenvolvido demanda alto custo computacional, devido a arquitetura da RNA (número de parâmetros e funções de ativação não-lineares), para que seja simulado em tempo real.

Tange *et al.* (2019) investigaram a integração do MPC com técnicas de aprendizado por reforço profundo (ARP) para melhorar o desempenho de sistemas de controle lineares e invariantes no tempo sujeitos a perturbação. Esta abordagem utiliza um modelo linear pré-identificado para prever erros de rastreamento futuros, que são então utilizados como estado observado por um compensador ARP. O artigo foca em casos em que, embora o modelo em espaço de estados seja contínuo, o sinal de controle só pode assumir níveis predeterminados, como -1, 0 ou 1 (controle multinível). Através de exemplos numéricos, de acordo com os autores, esta metodologia supera métodos de controle baseados em aprendizagem por reforço

que utilizam o controle proporcional (P) e proporcional-integral (PI) em termos de desempenho de controle e convergência.

Nos métodos tradicionais de MPC cada decisão de controle requer a resolução de um problema de otimização que minimiza uma função de custo prevista ao longo de um horizonte de tempo futuro, sujeita a restrições operacionais. Essa otimização é computacionalmente intensiva e precisa ser realizada em tempo real à medida que novos dados se tornam disponíveis. No entanto, o método proposto por Tange *et al.* (2019) utiliza um compensador (baseado em DRL) que já "aprendeu" a melhor política de controle durante a fase de treinamento. Portanto, em vez de resolver um problema de otimização a cada período de amostragem, o sistema consulta a política aprendida para determinar a ação de controle. Este método é menos oneroso do ponto de vista computacional.

Ressalta-se que no estudo de Tange *et al.* (2019), a aplicabilidade do sistema proposto foi demonstrada por meio de simulações realizadas no software Matlab, adotando-se um período de amostragem de um segundo. É importante notar que a implementação em sistemas baseados em microprocessadores não foi abordada.

Cotrufo *et al.* (2020) aliaram RNAs Profundas com técnicas com o MPC. Os autores usaram estas redes para predição do comportamento energético em edifícios (temperatura ao longo de 24 horas e demanda energética) que são aquecidos com mais de uma fonte de energia (exemplo: gás e elétrica). De acordo com o autor, os modelos baseados em aprendizagem profunda (AP) superaram seis outros métodos de IA que haviam sido usados para este fim, os quais foram: regressão multilinear (MLR), rede elástica (RE), florestas aleatórias (FA), máquinas de aumento de gradiente (MAG), regressão de vetores de suporte (SVR), árvores de aumento de gradiente extremo (AAGE). Os autores usaram a lei de MPC visando a minimização do consumo de gás natural. O controlador agiu de modo a selecionar o melhor perfil de *setpoint* que alcançasse o menor consumo de gás natural.

Embora o trabalho de Cotrufo *et al.* (2020) não trate especificamente sobre a sintetização de MPCs baseado em IA, este é um artigo que deve ser levado em consideração para o seguimento deste trabalho, uma vez que o MPC depende de um modelo, e a obtenção de um modelo caixa preta para implementação do controlador é uma estratégia que pode ser considerada.

Wang *et al.* (2021) treinaram uma RNA para emular um MPC em aplicações de conversores multiníveis modulares. Os autores destacam que, como o treinamento da rede foi realizado *offline*, a maior parte do custo computacional deste sistema de controle, tradicionalmente realizado em tempo real, deixou de existir. Através dessa abordagem, os

autores conseguiram que a RNA alcançasse um desempenho de controle comparável ao do MPC convencional para esta mesma aplicação, mas com uma significativa redução no custo computacional para aplicações em tempo real.

Lale *et al.* (2021) combinaram aprendizado por reforço (RL) e controle preditivo no que eles chamaram de Controle Preditivo de Aprendizagem de Modelo (MLPC). O MLPC usa um modelo identificado de um processo não-linear parcialmente observável (um pêndulo invertido) obtido *online* por meio de uma função Não-Linear Autoregressiva com Entradas Exógenas (NARX) (Nelles, 2001), em que Representações de Fourier Aleatórias (RFF) (Rahimi e Recht, 2007) são usadas para representar o comportamento não-linear da planta. O algoritmo do MPC usado para a planta é obtido de modo online via RL.

O MLPC descrito em Lale *et al.* (2021) usa um MPC (chamado pelo autor de “oráculo”), dentro da função custo para projetar o sinal de controle do MPC obtido via RL para controle da planta. O MPC “oráculo” resolve um problema de controle ótimo de horizonte finito e fornece um sinal de controle para o modelo NARX identificado. A saída da planta é comparada com a saída produzida pelo “oráculo” no modelo NARX, para planejamento, atualizando ocasionalmente as estimativas do modelo subjacente para melhorar o MPC. Como relatado pelos autores a estrutura do MLPC para o problema do pêndulo invertido foi implementada em um toolbox para RL chamado OpenAI Gym (BROCKMAN, 2016).

Sobre o método RFF, este se mostra bastante útil na modelagem de sistemas não-lineares. De acordo com Rahimi e Recht (2007), a RFF é uma função eficiente para aplicar técnicas de aprendizado de máquina baseadas em *kernel*, como Máquinas de Vetores de Suporte (SVMs) e regressão *kernel*, a grandes conjuntos de dados, que de outra forma seriam proibitivamente caros de se computar diretamente devido ao custo computacional de avaliar o *kernel*. Isso, porque esta função é capaz de mapear os sinais do sistema (relação entrada/saída) que se quer modelar para um espaço de alta dimensão (possivelmente infinita), sendo possível realizar operações lineares que correspondem a operações não-lineares no espaço original dos dados. Esse método é eficiente porque a dimensionalidade do espaço de características é finita (e frequentemente muito menor que a dimensionalidade do espaço de Hilbert original correspondente ao *kernel*), e as operações no espaço transformado são lineares e mais simples de calcular (Rahimi e Recht, 2007; Lale *et al.* 2021).

O conceito de *kernel* em aprendizado de máquina refere-se a uma função usada para mapear dados em um espaço de dimensão mais alta, onde é possível resolver problemas que não são linearmente separáveis no espaço original (Schölkopf e Smola, 2018). Essencialmente, um *kernel* é uma função que calcula o produto interno de dois vetores (representando dados)

em um espaço de características, sem a necessidade de calcular explicitamente a transformação para esse espaço (Müller, 2001).

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado da seguinte forma: A Seção 3.1 apresenta os princípios do Controle Preditivo Baseado em Modelo (MPC). A Seção 3.2 oferece uma breve revisão sobre Redes Neurais Artificiais (RNA) *feedforward*. A Seção 3.3 discute a função de correlação cruzada, utilizada na fase de pré-processamento dos sinais. Por fim, a Seção 3.4 aborda a equação de ajuste, aplicada na etapa de teste da RNA no presente trabalho.

As equações e deduções que serão mostradas nas subseções a seguir são bem conhecidas no campo do MPC e das RNA *feedforward*. Para um estudo detalhado sobre MPC recomenda-se a leitura dos livros de Camacho e Bordón (1995), Richalet e O'Donovan (2009) e Wang (2009), referenciados ao longo da Seção 3.1. Para um estudo detalhado sobre RNAs, recomenda-se a leitura de Li *et al.* (1991), Pham e Xing (1995), Haykin (2009) e Nelles (2001).

3.1 Controle Preditivo Baseado em Modelo (MPC)

Controle Preditivo Baseado em Modelo (MPC) é uma técnica avançada de controle de processos que usa modelos para prever e otimizar o comportamento futuro de um sistema (Richalet e O'Donovan, 2009). Utilizando um horizonte de previsão para antecipar eventos futuros e um horizonte de controle para determinar as ações de controle ótimas, o MPC ajusta as entradas do sistema de maneira a cumprir os objetivos de controle, como seguir uma trajetória desejada ou manter uma condição operacional estável, respeitando as restrições do sistema (Wang, 2009).

No projeto de um MPC, é comum a utilização de modelos em espaço de estados devido à sua capacidade de representação compacta e adequada das dinâmicas de um sistema (Richalet e O'Donovan, 2009). Esse modelo é essencialmente um conjunto de equações que descrevem como o estado do sistema evolui ao longo do tempo em resposta a entradas externas (Ogata, 1995). A informação atual do sistema é suficiente para prever seu comportamento futuro. Isso é formalizado através da variável de estado, que encapsula o estado presente e é utilizada para calcular o estado em um próximo instante de tempo (Wang, 2009)

$$x_m(k + 1) = A_m x_m(k) + B_m u(k) \quad (3.1)$$

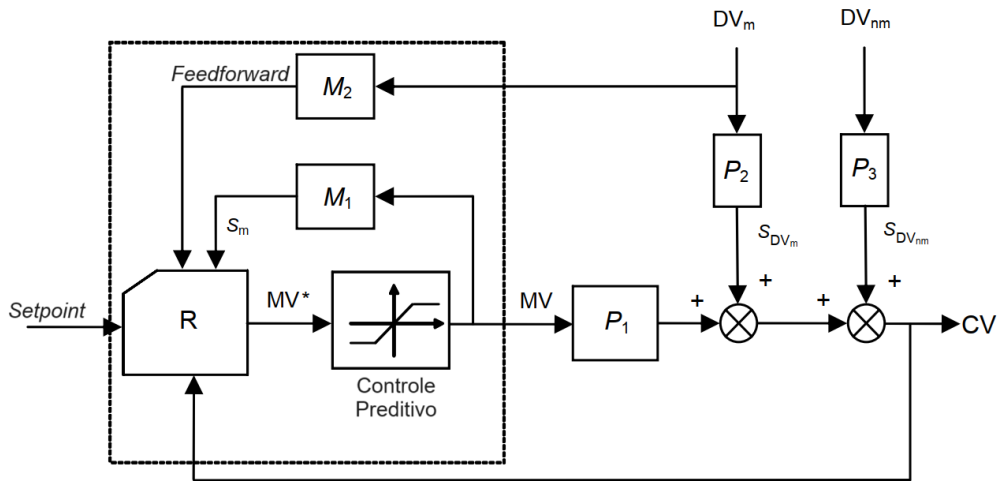
$$y(k) = C_m x_m(k) + D_m u(k) \quad (3.2)$$

onde $u(k)$ é a variável manipulada ou variável de entrada, $y(k)$ é a saída medida do processo e x_m o vetor de variáveis de estados com dimensão n_1 . A matriz A_m captura a dinâmica do sistema, enquanto B_m e C_m descrevem a relação entre a entrada e o estado, e o estado e a saída, respectivamente; e D_m representa a matriz que teoricamente transmitiria o sinal de entrada diretamente para a saída. No entanto, como a entrada $u(k)$ não pode afetar a saída $y(k)$ ao mesmo tempo, então, $D_m = 0$. Assim, consideramos somente:

$$y(k) = C_m x_m(k) \quad (3.3)$$

A Fig. 1 mostra um diagrama de blocos genérico de um MPC. O passo a passo do funcionamento deste controlador é dado a seguir (Richalet e O'Donovan, 2009):

Figura 1: Diagrama de blocos de um MPC.



Fonte: Richalet e O'Donovan (2009).

1. **Definição do *setpoint*:** O processo começa com a definição de um *setpoint*, que é o objetivo ou o valor desejado para a variável controlada (CV). Este é o alvo que o sistema de controle tenta alcançar e manter.
2. **Controlador recebe o sinal de *setpoint*:** O controlador R recebe o *setpoint* como referência e usa os modelos internos M_1 e M_2 para prever o comportamento futuro do processo. O modelo M_1 simula a resposta do processo a mudanças na variável manipulada (MV), enquanto M_2 considera as perturbações externas que podem afetar o processo.

