

**UNIVERSIDADE FEDERAL DE ALAGOAS  
UNIDADE ACADÊMICA CENTRO DE TECNOLOGIA  
CURSO DE ENGENHARIA QUÍMICA**

**MATHEUS HENRIQUE MONTEIRO SILVA**

**DESENVOLVIMENTO DE ESTRATÉGIAS PARA A UTILIZAÇÃO DAS TÉCNICAS  
DA ENGENHARIA 4.0 E TRATAMENTO DE SINAIS NA CONSTRUÇÃO DE  
CONTROLE PREDITIVO DE PROCESSOS EM LINGUAGEM PYTHON**

**MACEIÓ  
2024**

Matheus Henrique Monteiro Silva

DESENVOLVIMENTO DE ESTRATÉGIAS PARA A UTILIZAÇÃO DAS TÉCNICAS DA  
ENGENHARIA 4.0 E TRATAMENTO DE SINAIS NA CONSTRUÇÃO DE CONTROLE  
PREDITIVO DE PROCESSOS EM LINGUAGEM PYTHON

Trabalho de Conclusão de Curso (TCC), apresentado ao colegiado do curso de Engenharia Química da Universidade Federal de Alagoas (UFAL), como requisito parcial para a obtenção do título de Bacharelado em Engenharia Química.

Orientador: Prof. Dr. Frede de Oliveira Carvalho

Maceió  
2024

**Catálogo na Fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 – 1767

S586d Silva, Matheus Henrique Monteiro.  
Desenvolvimento de estratégias para a utilização das técnicas da engenharia 4.0 e tratamento de sinais na construção de controle preditivo de processos em linguagem *Python* / Matheus Henrique Monteiro Silva. – 2024.  
79 f. il. : figs. ; tabs. color.

Orientador: Frede de Oliveira Carvalho.  
Monografia (Trabalho de Conclusão de Curso em Engenharia Química).  
Universidade Federal de Alagoas. Centro de Tecnologia. Maceió, 2024.

Bibliografia: f. 75-79.

1. MPC (Controle preditivo). 2. Aprendizado profundo. 3. Método de decomposição. 4. Modelos híbridos. I. Título.

CDU: 66.0:004.81:159.953.5

Matheus Henrique Monteiro Silva

DESENVOLVIMENTO DE ESTRATÉGIAS PARA A UTILIZAÇÃO DAS TÉCNICAS DA ENGENHARIA 4.0 E TRATAMENTO DE SINAIS NA CONSTRUÇÃO DE CONTROLE PREDITIVO DE PROCESSOS EM LINGUAGEM PYTHON

Trabalho de Conclusão de Curso (TCC), apresentado ao colegiado do curso de Engenharia Química da Universidade Federal de Alagoas(UFAL), como requisito parcial para a obtenção do título de Bacharelado em Engenharia Química.

Data de Aprovação: 19/11/2024

**Banca Examinadora**

Documento assinado digitalmente  
 FREDE DE OLIVEIRA CARVALHO  
Data: 26/11/2024 15:02:51-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Frede de Oliveira Carvalho  
Universidade Federal de Alagoas-UFAL  
Campus A.C. Simões  
(Orientador)

Documento assinado digitalmente  
 ALTAIR MARQUES DA SILVA  
Data: 25/11/2024 11:26:49-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Altair Marques da Silva  
Universidade Federal de Alagoas-UFAL  
Campus A.C. Simões  
(Examinador)

Documento assinado digitalmente  
 JULIO INACIO HOLANDA TAVARES NETO  
Data: 26/11/2024 13:08:43-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Júlio Inácio Holanda Tavares Neto  
Universidade Federal de Alagoas-UFAL  
Campus CECA  
(Examinador)

## AGRADECIMENTOS

Primeiramente, agradeço a Deus por me dar força, esperança e resiliência mesmo nos momentos mais difíceis da minha vida, e que pela fé me fez chegar onde estou.

Agradeço aos meus avós Severino e Maria José, Sebastião e Maria do Carmo, por terem dado toda a estrutura e ensinamentos que embasaram minha família com amor, carinho, apoio e tranquilidade.

Agradeço aos meus pais Elias Severino da Silva e Lucicleide Dasdores Monteiro Silva e meu irmão Thiago Monteiro Silva e meus familiares por terem me dado apoio incondicional durante todos os momentos da minha vida e não ter medido esforços para me permitir alcançar oportunidades que me guiassem aos meus objetivos, não importando a dificuldade delas.

Agradeço a minha namorada, Síntia Freire, por ser uma companheira de todas as horas (boas e ruins), por ser paciente, compreensiva, por sempre me incentivar a ser a minha melhor versão e sempre me apoiar e ajudar quando eu achava que estava na minha pior fase.

Agradeço as amizades que fiz ao longo do caminho, que foram pilares de segurança, apoio nos momentos ruins e risadas nos momentos bons.

Agradeço ao meu Orientador Prof. Frede de Oliveira Carvalho, por me proporcionar a oportunidade de participar do LABSIA. E ao longo deste tempo, me incentivar a melhorar profissionalmente e academicamente a cada dia. Obrigado pela paciência, pelos conselhos e pelo aprendizado.

Agradeço ao PET PAESPE (Ciência e Tecnologia) por ter sido meu abrigo durante a faculdade, lá pude viver muitas experiências, aprender muitas coisas e conhecer muitas pessoas. Essas que foram fundamentais para minha jornada profissional e pessoal. Em especial, quero agradecer Marlos Barbosa, Andressa Veríssimo, Adiel Alves, Prof. Lucena e Prof. Luciano.

## RESUMO

Atualmente com a expansão da tecnologia, os processos industriais têm acesso a equipamentos cada vez mais eficientes, que permitem monitorar diversos aspectos ao mesmo tempo. Isto marcou o início de uma nova revolução, a Indústria 4.0, baseada na interconectividade dos dados da planta industrial. Desse modo, a área de controle também se adaptou, e criou o chamado controle avançado. O controle preditivo baseado em modelo, MPC é uma das novas técnicas criadas e baseia-se no uso de um modelo dinâmico do sistema para prever o seu comportamento futuro. Neste contexto, essa estratégia permite utilizar tecnologias da nova era industrial, principalmente no que se refere a modelos de inteligência artificial, máquinas de aprendizado, como modelos internos de previsão do sistema. Na atualidade, em termos de aplicação de máquinas de aprendizado, o aprendizado profundo é uma subclasse baseada em redes neurais artificiais que está sendo amplamente explorada e reportada na literatura. Dentre seus modelos, vale destacar as redes *long-short term memory* (LSTM) e a *gated recurrent unit* (GRU), assim como a perspectiva da utilização dessas técnicas acopladas com estratégias avançadas de pré-tratamento de dados como a *variational mode decomposition* (VMD), denominados de modelos híbridos. Portanto, a metodologia deste trabalho visa desenvolver, implementar e comparar, em *Python*, estratégias do controle clássico e avançado (MPC) para reação catalítica do etileno em um reator CSTR. Em relação ao controle clássico, o controlador proporcional-integral-derivativo (PID) obteve o melhor desempenho em relação a perturbações do tipo servo, ou seja, mudanças no valor ideal da variável controlada (*setpoint*). Além disso, ao avaliar sua aplicação em um sistema ruidoso, percebeu-se que não obteve um desempenho adequado. Em relação ao controle avançado, os controladores foram categorizados conforme o modelo interno empregado. O primeiro grupo utilizou apenas modelos de aprendizado profundo, LSTM e GRU, e obteve um desempenho satisfatório. Na avaliação perante a presença de ruído, ambos os controladores perderam desempenho, mas apresentaram valores melhores, ou comparáveis, em relação ao caso aplicado com PID. O segundo grupo utilizou modelos híbridos, combinando a VMD, na função de tratamento dos ruídos, e a previsão com a LSTM ou GRU. Porém, este grupo não obteve o desempenho esperado em nenhum dos casos, principalmente no que se refere a exposição ao ruído. Ao fim do trabalho, os controladores MPC-LSTM e MPC-GRU obtiveram as melhores métricas em relação aos dados sem ruído, e os controladores MPC-LSTM e MPC-VMD-LSTM para os ruidosos. Porém, os resultados encontrados indicam a necessidade de aprofundar em técnicas de sintonia dos parâmetros, principalmente nos controladores MPC com modelo híbrido. Por fim, conclui-se que a utilização do MPC em relação ao PID é promissora nos casos de aplicações mais complexas, ou que exijam maior flexibilidade do controlador. Enquanto que em aplicações mais simples, o PID tem vantagem de ser bem renomado e adaptado a indústria, além de ter um tempo de execução significativamente menor que o MPC.

**Palavras-chave:** MPC, Aprendizado Profundo, VMD, Modelos híbridos.

## ABSTRACT

Currently, with the expansion of technology, industrial processes have access to increasingly efficient equipment, allowing for the simultaneous monitoring of multiple aspects. This marked the beginning of a new revolution, Industry 4.0, based on the interconnectivity of data within industrial plants. Consequently, the control field has also adapted, creating what is known as advanced control. Model Predictive Control (MPC) is one of the new techniques developed, based on the use of a dynamic system model to predict its future behavior. In this context, this strategy enables the use of technologies from the new industrial era, particularly regarding artificial intelligence models and machine learning, as internal models for system prediction. Currently, in terms of machine learning applications, deep learning is a subclass based on artificial neural networks and is being widely explored and reported in the literature. Among its models, it is worth highlighting the long short-term memory (LSTM) networks and gated recurrent unit (GRU) networks, as well as the potential for using these techniques coupled with advanced data pre-processing strategies like variational mode decomposition (VMD), referred to as hybrid models. Thus, the methodology of this work aims to develop, implement, and compare, in Python, strategies of classical and advanced control (MPC) for the catalytic reaction of ethylene in a CSTR reactor. Regarding classical control, the proportional-integral-derivative (PID) controller achieved the best performance for servo-type disturbances, i.e., changes in the setpoint of the controlled variable. Additionally, when evaluating its application in a noisy system, it was observed that it did not perform adequately. In terms of advanced control, the controllers used in this work were divided based on the internal model employed. The first group used only deep learning models, LSTM and GRU, and achieved satisfactory performance. When evaluated under noisy conditions, both controllers lost performance but presented better or comparable results relative to the PID application. The second group used hybrid models, combining VMD for noise handling with predictions using LSTM or GRU. However, this group did not achieve the expected performance in any scenario, especially when exposed to noise. At the end of the study, the MPC-LSTM and MPC-GRU controllers achieved the best metrics for noise-free data, while the MPC-LSTM and MPC-VMD-LSTM controllers performed best under noisy conditions. However, the results indicate the need for further refinement of parameter tuning techniques, particularly for hybrid model MPC controllers. In conclusion, the use of MPC compared to PID is promising for more complex applications or those requiring greater controller flexibility. In contrast, for simpler applications, the PID has the advantage of being well-established and widely adapted in the industry, in addition to having significantly shorter execution times than MPC.

**Keywords:** MPC, Deep Learning, VMD, Hybrid Models .

## LISTA DE FIGURAS

Figura 1 – Estratégias de controle <i>Feedback</i> e <i>Feedforward</i> . . . . .	18
Figura 2 – Estratégia de controle cascata . . . . .	19
Figura 3 – Diagrama de blocos do controle <i>Feedback</i> . . . . .	20
Figura 4 – Diagrama do lugar das raízes . . . . .	22
Figura 5 – Oscilação sustentada para método de ZN . . . . .	25
Figura 6 – Estratégia MPC . . . . .	27
Figura 7 – Estrutura do MPC . . . . .	27
Figura 8 – Representação da RNN . . . . .	30
Figura 9 – Representação da LSTM . . . . .	31
Figura 10 – Representação da GRU . . . . .	33
Figura 11 – Estrutura do algoritmo genético . . . . .	36
Figura 12 – Método de <i>crossover</i> de um-ponto . . . . .	36
Figura 13 – Reator CSTR para o sistema estudado. . . . .	38
Figura 14 – Fluxograma de metodologia do trabalho. . . . .	39
Figura 15 – Pontos de equilíbrio para a reação de hidrogenação catalítica de etileno. . . . .	42
Figura 16 – Simulação dos dados de entrada das redes. . . . .	47
Figura 17 – Comparação dos controladores . . . . .	52
Figura 18 – Lugar das raízes para controlador proporcional . . . . .	53
Figura 19 – Lugar das raízes para controlador proporcional-integral . . . . .	54
Figura 20 – Lugar das raízes para controlador proporcional-integral-derivativo . . . . .	54
Figura 21 – Resposta do controlador P com $K_c = 153,23$ . . . . .	55
Figura 22 – Resposta do controlador PI com $K_c = 9,623$ e $K_i = 0.0368$ . . . . .	56
Figura 23 – Resposta do controlador PID, com $K_c = 53,2882$ e $K_i = 0, K_d = 20,2567$ . . . . .	56
Figura 24 – Resposta do controlador PID, com $K_c = 53,2882$ e $K_i = 0, K_d = 20,2567$ e dados ruidosos . . . . .	57
Figura 25 – Resultados do modelo LSTM . . . . .	58
Figura 26 – Resultados do modelo GRU . . . . .	59
Figura 27 – Gráfico dos efeitos . . . . .	60
Figura 28 – Resultado do MPC-GRU com $lag = 7, H_p = 7, H_c = 1, \delta = 0,1$ e $\lambda = 100$ . . . . .	61
Figura 29 – Resultado do MPC-LSTM com $lag = 7, H_p = 2, H_c = 1, \delta = 100$ e $\lambda = 0,1$ . . . . .	62
Figura 30 – Resultado do MPC-GRU com $lag = 7, H_p = 7, H_c = 7, \delta = 100$ e $\lambda = 100$ . . . . .	62
Figura 31 – Desempenho da LSTM com dados ruidosos . . . . .	63
Figura 32 – Desempenho da GRU com dados ruidosos . . . . .	64
Figura 33 – Resultado do MPC-LSTM com $lag = 7, H_p = 2, H_c = 1, \delta = 100$ e $\lambda = 0,1$ e dados ruidosos . . . . .	65
Figura 34 – Resultado do MPC-GRU com $lag = 7, H_p = 7, H_c = 7, \delta = 100$ e $\lambda = 100$ e dados ruidosos . . . . .	65