3. **Cálculo da variável manipulada (MV^*):** O controlador, utilizando a lógica do MPC e os modelos de previsão, calcula a variável manipulada ideal MV^* , que é a ação de controle que o sistema teoricamente deve aplicar para que a variável controlada (CV) atinja o *setpoint*.
4. **Aplicação de restrições à MV^* :** A variável manipulada calculada MV^* pode precisar ser ajustada por um limitador para garantir que não exceda os limites operacionais ou de segurança do processo. Isso resulta na variável manipulada aplicada MV , que é a ação de controle que realmente será implementada no processo.
5. **Ação de controle no processo:** A variável manipulada MV é então enviada para o modelo interno do regulador M_1 e para o processo P_1 afetando a variável controlada CV e tentando trazê-la para o valor desejado definido pelo *setpoint*.
6. **Medição da variável controlada e das perturbações:** Enquanto isso, o sistema de controle mede continuamente a variável controlada CV e pode também medir as perturbações conhecidas DV_m . As perturbações não medidas DV_{nm} , por definição, não são medidas diretamente, porém afetam diretamente o processo por meio do sinal $S_{DV_{NM}}$. Já a perturbação DV_m afeta o processo por meio do sinal S_{DV_M} .
7. **Feedforward das Perturbações:** Se perturbações são esperadas ou medidas, como DV_m , um sinal de *feedforward* pode ser gerado e aplicado ao controlador para compensar essas perturbações antes que elas afetem a variável controlada.
8. **Ajuste do controlador baseado nas medidas:** Com base nas medições e nas previsões atualizadas dos modelos internos M_1 e M_2 , o controlador ajusta continuamente a variável manipulada MV para minimizar a diferença entre o *setpoint* e a variável controlada CV, levando em consideração as perturbações.
9. **Resposta a perturbações não medidas:** Se houver perturbações não medidas afetando o processo, como $S_{DV_{NM}}$, o sistema de controle precisará ser robusto o suficiente para lidar com essas incertezas e manter a variável controlada o mais próximo possível do *setpoint*.

O processo de um MPC é iterativo e contínuo, com o controlador fazendo ajustes em tempo real com base nas informações que recebe e nas previsões que faz. A eficácia do MPC

reside na sua capacidade de antecipar eventos futuros e tomar ações preventivas para manter o processo estável e próximo ao seu objetivo desejado (Camacho e Bordón, 1995).

Ressalta-se que no presente trabalho tanto R quanto o bloco limitador serão substituídos por uma RNA.

As Subseções 3.1.1 e 3.1.2 contém fundamentos necessários para o entendimento do projeto do MPC.

3.1.1 Modelo em Espaço de Estado Aumentado

A fim de eliminar o erro do sistema de controle em regime permanente, é comum o uso de um modelo em espaço de estados aumentado. Para isso, se faz necessário incluir na Eq. (3.1) um termo somador. Este termo somador pode ser incluído obtendo-se a diferença para ambos os lados da Eq. (1). Esta operação pode ser vista na Eq. (3.4):

$$x_m(k+1) - x_m(k) = A_m(x_m(k) - x_m(k-1)) + B_m(u(k) - u(k-1)) \quad (3.4)$$

e pode ser reescrita, como:

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \quad (3.5)$$

onde $\Delta x_m(k+1) = x_m(k+1) - x_m(k)$, $\Delta x_m(k) = x_m(k) - x_m(k-1)$ e $\Delta u(k) = u(k) - u(k-1)$. Em que $\Delta x_m(k+1)$ é a variação do vetor de estado entre o número de amostra $k+1$ e k , $\Delta x_m(k)$ é a variação do vetor de estado entre o número de amostra k e $k-1$, e $\Delta u(k)$ é a variação do sinal de controle entre o número de amostra k e $k-1$.

De forma análoga, repetindo o procedimento de diferenças para a Eq. (3.3), obtém-se

$$y(k+1) - y(k) = C_m \Delta x_m(k+1) \quad (3.6)$$

Substituindo a Eq. (3.5) na Eq. (3.6), resulta

$$\begin{aligned} y(k+1) - y(k) &= C_m (A_m \Delta x_m(k) + B_m \Delta u(k)) \\ y(k+1) &= C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) + y(k) \end{aligned} \quad (3.7)$$

A partir das Eqs. (3.5) e (3.7) pode-se obter a equação em espaço de estados aumentada:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) \quad (3.8)$$

$$y(k) = [o_m \ 1] \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \quad (3.9)$$

onde $o_m = [0 \ 0 \ \dots \ 0]$ e $x(k) = [\Delta x_m(k) \ y(k)]^T$. As Eqs. (3.8) e (3.9) serão usadas no projeto do MPC.

Para fins de escrita, as Eqs. (3.8) e (3.9) serão renomeadas conforme a seguir:

$$A = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix} \quad B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \quad C = [o_m \ 1]$$

Em consequência, o modelo aumentado pode ser reescrito de acordo com as Eqs. (3.10) e (3.11) (Wang, 2009):

$$x(k+1) = Ax(k) + B\Delta u(k) \quad (3.10)$$

$$y(k) = Cx(k) \quad (3.11)$$

3.1.2 Controle Preditivo de Horizonte Recuado

No contexto do controle preditivo, uma vez que um modelo matemático é estabelecido, o próximo passo no desenvolvimento deste controlador consiste no cálculo da saída predita do processo com base nos sinais de controle futuros, que são tratados como variáveis que podem ser ajustadas. Esta predição é realizada dentro de uma janela de otimização específica. O sistema de controle preditivo tem como propósito principal a minimização do erro entre o set-point e a saída do processo prevista. Ele emprega informações atualizadas do processo, representadas pelo vetor de estado $x(k_i)$, e utiliza um modelo de espaço de estados para projetar o comportamento futuro da planta. A partir desse modelo, estabelece-se a trajetória de controle ótima por meio de uma série de incrementos de controle, denotada por $\Delta U = [\Delta u(k_i) \ \Delta u(k_i + 1) \ \Delta u(k_i + 2) \ \dots \ \Delta u(k_i + N_c - 1)]^T$ (Wang, 2009).

Embora a trajetória de controle ótima seja computada levando em conta N_c amostras futuras, o controle preditivo aplica somente a primeira amostra, $\Delta u(k_i)$, e desconsidera as restantes na implementação. Esse processo é repetido a cada novo período de amostragem, seguindo o princípio de controle de horizonte recuado, onde o *feedback* é intrínseco ao design do sistema de controle.

A lei de controle de MPC no contexto de restrições rígidas é proposta como a determinação do vetor de parâmetros ΔU que minimiza a função custo J (Wang, 2009):

$$J = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U \quad (3.12)$$

sujeito às restrições

$$\Delta U_{min} \leq \Delta U \leq \Delta U_{max} \quad (3.13)$$

$$Y_{min} \leq Fx(k_i) + \Phi \Delta U \leq Y_{max} \quad (3.14)$$

$$x(k_i)_{min} \leq x(k_i) \leq x(k_i) \quad (3.15)$$

onde a função de custo J apresentada na equação (3.12) representa o desvio entre o estado desejado R_s e o estado estimado $Fx(k_i)$, penalizando também as variações no vetor de controle ΔU ; F é uma matriz que relaciona os estados futuros com o estado inicial $x(k_i)$; Φ é uma matriz que relaciona as entradas futuras ΔU com os estados futuros; \bar{R} é uma matriz diagonal, formada como $\bar{R} = r_\omega I_{N_c \times N}$, em que $r_\omega \geq 0$ é usado como um parâmetro de sintonia para desejada performance em malha fechada e I é a matriz identidade.

A minimização de J resulta em uma lei de controle que ajusta os incrementos de controle ΔU de modo a aproximar o sistema do setpoint R_s , considerando as restrições do problema. As restrições vistas nas equações (3.13), (3.14) e (3.15), limitam tanto o valor do controle ΔU , quanto as saídas $Fx(k_i)$ e os estados $x(k_i)$:

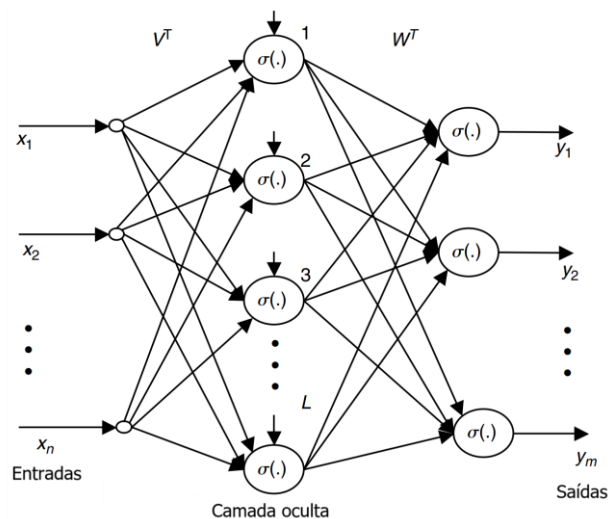
A técnica de controle preditivo discreto de horizonte recuado e com restrições é bem conhecida no campo do MPC. Logo, para o desenvolvimento da formulação matemática deste método, recomenda-se a leitura do livro de Wang (2009).

3.2 Rede Neural *Feedforward*

A RNA *feedforward*, também conhecida como rede neural de múltiplas camadas (MLP), é uma das arquiteturas mais antigas e mais simples dentro do campo da IA, com raízes históricas que remontam às primeiras tentativas de emular o comportamento de neurônios humanos (Aggarwal, 2018). Ela foi proposta inicialmente como um modelo matemático inspirado no cérebro humano, sendo capaz de aprender padrões a partir de sinais de entrada (Haykin, 2009).

A estrutura de uma RNA *feedforward* atualmente pode ser descrita como um conjunto de camadas interconectadas de neurônios artificiais, onde as conexões seguem um fluxo unidirecional, ou seja, os sinais percorrem da camada de entrada, passando pela(s) camada(s) oculta(s), até chegar à camada de saída. Não há realimentação entre os neurônios, o que caracteriza o nome "*feedforward*" (alimentação adiante). A Fig. 2 mostra uma rede *feedforward* com duas camadas.

Figura 2: RNA *feedforward* de duas camadas.



Para a Fig. 2, a camada de entrada é representada pelos neurônios x_1, x_2, \dots, x_n , que correspondem às variáveis de entrada do sistema. No contexto do controle preditivo, as entradas poderiam representar o vetor de erro do horizonte de predição. A camada oculta é composta por múltiplos neurônios interconectados que aplicam uma função de ativação não-linear, denotada como $\sigma(\cdot)$, com o objetivo de aprender padrões dos sinais de entrada. Cada neurônio nesta camada recebe entradas ponderadas, calculadas pelas matrizes de pesos V^T , e aplica uma função de ativação para gerar uma saída que será transmitida para a próxima camada. Após o processamento pela camada oculta (ou camadas ocultas), as saídas da rede y_1, y_2, \dots, y_m , são geradas na camada final (camada de saída). Neste trabalho, para o problema de emulação de um MPC, essas saídas representarão os incrementos de controle calculados pela RNA.

A equação matemática, em notação matricial, que descreve a RNA dada na Fig. 2 é mostrada na Eq. (3.16)

$$y = \bar{\sigma}(\bar{W}^T \bar{\sigma}(\bar{V}^T \bar{x} + b_v) + b_w) \quad (3.16)$$

Onde \bar{V} refere-se a matriz de pesos da primeira camada, \bar{W} a matriz de peso da segunda camada, $\bar{\sigma}$ o vetor de funções de ativação, b_v e b_w , os vetores de bias da primeira e da segunda camada, respectivamente.

O processo de treinamento desta rede consiste em ajustar os pesos V^T e W^T de forma que a rede minimize o erro entre suas saídas preditas e os valores esperados. Isso é feito utilizando um algoritmo de retropropagação, onde o erro calculado na saída é "propagado para trás" pela rede, ajustando os pesos de cada camada, até que a rede aprenda a aproximar a função desejada (Hecht-Nielsen, 1990; Haykin, 2009). Essa arquitetura, apesar de simples, tem sido amplamente utilizada devido à sua capacidade de aproximação universal, ou seja, teoricamente, uma rede *feedforward* com um número adequado de neurônios pode aproximar qualquer função contínua (Cybenko, 1989; Funahashi, 1989; Hecht-Nielsen, 1990; Hornik, 1991; Leshno *et al.*, 1993). Na prática, as redes *feedforward* são eficazes em uma variedade de aplicações, como previsão de séries temporais, classificação, e, como no caso deste trabalho, emulação de controladores avançados como o MPC.