Figura 35 – Tratamento do ruído com a VMD . . . . .	66
Figura 36 – Resultado do modelo híbrido VMD-GRU . . . . .	67
Figura 37 – Resultado do MPC-VMD-LSTM com $lag = 7$ , $H_p = 2$ , $H_c = 1$ , $\delta = 100$ e $\lambda = 0,1$ e dados ruidosos . . . . .	68
Figura 38 – Resultado do MPC-VMD-GRU com $lag = 7$ , $H_p = 7$ , $H_c = 7$ , $\delta = 100$ e $\lambda = 100$ e dados ruidosos . . . . .	69
Figura 39 – Resultado do modelo híbrido VMD-LSTM para dados sem ruído . . . . .	70
Figura 40 – Resultado do MPC-VMD-LSTM com $lag = 7$ , $H_p = 2$ , $H_c = 1$ , $\delta = 100$ e $\lambda = 0,1$ . . . . .	70
Figura 41 – Resultado do MPC-VMD-GRU com $lag = 7$ , $H_p = 7$ , $H_c = 7$ , $\delta = 100$ e $\lambda = 100$	71

## LISTA DE TABELAS

Tabela 1 – Estimativa dos parâmetros dos controlador por ZN, pelo primeiro método . . . . .	24
Tabela 2 – Estimativa dos parâmetros dos controlador por ZN . . . . .	25
Tabela 3 – Parâmetros utilizados no trabalho . . . . .	41
Tabela 4 – Parâmetros utilizados na otimização por algoritmo genético. . . . .	45
Tabela 5 – Hiperparâmetros utilizados no <i>RandomSearch</i> . . . . .	48
Tabela 6 – Níveis do planejamento experimental . . . . .	50
Tabela 7 – Métricas de erro para os controladores otimizados por AG . . . . .	57
Tabela 8 – Resultados dos hiperparametros para a LSTM . . . . .	58
Tabela 9 – Resultados dos hiperparametros para a GRU . . . . .	59
Tabela 10 – Comparação do desempenho das redes após a inserção do ruído . . . . .	64
Tabela 11 – Comparação do desempenho dos modelos híbridos com ruído . . . . .	67
Tabela 12 – Comparação dos controladores utilizados . . . . .	72
Tabela 13 – Comparação dos controladores com dados ruidosos . . . . .	73

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>2</b>	<b>OBJETIVOS</b>	<b>16</b>
2.1	Geral	16
2.2	Específicos	16
<b>3</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>17</b>
3.1	Controle clássico	17
3.1.1	Estratégias de controle clássico	17
3.1.2	Estabilidade	20
3.1.3	Tipos de controladores	22
3.1.4	Sintonia de controladores	24
3.2	Controle avançado	26
3.3	Aprendizado profundo	29
3.3.1	LSTM	31
3.3.2	GRU	32
3.4	VMD	34
3.5	Algoritmo genético	35
3.6	Processo estudado	37
<b>4</b>	<b>METODOLOGIA</b>	<b>39</b>
4.1	Modelagem e simulação	39
4.1.1	Obtenção das equações do sistema	40
4.1.2	Definição do ponto de operação do sistema	41
4.2	Controle clássico	42
4.2.1	Linearização	43
4.2.2	Cálculo da função de transferência	43
4.2.3	Estudo da estabilidade do sistema de malha fechada	44
4.2.4	Sintonia dos controladores	45
4.3	Controle avançado	45
4.3.1	Geração de dados	46
4.3.2	Treinamento das redes	47
4.3.3	Aplicação do MPC	49
<b>5</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>51</b>

5.1	Controle clássico . . . . .	51
5.2	Controle avançado . . . . .	58
5.2.1	MPC com aprendizado profundo . . . . .	58
5.2.2	MPC com modelos híbridos . . . . .	65
5.2.3	Comparação dos controladores . . . . .	71
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>74</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>75</b>

## 1 INTRODUÇÃO

O controle de processos busca garantir a estabilidade, ou seja, suprimir o impacto de perturbações para que as variáveis críticas para o processo estejam em uma faixa de valores desejados que assegurem fatores como a segurança, qualidade, restrições operacionais, economia, regulamentações, entre outros (Kravaris; Kookos, 2021).

Dentre as abordagens de controle, a clássica é uma abordagem consolidada e age sobre processo pela atuação de um controlador que em função do erro, desvio da variável de interesse em relação ao seu valor de referência (*setpoint*), altera os valores das variáveis manipuladas para a minimização do desvio. Todavia para sistemas complexos, como reações que são representadas por uma grande não-linearidade, essa abordagem se torna complicada e até ineficiente (Stephopoulos, 1984).

Neste contexto, estudos foram realizados com o objetivo de desenvolver novas formas de controle que atendessem esses sistemas. Dessa forma, foi criado o controle avançado que abrange, por exemplo, o controle adaptativo, robusto e preditivo com modelo (*Model Predictive Control*, MPC). Dentre essas técnicas, o MPC foi proposto no final da década de 70 com o intuito de desenvolver controladores multivariados com a capacidade de tomar decisões otimizadas de controle baseadas em dados do presente e previsões futuras. Essa abordagem obteve destaque em relação às outras e foi amplamente estudada e aplicada nas décadas subsequentes, com exemplos na indústria automotiva e de petróleo (Ren *et al.*, 2022).

O MPC é caracterizado pelo modelo utilizado que pode ser tanto linear (*Linear model predictive control*, LMPC), quanto não linear (*Non-Linear predictive control*, NMPC). Esses modelos são a base da técnica e são utilizados para realizar as previsões do comportamento do processo, que são usadas para atuar sobre as variáveis manipuladas, mediante minimização de uma função custo, para mitigar os impactos futuros causados por perturbações (Saraiva; Silva; Carvalho, 2020).

Ao mesmo tempo da elaboração de novas técnicas de controle, houve um acentuado desenvolvimento na indústria química em relação a sensores e equipamentos de automação que proporcionaram a disponibilidade de uma grande quantidade e qualidade de dados. Essa expansão tecnológica resultou em uma nova era baseada na interconectividade das tecnologias desenvolvidas.

A chamada indústria 4.0 é o atual cenário da revolução industrial, e é principalmente impulsionada por tecnologias que têm como propósito melhorar a eficiência, sustentabilidade,

qualidade e flexibilidade dos processos. Dentre as técnicas utilizadas estão a *internet* das coisas (IoT), robótica avançada, *Big Data*, computação em nuvem, inteligência artificial, entre outras. Nesse contexto, a Engenharia 4.0 busca atender às novas expectativas do mercado, visando suprir as necessidades dos processos a partir da aplicação e produção de sistemas inteligentes baseados nas técnicas citadas, incluindo o controle avançado de processos, como o MPC.

Dentre essas ferramentas, o uso das técnicas da inteligência artificial (IA), como modelos do MPC, teve bastante destaque no desenvolvimento de novas abordagens de controle (Ren *et al.*, 2022). A IA têm ganhado atenção e vem sendo amplamente aplicada nos mais diversos campos do conhecimento. Vale salientar que as Redes Neurais Artificiais (RNA), algoritmos pertencentes ao aprendizado de máquina (uma subclasse da IA), são amplamente utilizados nos mais diversos campos, como por exemplo engenharia e medicina, devido sua capacidade de aprender o comportamento do processo apenas a partir dos dados, sem requerer conhecimento prévio (Panerati *et al.*, 2019; Shahid; Rappon; Berta, 2019). Todavia, para problemas mais complexos e de difícil generalização, uma subclasse do aprendizado de máquina baseada em RNAs caracterizada por possuir uma densidade de neurônios e camadas elevada, foi desenvolvida e denominada aprendizado profundo (Géron, 2019; Pon Kumar *et al.*, 2018).

No conjunto de modelos propostos no aprendizado profundo as redes recorrentes, como as Redes Neurais Recorrentes (*Recurrent Neural Networks*, RNN), foram idealizadas para a predição de dados sequenciais (textos, áudio, séries temporais), pois sua arquitetura incorpora estruturas de retropropagação para simular a memória humana. Porém, a RNN enfrenta problemas ao lidar com dados sequenciais muito longos e perde informações importantes em posições iniciais. Para solucionar essa dificuldade foram desenvolvidas novas redes neurais, denominada de recorrentes, com a capacidade de reter as informações importantes em memórias de longo prazo. Dentre essas redes, a rede LSTM (*Long short-term memory*) e a GRU (*Gated recurrent unit*) são as que mais se destacaram na literatura (Géron, 2019).

Todavia, o desempenho dos modelos de IA depende, além da quantidade, da qualidade dos dados empregados no treinamento, de modo que dados ruidosos (comuns na indústria) dificultam a eficiência das técnicas de aprendizado de máquina e profundo, sendo necessário a aplicação de um pré-tratamento. Neste aspecto, as técnicas de decomposição de sinais são amplamente utilizadas para tratamento de sinais. Os modelos híbridos, ou seja, aqueles que utilizam mais de um algoritmo para realização de uma tarefa, por exemplo os de decomposição-predição que combinam a capacidade de previsão e generalização do aprendizado profundo com

o tratamento de dados dos métodos de decomposição (Huang *et al.*, 2021). Dentre os algoritmos de decomposição, a VMD (*Variational Mode Decomposition*) têm ganhado destaque por permitir dividir um sinal complexo em componentes modais mais simples a partir de uma decomposição variacional (Li *et al.*, 2021; Song *et al.*, 2021).

Neste sentido, há a possibilidade de empregar técnicas de aprendizado profundo e modelos híbridos para a construção de modelos de predição para MPC capazes de descrever o comportamento de sistemas complexos, não lineares e ruidosos de maneira eficiente (Qi; Li; He, 2018; Ren *et al.*, 2022).

Desta forma, neste trabalho serão investigadas e comparadas as estratégias de controle clássico e controladores MPC utilizando como modelos internos as técnicas da engenharia 4.0 (aprendizado profundo) e tratamento de sinais (VMD) para o controle de uma reação de hidrogenação catalítica de etileno realizada em um reator CSTR.

## 2 OBJETIVOS

### 2.1 Geral

Desenvolver e implementar, em linguagem de programação *Python*, estratégias de controle clássico e preditivo (MPC), baseado em aprendizado profundo e modelos híbridos, para reação de hidrogenação catalítica de etileno em um reator CSTR.

### 2.2 Específicos

- Desenvolver a modelagem fenomenológica do processo de hidrogenação catalítica de etileno em reator CSTR pelo uso de equações do balanço de massa e energia com sua simulação dinâmica realizada pelo método de *Runge-Kutta*;
- Aplicar controle clássico, abordando a construção da malha do controle, da função de transferência, do estudo de estabilidade do sistema e da sintonia clássica do controlador por Ziegler-Nichols ;
- Desenvolver e avaliar estratégias de controle preditivo (MPC) pela construção de controladores utilizando aprendizado profundo (LSTM e GRU) e modelos híbridos de decomposição-previsão (VMD-LSTM e VMD-GRU) como modelos internos do MPC;
- Comparar a eficiência das estratégias de controle utilizadas: controle clássico, MPC com LSTM, MPC com GRU, MPC com VMD-LSTM e MPC com VMD-GRU;
- Investigar a viabilidade da implementação desses modelos na indústria.