Neste trabalho, a rede *feedforward* será treinada por meio do algoritmo de Levenberg-Marquardt, que é conhecido por ser eficiente para redes de tamanho moderado. Como descrito no livro do Haykin (2009), este método é um compromisso entre o método de Newton e o método do gradiente descendente. Para uma compreensão de como este método funciona, recomenda-se a leitura de Haykin (2009).

3.3 Função de Correlação Cruzada

Para conhecer melhor o grau de relacionamento linear entre os sinais de entrada u_1 e u_2 , e entre a entrada u_1 e saída y_2 , e entre a entrada u_2 e saída y_1 , a função de correlação cruzada dada pela equação Eq. (3.17) será aplicada

$$r_{AB}(n) = \frac{1}{N} \sum_{k=-\infty}^{\infty} f(k)g(n+k) \quad (3.17)$$

O resultado da FCC é normalmente dado entre -1 e 1, em que valores próximos de 1 mostram forte relação linear, já valores próximos de -1 mostram forte relação linear inversa. O eixo das abscissas, no cálculo da FCC, é chamado de atraso de tempo e diz respeito a defasagem no deslocamento entre o sinal móvel e o fixo, dos sinais que se quer analisar (Isermann e Münchhof, 2010).

Na prática, para que o resultado da FCC fique entre -1 e 1 se faz necessário realizar a adimensionalização dos sinais. A adimensionalização, para o cálculo da FCC é dada da seguinte

$$\text{forma } g_{adm} = \frac{g}{\sqrt{\sum g^2}} \text{ e } f_{adm} = \frac{f}{\sqrt{\sum f^2}}.$$

3.4 Teste

No teste o desempenho do modelo identificado é avaliado para um conjunto de dados diferente daquele usado para a estimação dos parâmetros. Isto pode ser possível, por exemplo, dividindo-se o conjunto de dados em três partes: os primeiros 2/3 do total do número de amostras é utilizado para a estimação de parâmetros, enquanto o 1/3 restante é usado para avaliar a qualidade do modelo (Juditsky et al., 1995).

A Eq. (3.18) mostra, em porcentagem, o quanto o modelo estimado se ajusta aos dados experimentais de validação:

$$ajuste = \left(1 - \frac{\sum_{k=1}^N |y_k - y_{sk}|}{\sum_{k=1}^N |y_k - \bar{y}|} \right) \times 100, \quad (3.18)$$

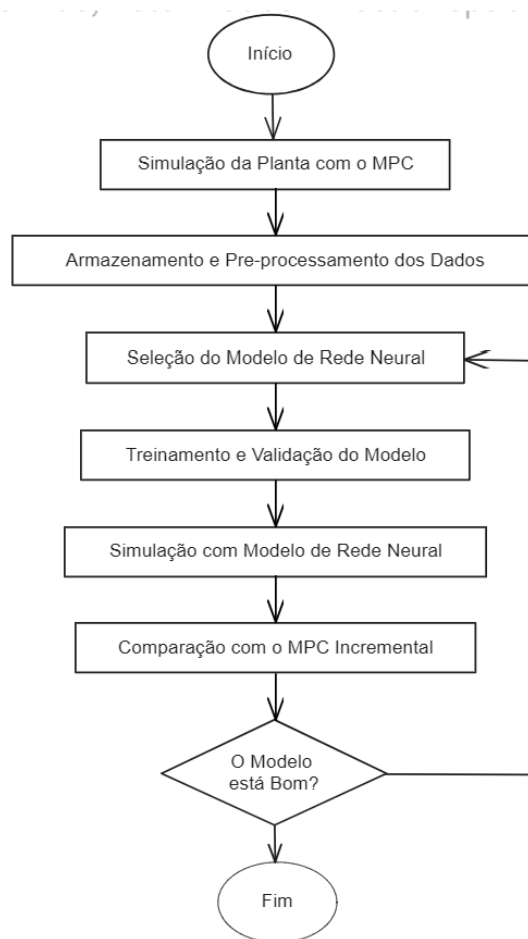
onde y_k (y, \dots, y_N) é a saída medida, y_{sk} (y_{s1}, \dots, y_{sN}) é a saída do modelo estimado, e \bar{y} é a média do vetor y_k . Para esta equação o valor de 100% corresponde a um ajuste perfeito.

Ressalta-se que no campo de identificação de sistemas dinâmicos, o termo “teste” é chamado de “validação”, como pode ser visto no artigo de Ljung e Hjalmarsson (1995). Neste trabalho, quando estiver sendo realizada a comparação visual e numérica entre as saídas y_1 e y_2 geradas pelo MPC (ao injetar o incremento de controle na planta), e as saídas y_1 e y_2 geradas pela RNA treinada para emular o MPC (ao injetar o incremento de controle na planta), o termo validação será usado com o mesmo contexto do termo teste. Este mesmo raciocínio será considerado para os sinais u_1 e u_2 .

4 METODOLOGIA

As etapas realizadas para a substituição do MPC por uma RNA são dadas no diagrama da Figura 3, e são detalhas nas subseções a seguir. Os resultados e discussões serão apresentados no capítulo 5.

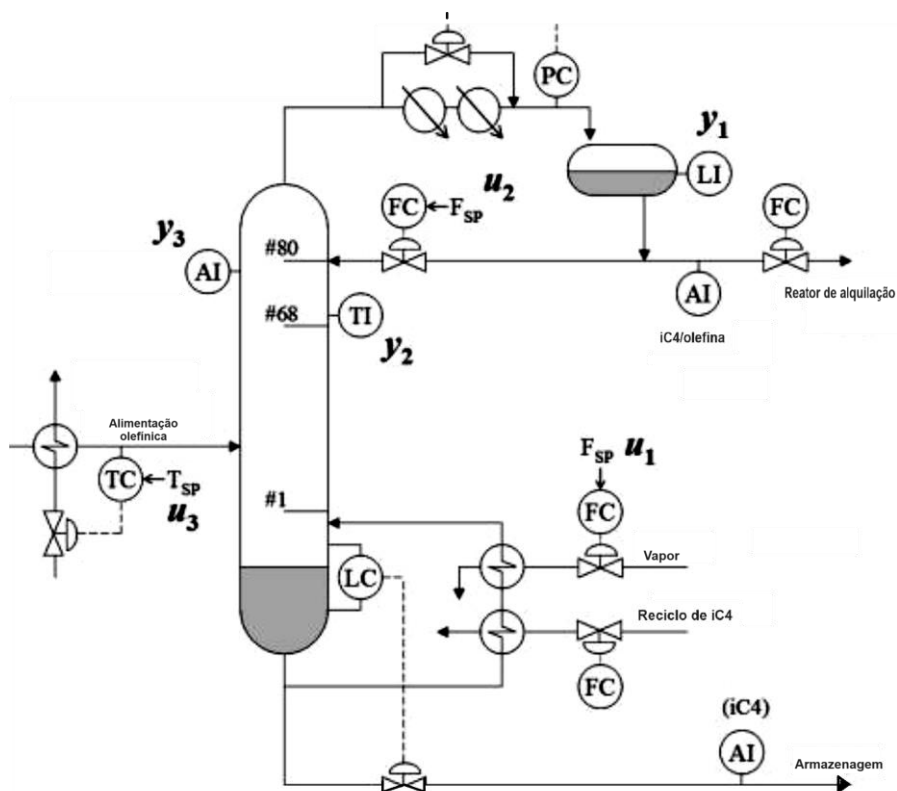
Figura 3: Etapas usadas para substituição de um MPC por uma RNA.



4.1 Simulação da Planta com o MPC

O sistema da torre de destilação usado para simulação da planta com o MPC foi obtido do trabalho de Alvarez *et al.* (2009), e é mostrado na Figura 4.

Figura 4: Diagrama de instrumentação e tubulação de uma torre de destilação deisobutanizadora.



Fonte: Alvarez, *et al.* (2009) (adaptado).

O estudo considera uma coluna de destilação deisobutanizadora, usada na unidade de alquilação da refinaria de petróleo da PETROBRAS/Cubatão. Essa coluna recebe uma alimentação da unidade FCC, que inclui isobutano, 1-butenos, cis-2-butenos, trans-2-butenos, n-butano e n-pentano. O produto do topo, composto principalmente por isobutano e butenos leves, é direcionado ao reator de alquilação, enquanto a corrente do fundo, rica em n-butano e butenos pesados, é armazenada para posterior processamento (Alvarez *et al.*, 2009).

Para simulação deste processo Alvarez *et al.* (2009) consideraram um modelo de função de transferência simplificado, derivado de um modelo experimental mais complexo. Este modelo captura a relação entre as entradas e saídas do sistema. No presente trabalho, para controlar este sistema serão consideradas as entradas u_1 e u_2 as quais referem-se, respectivamente, a taxa de fluxo de vapor no refeedor (ton/h) e a taxa de fluxo de refluxo (m^3/d). Já as saídas y_1 e y_2 serão consideradas como nível de líquido no vaso superior (%) e temperatura no prato 68 ($^{\circ}C$), respectivamente.

Para a simulação do sistema controlado por meio do MPC considerou-se as seguintes característica do processo e MPC:

Quadro 1: Parâmetros do processo e do MPC considerados para coleta de dados.

Característica	Valor
Período de amostragem	1 minuto
Tempo de experimento	3000 minutos
Horizonte de predição	120 minutos
Horizonte de controle	3 minutos
Valores de regime permanente para u_1	4,7 ton/h
Valores de regime permanente para u_2	2,65 m³/d
Valores de regime permanente para y_1	47 %
Valores de regime permanente para y_2	52,5 °C
Restrições nas saídas (y_1)	$42,5 \leq y_1(k) \leq 43,5$ %
Restrições nas saídas (y_2)	$53,5 \leq y_2(k) \leq 54,5$ °C
Restrições nas entradas (u_1)	$0 \leq u_1 \leq 10$ ton/h
Restrições nas entradas (u_2)	$0 \leq u_2 \leq 10$ m³/d
Restrições nos incrementos de controle (Δu_1)	3,0 ton/h
Restrições nos incrementos de controle (Δu_2)	3 m³/d

Ressalta-se que o período de amostragem de 1 minuto foi mantido do trabalho de Alvarez *et al.* (2009). Como se trata de um sistema lento, este período de amostragem é suficiente para uma representação adequada do comportamento dinâmico do sistema. Com relação as restrições impostas ao MPC, elas permitem que o sistema opere dentro dos limites desejados, assegurando a estabilidade e eficiência do processo de destilação.

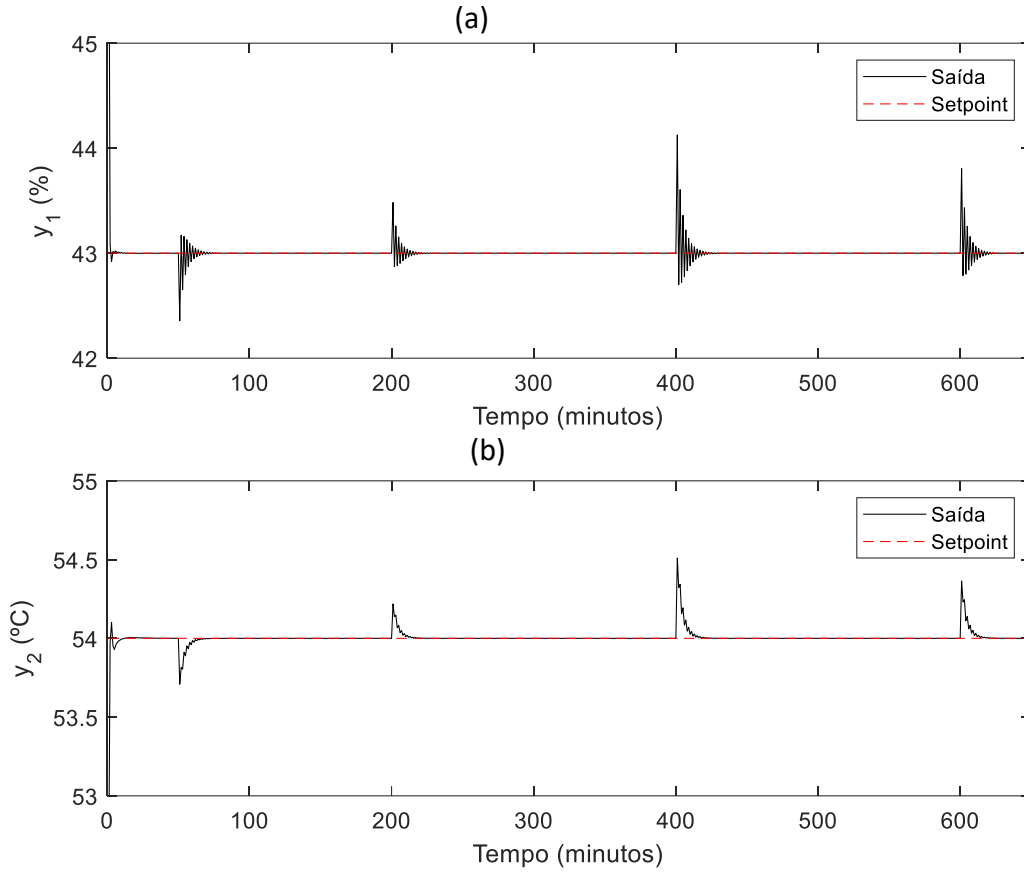
Para avaliar a robustez e a capacidade de generalização da RNA no controle do sistema de torre de destilação, foi necessário introduzir uma variedade de perturbações não medidas durante a simulação. Essas perturbações foram inseridas para simular condições de distúrbios que o sistema pode enfrentar. Elas foram introduzidas em quinze instantes específicos ao longo da simulação. Os momentos de aplicação das perturbações foram escolhidos para garantir uma distribuição uniforme ao longo do tempo de simulação, enquanto as magnitudes das perturbações foram variadas para simular diferentes intensidades de distúrbios. A Tabela 1 apresenta os instantes e magnitudes das perturbações aplicadas.