### 3 REVISÃO BIBLIOGRÁFICA

Para embasar este trabalho, é essencial conduzir uma revisão da literatura sobre os seguintes tópicos relevantes: controle clássico, controle avançado, aprendizado profundo, técnicas de otimização, decomposição de sinais (VMD) e o processo de hidrogenação catalítica de etileno. Os temas a serem revisados estão trabalhados a seguir.

#### 3.1 Controle clássico

O controle automático, ou controle clássico, é fundamental para o desempenho de qualquer campo da engenharia como no projeto de sistemas elétricos, robóticos e em especial para a engenharia química para as plantas industriais ( Kravaris; Kookos, 2021). O controle de processos é uma área essencial da indústria que visa garantir que o valor de variáveis importantes para o processo esteja em torno de um valor desejado, chamado de *setpoint*, de modo garantir a segurança, as condições operacionais, a economia e a qualidade do processo ( Stephopoulos, 1984). Para que isso aconteça, o controlador atua sobre uma ou algumas variáveis que impactam o sistema, chamadas de variáveis manipuladas, com objetivo de direcionar a variável de interesse, também denominada de variável controlada, até o *setpoint* (Seborg; Edgar; Mellichamp, 2016).

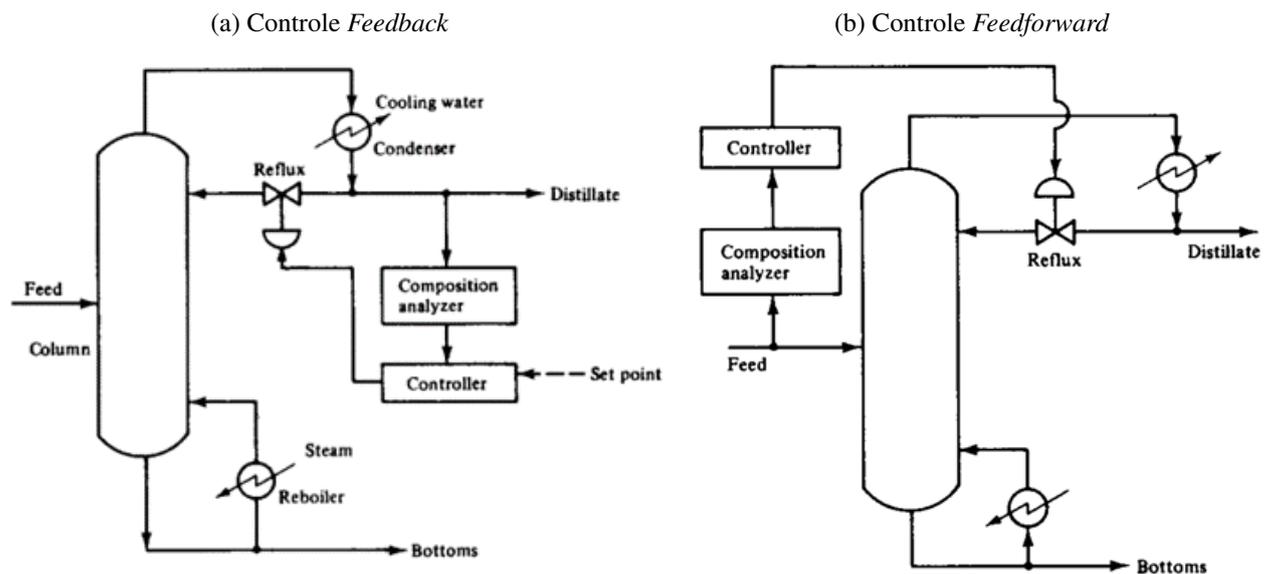
Os sistemas de controle podem ser definidos de duas formas a depender do uso das medições realizadas no processo. Os sistemas de retroalimentação, ou seja, em que ocorre a comparação dos valores aferidos no processo, com valores desejados, são denominados como sistemas de malha fechada. Esses sistemas utilizam dessa comparação para gerar um sinal de erro que é repassado a um controlador para que através da interpretação do erro, ele atue no processo a fim de minimizar o desvio. Enquanto isso, os sistemas de malha aberta não possuem nenhum tipo de retroalimentação para atuarem no processo, de modo a não atuarem diante a presença de qualquer tipo de distúrbio.

##### 3.1.1 Estratégias de controle clássico

No sistema de malha fechada, a atuação do controlador pode acontecer por três metodologias, a *feedback*, a *feedforward* e a cascata que dependem de qual medida será utilizada para a retroalimentação. Na configuração *feedback*, Figura 1a, a perturbação, interna ou externa ao sistema, deve acontecer primeiro para que o sistema possa perceber a mudança da variável controlada e assim comparar seu valor com o *setpoint*, como ocorre na análise da composição da Figura 1a, para realizar a ação de controle por intermédio das variáveis manipuladas. No controle

*Feedforward*, Figura 1b, diferentemente do *feedback* a mensuração dos distúrbios ocorrem antes que eles impactem o sistema. Dessa forma, por meio do acompanhamento de variáveis de impacto significativo no sistema, como por exemplo a composição inicial de alimentação mostrada na Figura 1b, o controlador pode atuar para ajustar os valores de entrada para os projetados, minimizando o desvio na variável controlada sem ter nenhuma informação sobre as condições de saída.

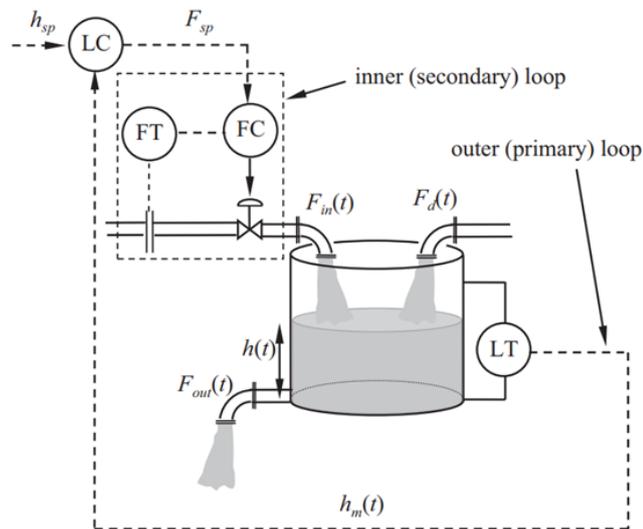
Figura 1 – Estratégias de controle *Feedback* e *Feedforward*



Fonte: Stephanopoulos (1984)

A estratégia em cascata é utilizada para aplicações que exigem uma abordagem mais robusta de controle e é realizada pela conexão hierárquica de dois controladores com duas variáveis de medida diferentes, como vista na Figura 2. O controlador primário usa a medição da variável de saída, normalmente a controlada (no caso da Figura 2, o nível), e atua com *setpoint* próprio. A saída do controlador primário é repassada ao secundário como um *setpoint* que somado com a medição de uma outra variável (no caso da Figura 2, a vazão de entrada) são utilizados pelo controlador secundário para decidir a atuação na variável manipulada.