Tabela 1: Instantes de aplicação de perturbações.

Instantes	Magnitude
50	-0,4
200	0,3
400	0,7
600	0,5
800	-0,8
1000	-0,5
1200	0,4
1400	-0,9
1600	0,2
1800	-0,2
2000	-0,3
2200	0,8
2400	0,6
2600	-0,6
2800	0,7

O resultado da simulação, para uma janela de tempo de 450 minutos, para os sinais y_1 e y_2 podem ser observados nas Figuras 5 (a) e (b).

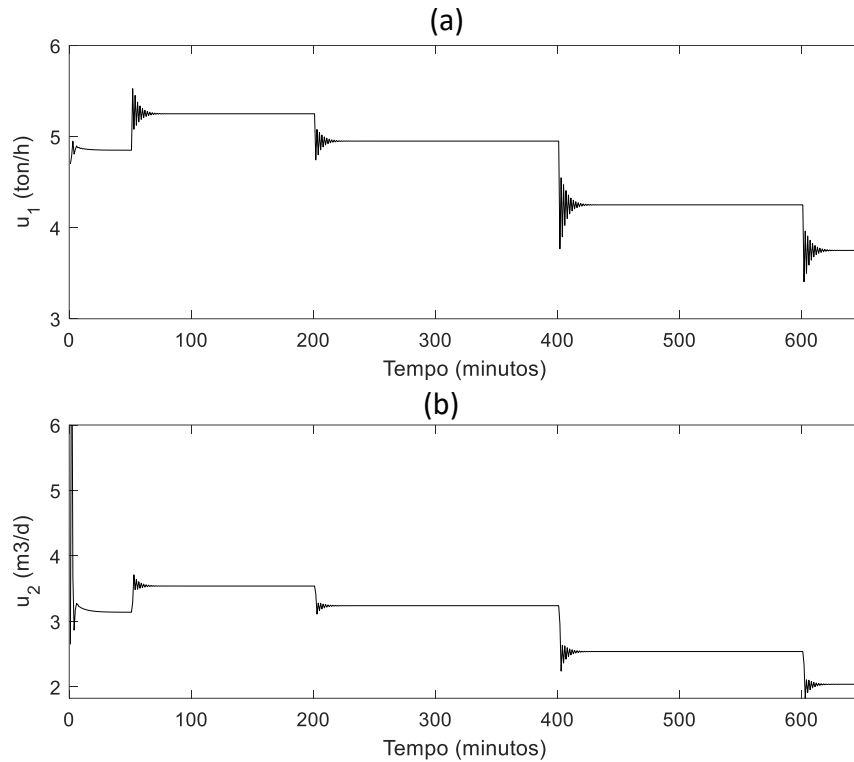
Figura 5: (a) Sinal de nível no vaso superior. (b) Sinal de temperatura no prato 68.



Observa-se que os sinais de saída permanecem obedecendo as restrições impostas mesmo após o sistema ter sido submetido a sequência de perturbações dada na Tabela 1. É importante ressaltar que o código implementado torna aleatória a sequência de aplicação das perturbações paramétricas a cada simulação. Logo, a ordem dos valores apresentados na Tabela 1, e consequentemente os resultados das Figuras 5 e 6, sofrerão alterações a cada simulação, porém os valores de magnitude são os mesmos.

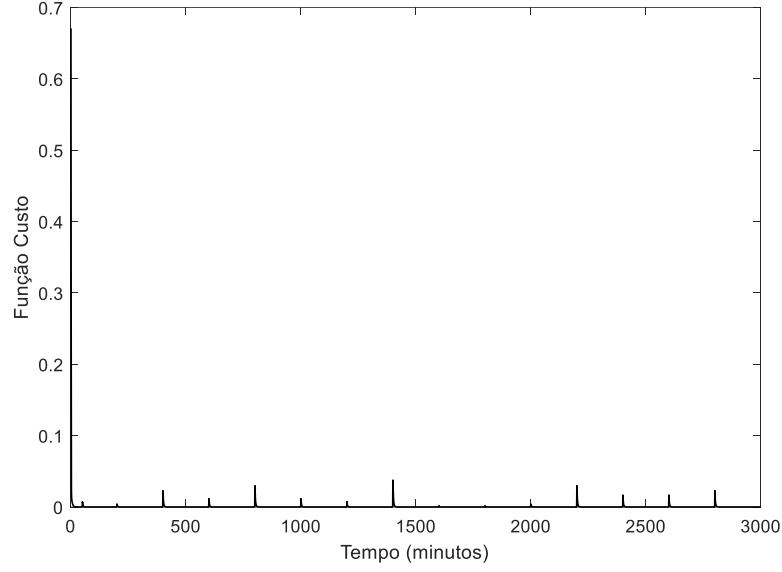
A Figura 6 mostra uma janela de tempo de 450 minutos dos sinais de controle u_1 e u_2 . Observa-se que os sinais de controle reagem às perturbações sofridas pelo sistema, na forma de variação paramétrica, para compensar as alterações sofridas nas variáveis controladas y_1 e y_2 .

Figura 6: (a) Taxa de fluxo de vapor no refeedor (ton/h). (b) Taxa de fluxo de refluxo (m³/d).



A Figura 7 mostra o resultado da função custo para um tempo de experimento de três mil segundos. Observa-se que as variações na função custo são resultantes das perturbações impostas na forma de variação paramétrica ao processo controlado. É importante notar que o algoritmo de otimização do MPC converge rapidamente e o sistema é controlado satisfazendo todas as restrições. Espera-se que esse mesmo comportamento seja obtido por meio da rede neural que será usada para emulação do MPC.

Figura 7: Resultado da função custo para um tempo de 300 minutos.



4.2 Armazenamento e Pré-Processamento dos Dados

Durante a simulação do processo controlado com o MPC, os sinais de erro e de incremento de controle ($\Delta u(k)$) são coletados. O vetor de erro, calculado como a diferença entre o vetor de *setpoint* (y_{sp}) e as saídas previstas do modelo ($F x_m$), é construído empilhando os *setpoints* para cada passo do horizonte de predição (p), resultando em um vetor de dimensão $p \cdot n_y$, em que n_y é o número de saídas do processo. Assim, a entrada da RNA é o vetor de erro calculado em cada passo de tempo, e, ao longo de todo o tempo de simulação, terá uma dimensão de $N \times p \cdot n_y$. Onde N se refere ao número total de amostras obtido durante a simulação. Do ponto de vista teórico, este número de amostras é dado por $\frac{t_{final}}{T_s}$, onde T_s representa o período de amostragem e t_{final} representa o tempo de simulação.

Para o presente trabalho será considerado $p = 120$, $T_s = 1$ s e um $t_{final} = 3000$ s. Isto resultará em um vetor de entrada da RNA de tamanho $p \cdot n_y = 120 \cdot 2 = 240$. Assim, a RNA que será considerada tanto no treinamento quanto na simulação terá dimensão de 1×240 a cada período de amostragem, totalizando ao longo de toda a simulação uma dimensão de 3000×240 .

Os sinais de incremento de controle $\Delta u(k)$ de saída que serão usados no treinamento da RNA, e posteriormente na simulação, são coletados a cada período de amostragem. Assim, a dimensão dos sinais de saída da RNA será 3000×2 .

Durante a simulação do processo foram adquiridas 720 000 amostras relativas aos sinais de entrada e 6000 amostras relativas aos sinais de saída. Estas amostras foram armazenadas do seguinte modo:

entradas de treinamento

$$= [e_1(1|k), e_2(1|k), e_1(2|k), e_2(2|k), \dots, e_1(120|k), e_2(120|k)], \text{ para } 1 \leq k \leq 3000;$$

$$\text{saídas de Treinamento} = \begin{bmatrix} \Delta u_1(1), \Delta u_1(2), \dots, \Delta u_1(3000) \\ \Delta u_2(1), \Delta u_2(2), \dots, \Delta u_2(3000) \end{bmatrix}^T.$$

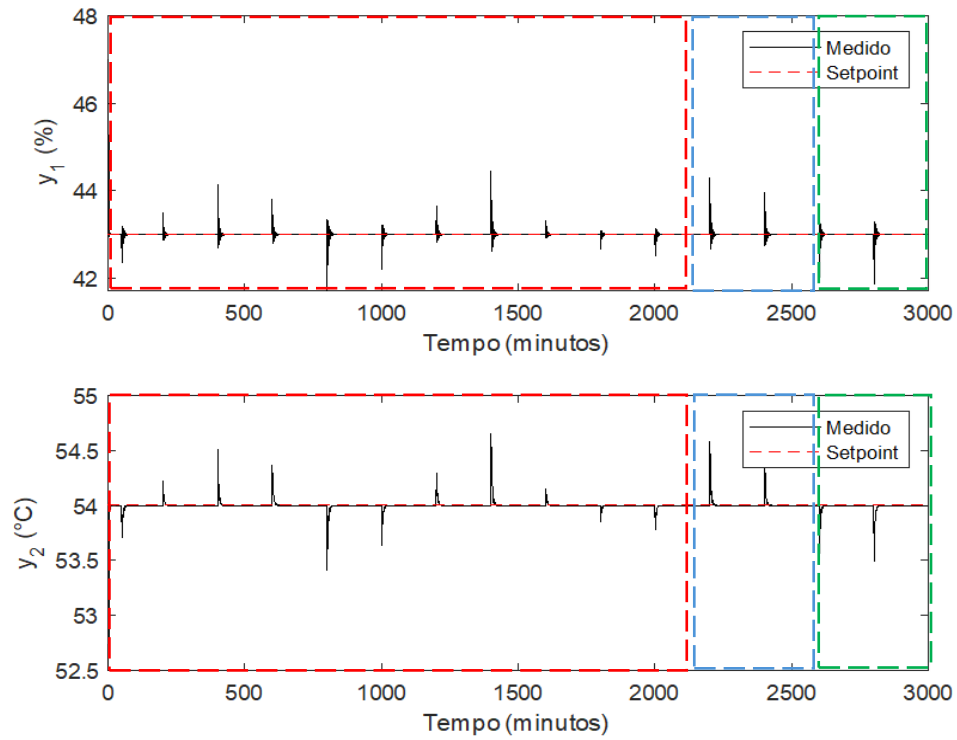
Em seguida, foram separadas em 70% para treinamento, 15% para validação e 15% para teste.

É importante ressaltar que os sinais $\Delta u_1(k)$ e $\Delta u_2(k)$ são selecionados baseados no conceito de horizonte recuado (Wang, 2009). Devido ao princípio do controle de horizonte recuado só se faz necessário tomar o primeiro elemento de Δu_1 e de Δu_2 , para cada instante k .

Quando o sistema começa a operar a partir de condições iniciais nulas, ele experimenta um transiente devido à resposta inicial aos valores de entrada e às condições de contorno impostas pelo controlador. Deste modo, o MPC ajusta as entradas para levar o sistema ao ponto de operação desejado (4,7 ton/h e 2,65 m³/d para u_1 e u_2 e 47 % e 52,5 °C para y_1 e y_2 , respectivamente) o que pode causar oscilações iniciais. Assim, como este comportamento só ocorre na partida no sistema, as amostras iniciais poderiam ser descartadas. No entanto, neste trabalho, decidiu-se manter este comportamento inicial do MPC para a RNA, pois em situações reais, no momento da partida do sistema, a RNA inevitavelmente será submetida a esse tipo e comportamento.

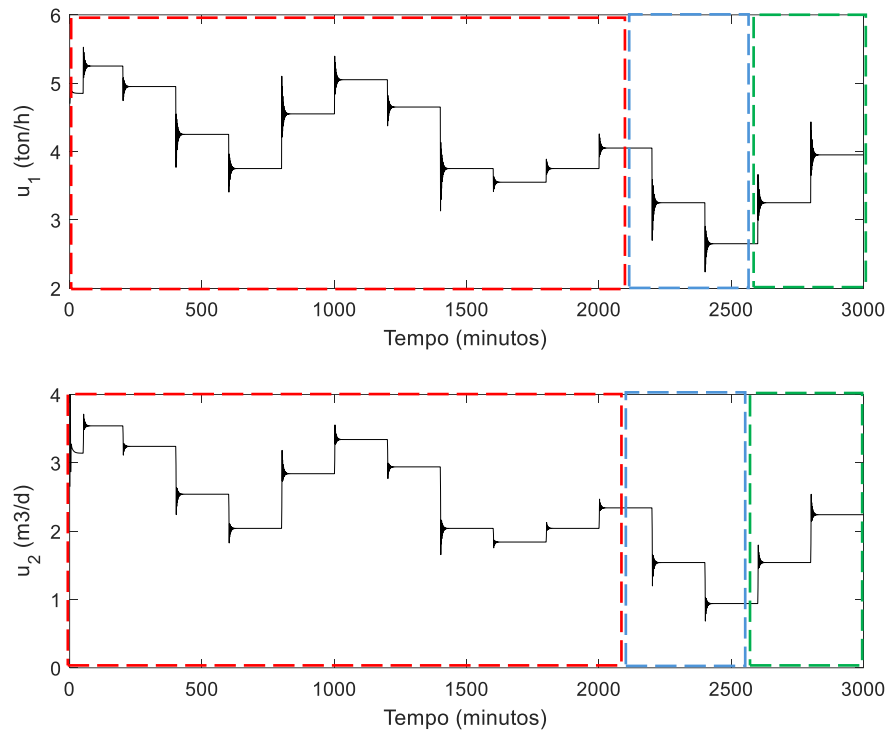
As janelas de dados usadas para treinamento, validação e teste da rede neural são ilustradas de acordo com a ordem mencionada na Figura 8 para os sinais de saída.

Figura 8: Separação dos sinais de saída em subconjuntos para treinamento (janela vermelha), validação (janela azul) e teste (janela verde) da rede neural.



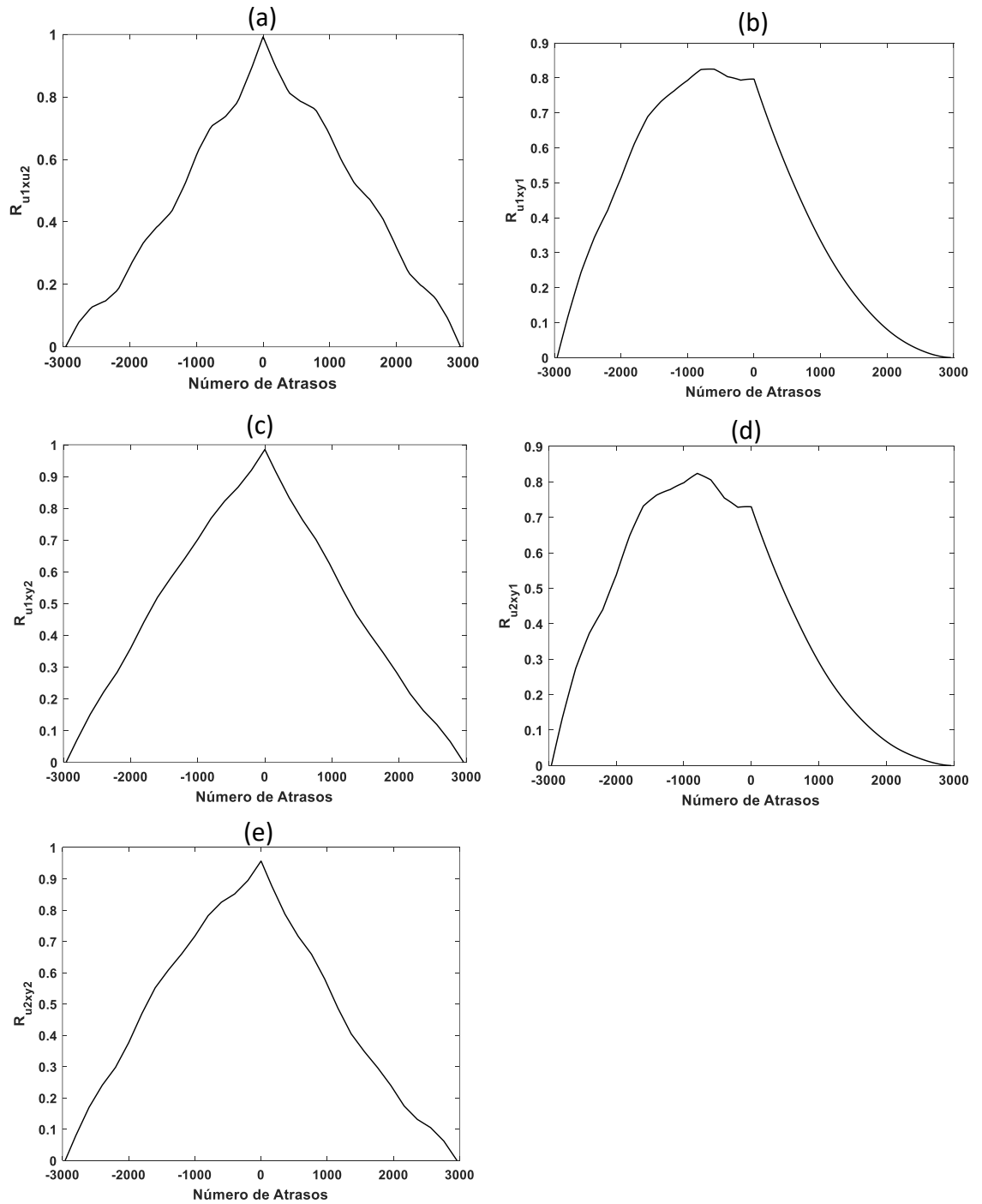
A Figura 9 ilustra as janelas de sinais de treinamento (vermelha), validação (azul) e teste (verde) para os sinais de entrada da rede neural.

Figura 9: Separação dos sinais de entrada em subconjuntos para estimação (janela vermelha), validação (janela azul) e teste (janela verde).



A fim de conhecer melhor o grau de relacionamento linear entre os sinais de entrada de entrada da planta u_1 e u_2 , e entre estes sinais com relação as saídas y_1 e y_2 , a função de correlação cruzada dada pela equação Eq. (22) foi aplicada. Os resultados obtidos são apresentados na Figura 10.

Figura 10: Resultado da FCC entre os sinais: (a) u_1 e u_2 ; (b) u_1 e y_1 ; (c) u_1 e y_2 ; (d) u_2 e y_1 ; e (e) u_2 e y_2 .



Observa-se na Figura 10 que os sinais estão fortemente relacionados de modo linear e de forma direta entre si, e isto indica que qualquer um destes sinais pode ser obtido, de forma aproximada, por meio de uma combinação linear com o outro sinal. Além disso, este é um indicativo que a

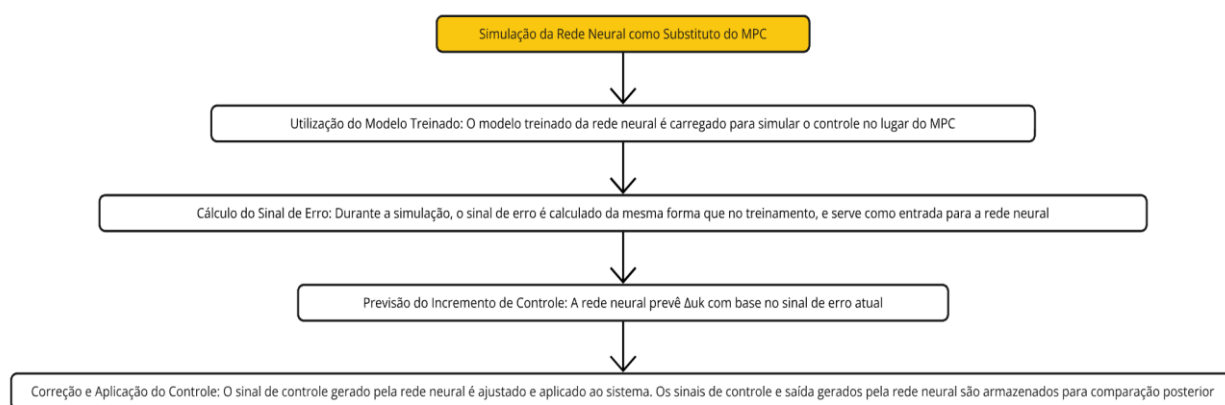
RNA que será treinada considerando o *setpoint* do sistema fixo, será suficiente para substituir o MPC quando for realizada uma simulação do MPC com mudanças aleatórias no *setpoint*.

4.3 Simulação com Modelo de Rede Neural

A simulação do processo utilizando uma das RNAs como controlador será conduzida de duas maneiras: com o *setpoint* constante e com variação do *setpoint*. Em ambos os casos, o modelo do processo não nominal estará sujeito tanto às entradas não medidas, conforme discutido na seção 4.1, quanto a desvios constantes no modelo do processo.

A Figura 11 apresenta o diagrama de fluxo do algoritmo implementado para a simulação da rede neural *feedforward* que emula o controlador preditivo baseado em modelo (MPC). O fluxo descreve as etapas seguidas durante a simulação do controle, partindo do momento em que o modelo da rede neural é carregado até a aplicação do controle no sistema.

Figura 11. Diagrama de fluxo para simulação da RNA que substituiu o MPC.



Inicialmente, a RNA previamente treinada para substituir o MPC, é carregada para realizar o controle em tempo real (primeira etapa). O sinal de erro, calculado como a diferença entre o *setpoint* e a saída prevista ao longo do horizonte de predição, é então determinado e utilizado como entrada da RNA, exatamente da mesma forma como foi utilizado no processo de treinamento (segunda etapa).

Na sequência, a RNA calcula o incremento de controle com base no erro, que será usado para corrigir o comportamento do sistema (terceira etapa). Este incremento é então somado ao controle anterior, e o controle ajustado é aplicado aos modelos simulados nominais e não nominais (quarta etapa) da torre de destilação. Durante essa última etapa, tanto os sinais de

controle quanto as saídas geradas pela RNA são armazenados para análise e comparação posterior com o MPC. E isso permite verificar a eficácia da RNA como substituta do controlador original.