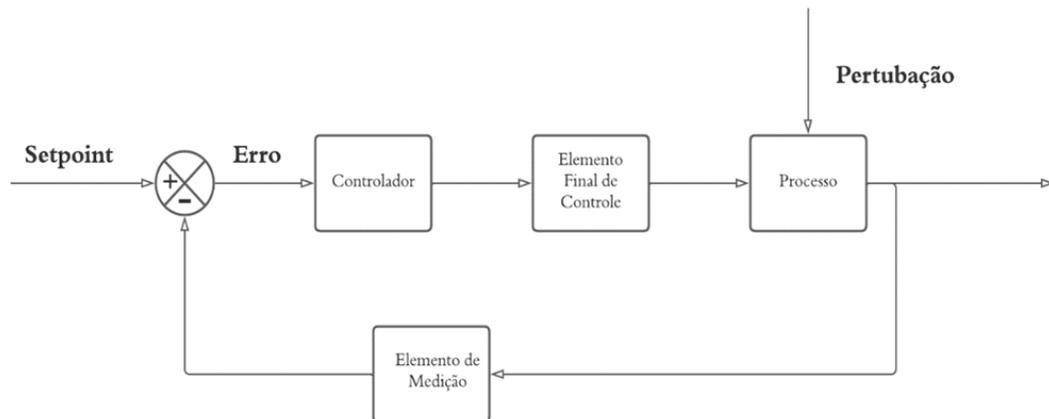
Figura 2 – Estratégia de controle cascata



Fonte: Kravaris e Kookos (2021),

Essas estratégias podem ser classificadas também em relação a quantidade de variáveis de entrada e saída. A partir dessa classificação os sistemas podem ser denominados como SISO (*Single Input Single Output*), sendo aqueles que têm uma única variável de entrada e saída, SIMO (*Single Input Multiple Output*) que têm uma entrada e múltiplas saídas, MISO (*Multiple Inputs e Single Output*) que têm múltiplas entradas e uma saída e, por fim, o que é considerado o mais complexo de todos o MIMO (*Multiple Input Multiple Output*) que têm múltiplas variáveis de entradas e saídas.

No âmbito da estratégia *feedback*, o diagrama de blocos utilizado para representar uma malha *feedback*, a estratégia utilizada neste trabalho, pode ser visualizada na Figura 3, e é composta em geral por quatro elementos: elemento de medição, controlador, elemento final de controle e o processo em si. A medição é responsável por aferir o valor da variável controlada do processo e enviar essa informação para ser comparada com o *setpoint*, que forma o sinal de erro, este sinal é repassado ao controlador que o interpreta e envia um sinal ao elemento final de controle para que ele atue no processo e assim reduza o valor do desvio.

Figura 3 – Diagrama de blocos do controle *Feedback*

Fonte: Autor (2024),

### 3.1.2 Estabilidade

Para a realização de qualquer estratégia de controle, no âmbito da abordagem clássica, é necessário o desenvolvimento de uma equação matemática derivada da modelagem do sistema. Esta equação é denominada função de transferência (FT) e visa representar o comportamento da variável de saída devido a alteração e perturbação do conjunto de variáveis de entrada. Em geral esta equação é uma razão de dois polinômios, como demonstrada na Equação 1 para a FT de um sistema de malha aberta, em que suas respectivas raízes são importantes para o estudo da estabilidade do sistema.

$$G(s) = \frac{Q(s)}{P(s)} \quad (1)$$

Um sistema dinâmico é considerado estável se a partir de qualquer que seja a alteração limitada no seu estado inicial, este produza uma saída também limitada, ou seja, que não divirja. Segundo Ogata (2010) esta capacidade de que a saída sempre retorne a um estado de equilíbrio para condições iniciais diversas, é a característica mais importante do comportamento dinâmico dos sistemas de controle.

As raízes dos polinômios  $Q(s)$  e  $P(s)$  são indispensáveis, principalmente a última, para a análise do comportamento do sistema quanto sua estabilidade. As raízes de  $Q(s)$  são denominadas de zeros do sistema e representam os pontos onde a FT é nula, enquanto as raízes de  $P(s)$  são chamadas de polos e representam os pontos em que a FT tende a valores infinitos. Os polos têm um papel fundamental, pois seus valores influenciam significativamente no comportamento dinâmico do sistema a medida que o tempo tende ao infinito. Os polos podem ter parte real

e imaginária, a parte imaginária sempre virá conjugada com a parte real e representará que o sistema dinâmico apresenta um comportamento oscilatório. A parte do polo que define a estabilidade do sistema é a parte real, se ela tiver valor negativo então o sistema será estável porque a resposta do sistema dinâmico tenderá a um valor nulo.

Portanto, um sistema só será estável se todos os seus polos, não importando a quantidade, tiverem suas respectivas partes reais negativas. Neste contexto, para analisar a estabilidade do sistema de malha fechada precisamos realizar a FT desta que levará em consideração os diferentes elementos da malha, como o processo ( $G_p(s)$ ), o elemento de medição ( $G_m(s)$ ), o elemento final de controle ( $G_f(s)$ ) e o controlador ( $G_c(s)$ ). De modo que ao associar todas as FT's da malha gera-se a Equação 2.