5 RESULTADOS E DISCUSSÃO

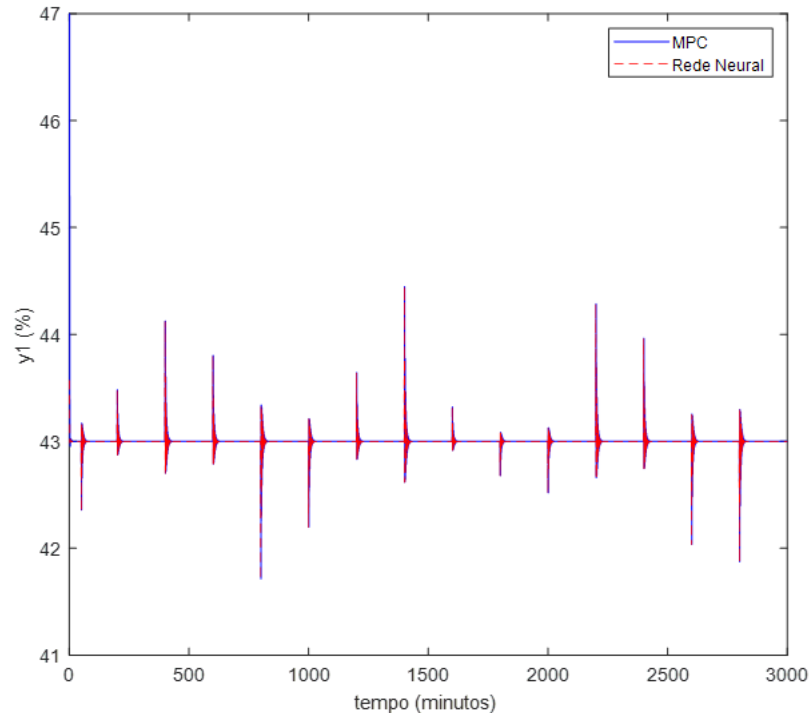
Neste capítulo, são apresentados e discutidos os resultados das simulações realizadas com o controlador baseado em RNAs, emulando o MPC linear na forma incremental.

5.1 Emulação do MPC linear incremental via RNA *Feedforward*

A arquitetura implementada nesta seção é uma RNA *feedforward* composta por uma camada de entrada, uma camada oculta e uma camada de saída. Como abordado no capítulo 4, a camada de entrada recebe um vetor de erro ao longo do horizonte de predição do controlador, com dimensão definida pelo número de elementos no vetor de erro. A camada oculta contém 10 neurônios, configurados com a função de ativação tangente hiperbólica, que mapeia os valores para o intervalo $[-1, 1]$. A rede foi treinada utilizando o algoritmo de Levenberg-Marquardt, com 70% dos dados alocados para treinamento, 15% para validação e 15% para teste. O treinamento teve como objetivo mapear os sinais de erro ao longo do horizonte de predição para os incrementos de controle correspondentes, permitindo que a rede reproduza o comportamento do MPC.

A Figura 12 compara o resultado da simulação da emulação do MPC linear incremental utilizando uma RNA *feedforward* com os resultados obtidos pelo MPC linear incremental como controlador, considerando o *setpoint* constante para a saída y_1 (nível no vaso).

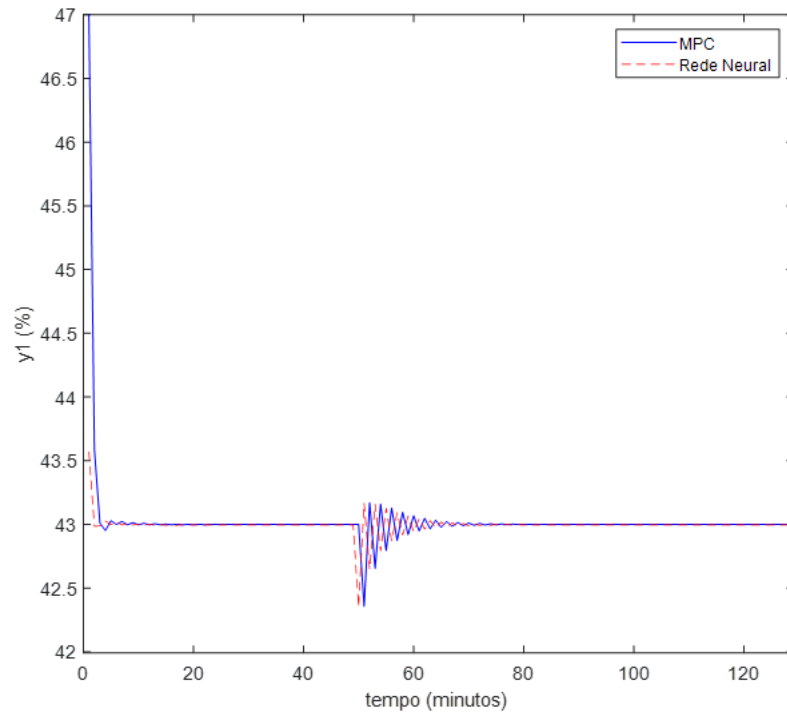
Figura 12: y_1 gerado pelo MPC versus y_1 gerado pela RNA *feedforward* com o *setpoint* constante.



A Figura 13 mostra uma janela de tempo de cerca de 130 minutos do resultado apresentado na Figura 12. Observa-se que a RNA *feedforward* emula de modo satisfatório o comportamento do sistema controlado pelo MPC. Como dito anteriormente, a RNA foi iniciada de modo aleatório e é capaz de se antecipar de um número de amostra (1 minuto) em relação a saída dada pelo MPC. Isto ocorre, pois embora a RNA *feedforward* não tenha memória temporal explícita, ao alimentar a rede com um vetor de erros futuros preditos, ela implicitamente aprende a realizar correções antecipadas. Esse é um comportamento similar ao de um controlador preditivo. Ou seja, a antecipação da rede neural não é um resultado intrínseco da arquitetura *feedforward* em si, mas da informação contida no vetor de erro ao longo do horizonte de predição.

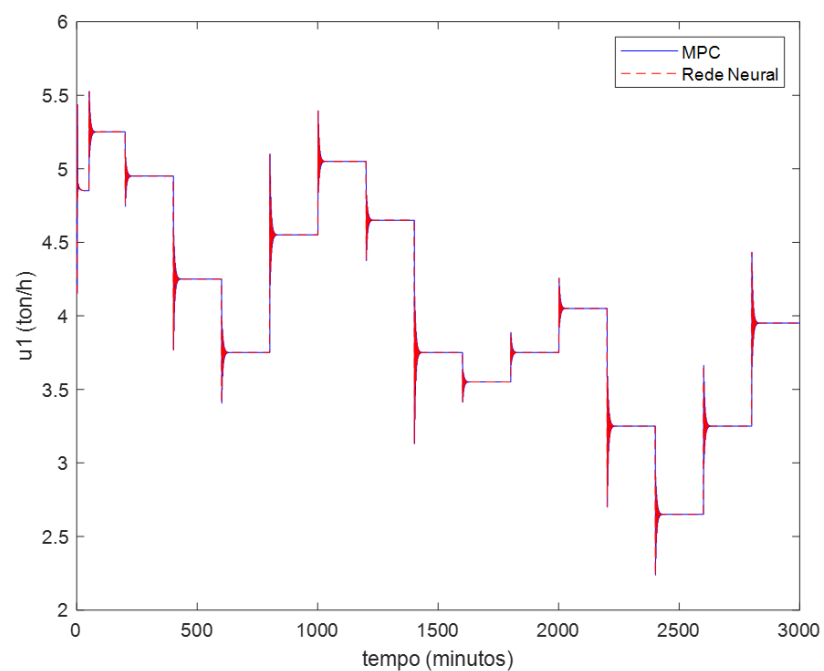
Em seguida, realizou-se uma comparação quantitativa entre os sinais apresentados na Figura 12. Para garantir a coerência dessa comparação, os sinais foram ajustados para estarem em fase e as 20 primeiras amostras foram descartadas. Esse procedimento é necessário para evitar que os valores do início da simulação, onde a RNA inicia com condições iniciais distintas das utilizadas pelo MPC, impactem negativamente o cálculo. Após o processamento, obteve-se um ajuste de aproximadamente 95% entre os sinais de saída y_1 , evidenciando um impacto similar entre ambos os controladores nesta saída do sistema.

Figura 13: y_1 gerado pelo MPC versus y_1 gerado pela RNA *feedforward* com o *setpoint* constante.



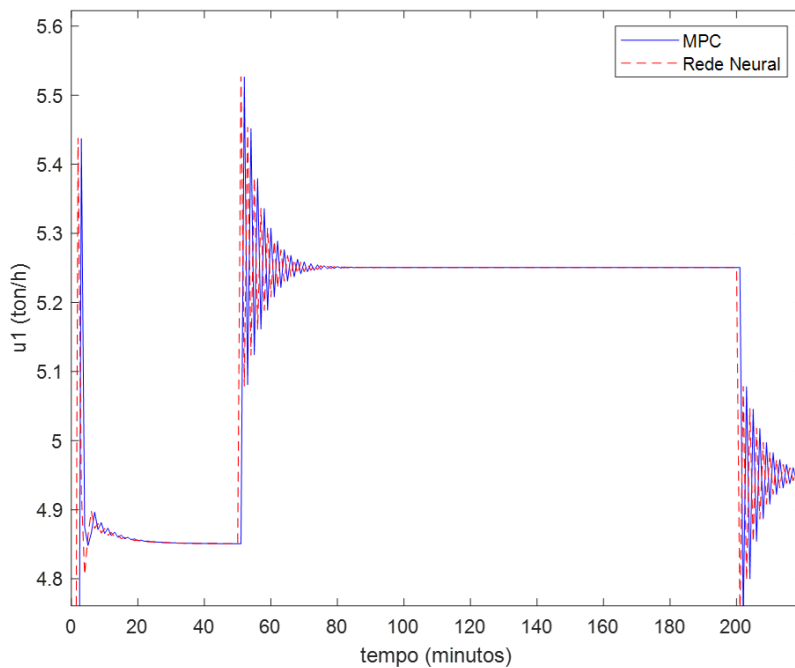
A Figura 14 mostra o sinal de controle u_1 (vazão de vapor) e compara o resultado deste sinal para a simulação da emulação do MPC linear incremental utilizando uma RNA *feedforward* com os resultados obtidos pelo MPC linear incremental como controlador.

Figura 14: u_1 gerado pelo MPC versus u_1 gerado pela RNA *feedforward*.



A Figura 15 mostra uma janela de tempo de cerca de 220 minutos do resultado apresentado na Figura 14. Observa-se que o sinal de controle produzido pela rede neural emula de modo aproximado o resultado produzido pelo MPC. Devido as razões já discutidas para o sinal de saída y_1 , o sinal de controle u_1 produzido pela RNA *feedforward* está antecipado de um número de amostra em relação ao do MPC.

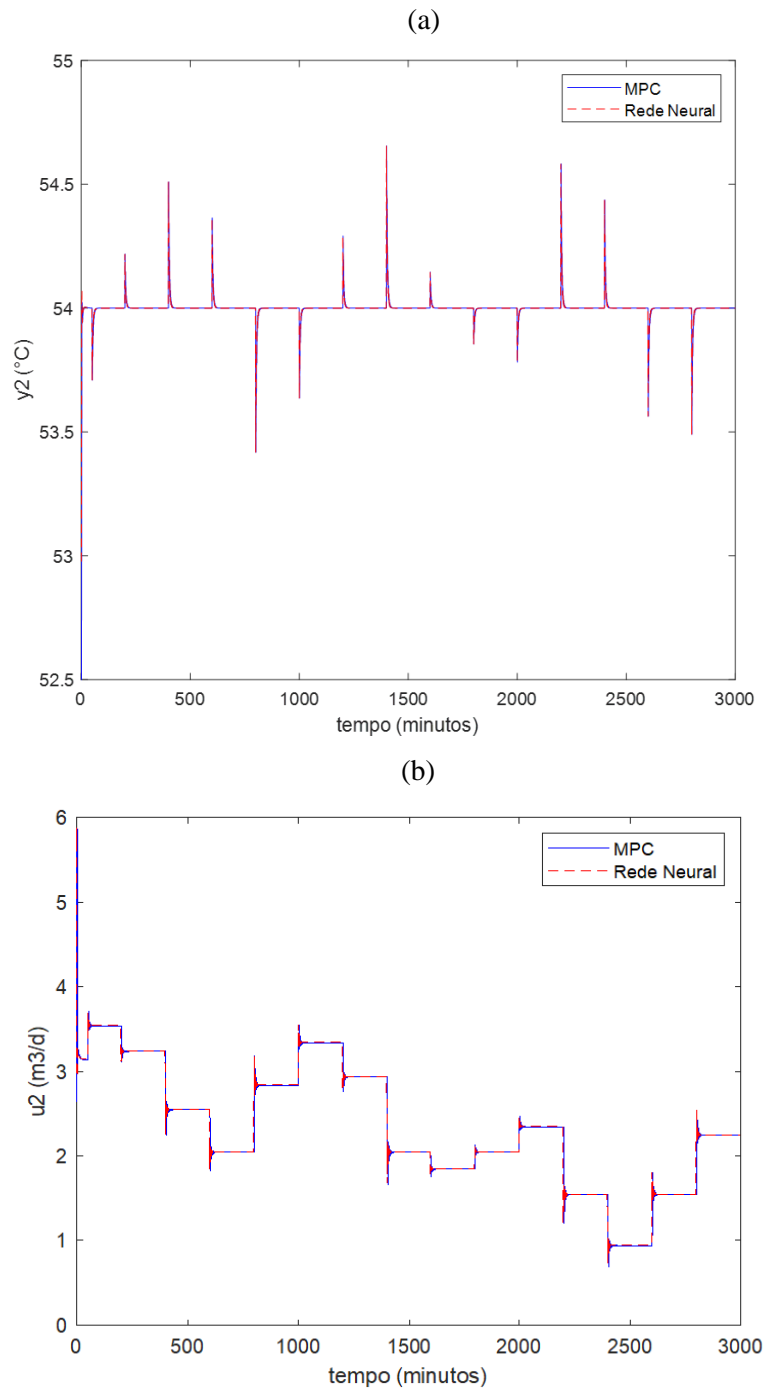
Figura 15: u_1 gerado pelo MPC versus u_1 gerado pela RNA *feedforward*.



Uma comparação quantitativa entre os sinais de controle u_1 gerados pela RNA e pelo MPC foi realizada, utilizando os mesmos critérios adotados para a saída y_1 . Assim, os sinais foram postos em fase e as 20 primeiras amostras foram descartadas para evitar influências das condições iniciais. Como resultado, obteve-se um ajuste de aproximadamente 99%, demonstrando uma forte concordância entre as ações de controle calculadas pela RNA e pelo MPC.

Resultados similares aos apresentados anteriormente, também foram obtidos para os sinais de temperatura y_2 e de vazão de refluxo u_2 . Os resultados da simulação para estes sinais, considerando o *setpoint* constante, são mostrados na Figuras 16 (a) e (b).

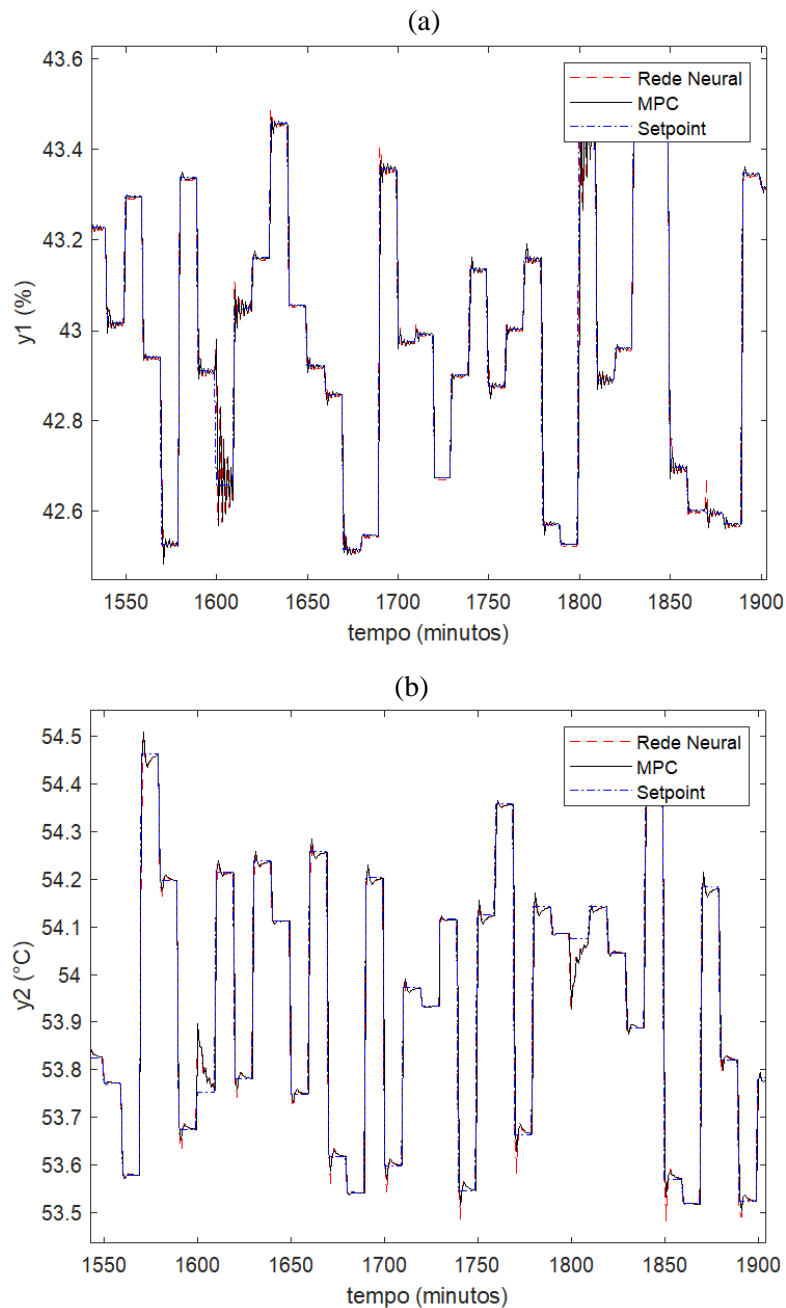
Figura 16: (a) y_2 gerado pelo MPC versus y_2 gerado pela RNA *feedforward* considerando *setpoint* constante. (b) u_2 gerado pelo MPC versus u_2 gerado pela RNA *feedforward*.



Seguindo os mesmos critérios adotados nas comparações anteriores para os sinais y_1 e u_1 , realizou-se a análise quantitativa para os sinais y_2 e u_2 . Os resultados mostram um ajuste de aproximadamente 99% tanto para o sinal y_2 quanto para o sinal u_2 , evidenciando uma forte correspondência entre os valores produzidos pela RNA e pelo MPC.

Na sequência deste estudo, a próxima simulação consistiu na mudança do *setpoint* do MPC, de forma aleatória, adotando uma *seed* para fins de comparação com a emulação do MPC por meio da RNA. Para esta emulação, utilizou-se a RNA treinada para o caso sem mudança de *setpoint*. Os resultados nas saídas do sistema y_1 e y_2 para uma janela de tempo de cerca de 370 minutos são apresentados na Figura 17.

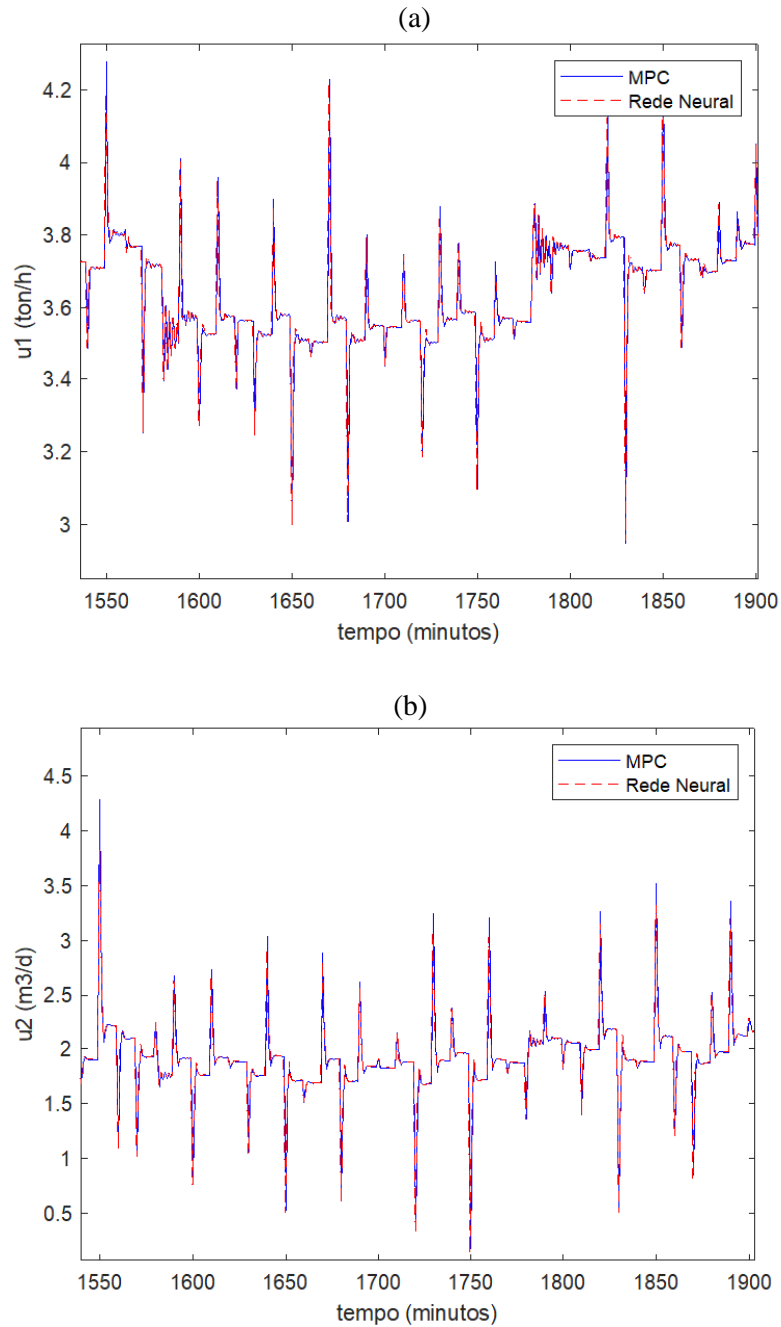
Figura 17: (a) y_1 gerado pelo MPC versus y_1 gerado pela RNA *feedforward* para uma mudança aleatória de *setpoint*. (a) y_2 gerado pelo MPC versus y_2 gerado pela RNA *feedforward* para uma mudança aleatória de *setpoint*.



Devido à forte relação linear entre os sinais de controle (u_1 e u_2) e saída (y_1 e y_2) da planta, observa-se que a RNA que havia sido treinada sem levar em conta a variação do *setpoint* entregou um resultado satisfatório mesmo considerando mudanças aleatórias no *setpoint*. O uso da equação do ajuste demonstra um ajuste de aproximadamente 97% entre os sinais de saída y_1 , e de 95% entre os sinais de saída y_2 .

Os resultados para os sinais de controle u_1 e u_2 , para uma janela de tempo de cerca de 370 minutos, são mostrados na Figura 18. Para estes sinais, o uso da equação do ajuste demonstra um ajuste de aproximadamente 98% entre os sinais de saída u_1 , e de 93% entre os sinais de saída u_2 .

Figura 18: (a) u_1 gerado pelo MPC versus u_1 gerado pela RNA *feedforward* com o *setpoint* aleatório. (b) u_2 gerado pelo MPC versus u_2 gerado pela RNA *feedforward* com o *setpoint* aleatório.