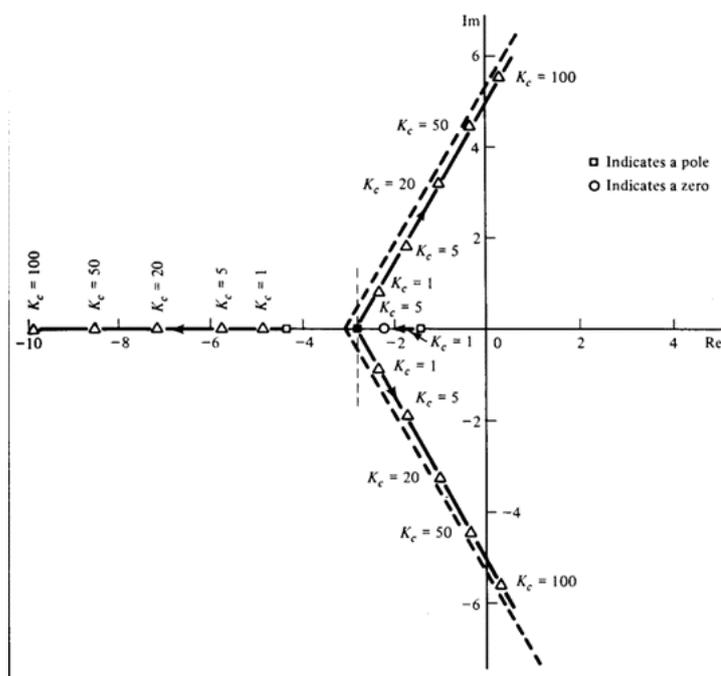
$$\bar{y}(s) = \frac{G_p(s)G_f(s)G_c(s)}{1 + G_p(s)G_f(s)G_c(s)G_m(s)}\bar{y}_{SP}(s) + \frac{G_d(s)}{1 + G_p(s)G_f(s)G_c(s)G_m(s)}\bar{d}(s) \quad (2)$$

Ao analisar a equação, pode-se perceber que a resposta do sistema,  $\bar{y}(s)$ , depende de dois termos, em que o primeiro está relacionado a alteração do *setpoint*,  $\bar{y}_{SP}(s)$ , e o segundo está relacionado as perturbações relacionadas a variáveis de entrada,  $\bar{d}(s)$ . Portanto, pode-se diferenciar o problema de controle de uma malha fechada em dois principais tipos, o servo, ou seja, baseado na mudança de *setpoint*, ou do tipo regulatório relacionado aos distúrbios da variável de entrada.

Porém, independentemente do tipo, é possível notar que o denominador de ambas FTs é idêntico e este é denominado equação característica do sistema. Então, de maneira análoga ao sistema de malha aberta às raízes da equação característica, os polos, permitem determinar a estabilidade dos sistemas de malha fechada.

Esta análise pode ser feita de maneira visual pelo método do local das raízes que determina valores para diversos parâmetros importantes da equação característica, como, principalmente, o ganho do controlador  $K_c$  (que será melhor definido posteriormente) plotando os resultados em um plano cartesiano, de eixos real e imaginário, para realizar a análise das parcelas reais dos polos.

Figura 4 – Diagrama do lugar das raízes



Fonte: Stephanopulos (1984)

Porém se a determinação dos valores dos polos for impossibilitada pela complexidade da equação característica, impedindo assim a análise visual do local das raízes, outros métodos podem ser utilizados para determinar a estabilidade do sistema sem realizar estes cálculos. Dentre esses métodos, o mais famoso é o de Routh-Hurwitz que permite determinar não os valores, mas sim a quantidade de polos que possuem parcelas reais positivas ou negativas, assim determinando a estabilidade do sistema. Isso é possível devido aos cálculos com os coeficientes do polinômio da equação característica na matriz de Routh. Nela a primeira coluna da matriz deve ter todos os valores, parcelas reais dos polos, positivos para que o sistema seja estável. Permitindo, desse modo, determinar os intervalos de parâmetros do controlador em que a estabilidade é garantida.

### 3.1.3 Tipos de controladores

O controlador, portanto, é o elemento que toma a decisão da intensidade da atuação que o sistema precisa receber para retornar ao valor do *setpoint*. No controle clássico, existem diversos tipos desde dos mais simples como o controlador *on-off* que trabalha de modo a ligar ou desligar o elemento final de controle a depender da necessidade, a exemplo do funcionamento de uma bomba no enchimento de um tanque que liga ao chegar em um limite mínimo e desliga ao chegar no máximo. Porém, outros tipos de controle atuam a depender de um sinal constante de erro, destes quatro principais variações são o controlador proporcional (P), proporcional-integral (PI),

proporcional-derivativo (PD) e proporcional-integral-derivativo (PID).

Como a própria denominação indica, o controlador P atua baseado no valor do erro retroalimentado, ajustando o esforço do controlador a partir de uma constante de proporcionalidade denominada de ganho do controlador ( $K_c$ ), como demonstrado na Equação 3 que representa sua FT em relação ao erro. Esta configuração têm como benefício uma atuação rápida e simples do controlador, porém não reduz o erro no novo estado estacionário a zero, assim tendo um valor chamado de *offset*. Este valor pode ser minimizado a medida que o valor do ganho aumenta, porém isso não pode ser feito de maneira descuidada, pois valores muito grandes do mesmo podem causar instabilidade no sistema e assim fazer com que ele obtenha um comportamento oscilatório que não tenda ao *setpoint*.

$$G_c(s) = K_c \quad (3)$$

Pensando neste problema, o controlador proporcional-integral (PI) foi criado com o intuito de adicionar uma ação integral que é baseada na magnitude do erro acumulado, representado pela integral da Equação 4. Desta forma, a minimização do erro só possível quando a soma do erro no tempo integral  $\tau_i$  é nula, ou seja o erro no estado estacionário (*offset*) seja zero. Desta maneira, ao aplicar Laplace a FT para esse controlador, o mesmo dependerá de um novo parâmetro, o ganho integral  $K_i$ , que é a razão de  $K_c$  por  $\tau_i$ , assim como representado pela Equação 5. Porém, mesmo reduzindo o *offset* a zero, a atuação baseada nesse tipo de erro pode ocasionar a demora na resposta do sistema, o que pode gerar um comportamento oscilatório. Dessa forma, a otimização dos parâmetros é novamente requerida, já que grandes valores de  $K_i$  aceleram a resposta do sistema, porém podem gerar instabilidade.

$$u(t) = K_c \left( e(t) + \frac{1}{\tau_i} \cdot \int e(t) dt \right) \quad (4)$$

$$G_c(s) = K_c + \frac{K_i}{s} \quad (5)$$

O controlador proporcional-derivativo (PD), representado pela FT da Equação 6, adiciona a ação derivacional ao erro, que é relativa a taxa de variação do erro no tempo derivativo ( $\tau_d$ ). Essa abordagem permite atuar sobre o problema apresentado pelo controle integral e assim reduzir a diminuição do tempo de resposta do sistema, ou seja, que o novo estado estacionário seja alcançado mais rapidamente.