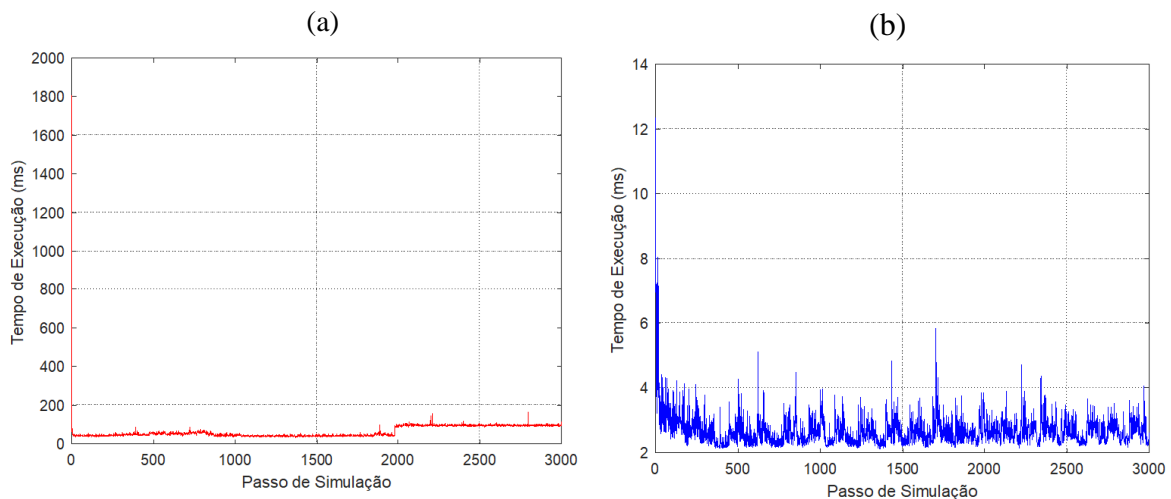


5.2 Análise de Tempo de Execução dos Controladores MPC e RNA

As Figuras 19 (a) e (b) mostram os resultados do tempo de execução do algoritmo de otimização do controlador MPC e da função de predição da RNA para cada passo de simulação considerando o *setpoint* constante. Observa-se que o tempo de execução da função de predição

da RNA é inferior ao do algoritmo de otimização do MPC. A fim de obter um valor médio comparativo, cada código de simulação dos controladores foi executado cinco vezes. Em seguida, calculou-se a média de tempo, para cada uma das cinco execuções, a partir do tempo médio gasto nos três mil passos de simulação. Para o MPC, obteve-se um valor médio de 40,7334 ms, enquanto para a RNA, obteve-se um valor médio de 2,4640 ms. Este resultado demonstra que a emulação do MPC via RNA é aproximadamente 16,5 vezes mais rápida do que a execução do algoritmo de otimização do MPC.

Figura 19: (a) Tempo de execução do algoritmo de otimização do MPC. (b) Tempo de execução da função de predição da RNA.



Cálculos similares de tempo de execução foram realizados considerando a variação aleatória do setpoint. Para o MPC, foi obtido um tempo médio de execução de 47,7619 ms, enquanto para a RNA o valor médio foi de 2,3098 ms. Esse resultado demonstra que a emulação do MPC por meio da RNA, mesmo com o setpoint variável, praticamente não sofreu alteração em relação ao caso de setpoint constante. Por outro lado, observa-se que a variação do setpoint aumentou significativamente o tempo de execução do algoritmo de otimização do MPC, resultando em um acréscimo de cerca de 7 ms no tempo médio.

6 CONCLUSÃO

O controlador preditivo baseado em modelo (MPC), a cada instante de tempo, resolve um problema de otimização com o objetivo de minimizar uma função custo que leva em conta o erro ao longo de um horizonte de predição. Contudo, o tempo gasto para a solução desse problema, seguido pela aplicação da correção, pode gerar uma latência indesejada no sistema de controle. Tal questão é particularmente crítica em sistemas que apresentam dinâmicas rápidas e alta complexidade, nos quais o MPC pode não ser capaz de fornecer uma solução dentro do tempo necessário.

A rede neural artificial (RNA) treinada neste trabalho, por outro lado, foi concebida para imitar o MPC sem a necessidade de resolver problemas de otimização a cada período de amostragem. A rede aplica diretamente as correções com base no vetor de erro ao longo do horizonte de predição, tornando-a uma alternativa promissora em situações em que o MPC é inviável, especialmente devido às suas altas exigências computacionais. Isso é especialmente relevante em sistemas embarcados, como controladores lógicos programáveis (CLPs) e microcontroladores, onde os recursos de processamento são limitados.

Apesar de o sistema analisado neste estudo ser de baixa complexidade - linear, invariante no tempo e estável - os resultados indicam que a substituição do MPC por uma RNA tem grande potencial. Como visto no capítulo 2, a literatura já sugere que RNAs podem ser aplicadas com sucesso em sistemas não-lineares e variantes no tempo, abrindo caminho para futuras investigações mais abrangentes sobre o tema.

Inicialmente, a proposta deste trabalho incluía a exploração de diferentes arquiteturas de inteligência artificial para emular o comportamento do MPC. Entretanto, como a rede *feedforward* selecionada conseguiu alcançar os resultados esperados, optou-se por não utilizar outras arquiteturas.

Para trabalhos futuros, seria interessante explorar a eficácia de outras configurações de IA, como redes neurais recorrentes ou com mecanismos de atenção, ampliando o escopo da emulação de MPCs.

Por fim, sugere-se a implementação desta rede neural em microcontroladores, CLPs ou outros sistemas embarcados, com o objetivo de testar seu desempenho em cenários mais próximos de aplicações reais. Essa etapa permitirá avaliar a viabilidade prática da proposta em termos de desempenho computacional e capacidade de controle.

REFERÊNCIAS

- AGGARWAL, C. C. **Neural Networks and Deep Learning: A Textbook**. 1. ed. Springer, 2018.
- ALVAREZ, L. A.; FRANCISCHINELLI, E. M.; SANTORO, Bruno F.; ODLOAK, Darci. **Stable Model Predictive Control for Integrating Systems with Optimizing Targets**. Industrial & Engineering Chemistry Research, v. 48, n. 20, p. 9141-9150, 2009.
- ATIA, I.; BAYOUMY, A. **A Neural Predictive Control Scheme for Small Turbojet Engines**. 14, 1-19, 2011.
- BROCKMAN, G.; CHEUNG, V.; PETTERSSON, L.; SCHNEIDER, J.; SCHULMAN, J.; TANG, J.; ZAREMBA, W. **OpenAI Gym**. 2016. Disponível em: <https://doi.org/10.48550/arXiv.1606.01540>. Acesso em: 12 de abril de 2024.
- CAMACHO, E. F.; BORDONS, C. **Model Predictive Control in the Process Industry**. Springer-Verlag London, 1995.
- CHAN, K. J.; PAULSON, J. A.; MESBAH, A. **Deep Learning-based Approximate Nonlinear Model Predictive Control with Offset-free Tracking for Embedded Applications**. In: 2021 American Control Conference (ACC), New Orleans, LA, USA, 2021. p. 3475-3481, doi: 10.23919/ACC50511.2021.9482849.
- CYBENKO, G. **Approximation by superpositions of a sigmoidal function**. Mathematics of Control Signal Systems 2, p. 303-314, Springer, 1989, <https://doi.org/10.1007/BF02551274>.
- COTRUFO, N. et al. **A practical artificial intelligence-based approach for predictive control in commercial and institutional buildings**. Energy and Buildings, [S.l.], v. 206, 2020. ISSN 0378-7788.
- DE PRADA, C. **Control Hierarchy of Large Processing Plants: An Overview**. In: Baillieul, J., Samad, T. (eds) Encyclopedia of Systems and Control, Springer, 2021, https://doi.org/10.1007/978-3-030-44184-5_241.
- FUNAHASHI, K. I. **On the Approximate Realization of Continuous Mappings by Neural Networks**. Neural Networks, Vol. 2, Issue 3, 1989. [https://doi.org/10.1016/0893-6080\(89\)90003-8](https://doi.org/10.1016/0893-6080(89)90003-8).
- HECHT-NIELSEN, R. **Theory of the backpropagation neural network**. In: International Joint Conference on Neural Networks, 1., 1989, Washington, D.C. Anais, 1989. v. 1, p. 593-605. DOI: 10.1109/IJCNN.1989.118638.
- HORNIK, K., STINCHCOMBE, M., & WHITE, H. **Multilayer feedforward networks are universal approximators**. Neural Networks, Volume 2, Issue 5, p. 359-366, 1989, [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8);
- HAYKIN, S. **Neural Networks and Learning Machines**. 3ª ed. Pearson, 2009. ISBN-13: 978-0-13-147139-9.
- ISERMANN, R, M; MÜNCHHOF, M. **Identification of Dynamic Systems - An Introduction with Applications**. Springer, 2010.

KHATHER, S.; IBRAHIM, M.; ABDULLAH, A. **Review and Performance Analysis of Nonlinear Model Predictive Control—Current Prospects, Challenges and Future Directions.** Journal Européen des Systèmes Automatisés, [s.l.], v. 56, n. 4, 2023. Disponível em: <https://doi.org/10.18280/jesa.560409>.

LESHNO, M., LIN YA., V.; PINKUS, A.; SCHOCKEN, S. **Multilayer feedforward networks with a nonpolynomial activation function can approximate any function.** Neural Networks, Vol. 6, Issue 6, p. 861-867, 1993. [https://doi.org/10.1016/S0893-6080\(05\)80131-5](https://doi.org/10.1016/S0893-6080(05)80131-5).

LI, J.; XU, X.; XI, Y. **Artificial neural networks-based predictive control.** In: International Conference on Industrial Electronics, Control, and Instrumentation (IECON), 17., 1991, Kobe, v. 2, p. 1405-1410. DOI: 10.1109/IECON.1991.239136.

LJUNG, L.; HJALMARSSO, H. **System Identification Through The Eyes Of Model Validation.** In Proc. Third European Control Conference, volume 3, Rome, Italy, pp 949-954, September, 1995.

LUYBEN, W. L. **Distillation Design and Control Using Aspen Simulation.** Wiley-AIChE, 2013.

MÜLLER, K. R.; MIKA, S.; RÄTSCH, G.; TSUDA, K.; SCHÖLKOPF, B. **An introduction to kernel-based learning algorithms.** IEEE Transactions on Neural Networks, v. 12, n. 2, p. 181-201, mar. 2001. DOI: 10.1109/72.914517.

NELLES, O. **Nonlinear System Identification: From Classical Approaches to Neural Network and Fuzzy Models,** Springer, 2001.

OGATA, K. **Discrete-time Control Systems.,** 2^a ed., Prentice-Hall, 1995.

PHAM, D. T.; XING, L. **Neural Networks for Identification, Prediction and Control.** London: Springer-Verlag, 1995. ISBN 978-1-4471-3246-2.

RAHIMI, A.; RECHT, B. **Random features for large-scale kernel machines.** In: Conference on Neural Information Processing Systems (NIPS). Citeseer, v. 3, n. 4, p. 5, 2007.

RICHALET, J.; O'DONOVAN, D. **Predictive Functional Control: Principles and Industrial Applications.** Springer-Verlag London Limited, 2009.

SANTANA, B.A.; MATOS, V.S.; SANTANA, D.D.; MARTINS, M.A.F. **Embedded MPC Strategies for ESP-Lifted Oil Wells: Hardware-in-the-Loop Performance Analysis of Nonlinear and Robust Techniques.** Processes, 2023, 11, 1354.

SCHÖLKOPF, B.; SMOLA, A. J. **Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.** The MIT Press, 2018.

TANGE, Y.; KIRYU, S.; MATSUI, T. **Model Predictive Control Based on Deep Reinforcement Learning Method with Discrete-Valued Input.** 2019 IEEE Conference on Control Technology and Applications (CCTA), 308-313, 2019.

WANG, L. **Model Predictive Control System Design and Implementation Using MATLAB®.** Advances in Industrial Control, Springer-Verlag London Limited, 2009.

WANG, Y.; BOYD, S. **Fast Model Predictive Control Using Online Optimization.** IEEE Transactions on Control Systems Technology, 18, 267-278, 2010.

WANG, S., DRAGIČEVIĆ, T., GONTIJO, G., CHAUDHARY, S., & TEODORESCU, R. **Machine Learning Emulation of Model Predictive Control for Modular Multilevel Converters.** IEEE Transactions on Industrial Electronics, 68, 11628-11634, 2021.

WILLIAMS, G.; WAGENER, N.; GOLDFAIN, B.; DREWS, P.; REHG, J. M.; BOOTS, B.; THEODOROU, E. A. **Information theoretic MPC for model-based reinforcement learning.** In: IEEE International Conference on Robotics and Automation (ICRA), 2017, [S.l.]. Anais... [S.l.]: IEEE, 2017. p. 1714-1721. DOI: 10.1109/ICRA.2017.7989202.