$$G_c(s) = K_c(1 + \tau_d \cdot s) \quad (6)$$

O controlador proporcional-integral-derivativo (PID), representado pela FT da Equação 7, é o tipo de controlador mais utilizado devido a sinergia causada pela junção das características de todas as ações de controles citadas anteriormente permitindo que as desvantagens apresentadas por uns sejam mitigadas pelas vantagens de outros. Desse modo, essa abordagem gera um controlador veloz, sem *offset* e estável. Porém, como dito em outras ações de controle, neste caso os três parâmetros,  $K_c$ ,  $K_i$  e  $K_d$  (multiplicação de  $K_c$  por  $\tau_d$ ) devem ser bem otimizados para que sua aplicação não seja prejudicada.

$$G_c(s) = K_c + \frac{K_i}{s} + K_d \cdot s \quad (7)$$

### 3.1.4 Sintonia de controladores

A otimização destes parâmetros é um processo chamado de sintonia. Dentre os métodos clássicos, pode-se mencionar o de Ziegler-Nichols (ZN) que funciona tanto para sistemas em que a modelagem matemática deles permite uma solução analítica quanto para sistemas complexos em que é necessário recorrer a abordagens empíricas. Essa estratégia de sintonia possui duas metodologias que são baseadas na estimativa de constantes a partir do comportamento do sistema de controle, que serão utilizadas para calcular os parâmetros do controlador.

A primeira abordagem utiliza uma perturbação do tipo degrau unitário no sistema em malha aberta, e depende da curva resposta do sistema ter um formato de “S”. Caso isso seja contemplado, os parâmetros do PID podem ser aproximados baseados em dois valores chamados de atraso (L), que representa o tempo morto do sistema, e constante do tempo (T) encontrados pela tangente da curva no ponto de inflexão, como mostrado na Tabela 1 abaixo.

Tabela 1 – Estimativa dos parâmetros dos controlador por ZN, pelo primeiro método

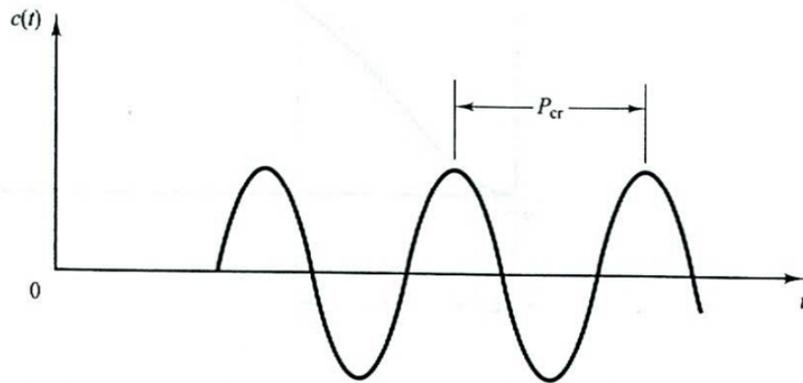
Controlador	$K_p$	$K_i$	$K_d$
P	$\frac{T}{L}$	-	-
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	-
PID	$1,2\frac{T}{L}$	$2L$	$0,5L$

Fonte: Adaptado de Ogata (2010)

Para o segundo método de ZN utiliza-se a equação de malha fechada, apenas usando a parte proporcional do controlador, e varia-se ela de 0 até que pela primeira vez, a saída alcance

uma oscilação sustentada como mostrada na Figura 5.

Figura 5 – Oscilação sustentada para método de ZN



Fonte: Ogata (2010)

Esse valor representa a constante chamado de ganho crítico ( $K_{cr}$ ). Após isso, o  $K_{cr}$  e o período dessa oscilação, chamado de período crítico ( $P_{cr}$ ), são utilizados para encontrar os parâmetros do controlador, como visto na Tabela 2 (Ogata, 2010).

Tabela 2 – Estimativa dos parâmetros dos controlador por ZN

Controlador	$K_p$	$K_i$	$K_d$
P	$0,5K_{cr}$	-	-
PI	$0,45K_{cr}$	$0,8333P_{cr}$	-
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

Fonte: Adaptado de Ogata (2010)

Além disso dos métodos clássicos, existem também aqueles baseados em técnicas de otimização a partir de uma função objetivo, como a própria função erro por exemplo. Dentre estas métricas estão as integrais do erro, que são: a Integral do erro absoluto (IAE); a Integral do erro quadrático (ISE); Integral do erro absoluto multiplicado pelo tempo (ITAE); e Integral do erro quadrático multiplicado pelo tempo (ITSE), visualizadas nas Equações abaixo, em que  $t$  é o tempo e  $e(t)$  é o sinal de erro.

$$IAE = \int |e(t)| dt \quad (8)$$

$$ITAE = \int t|e(t)| dt \quad (9)$$

$$ISE = \int [e(t)]^2 dt \quad (10)$$

$$ITSE = \int t[e(t)]^2 dt \quad (11)$$

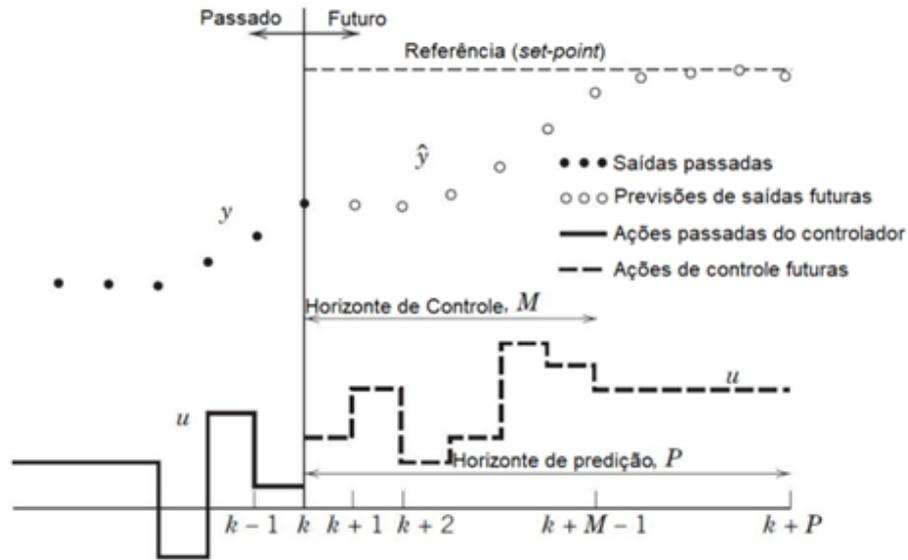
### 3.2 Controle avançado

O controle clássico é uma área conceituada e difundida na indústria principalmente devido ao desempenho dos controladores PID, porém é comum que os processos demonstrem condições que fujam de hipóteses ideais tornando a aplicação dessa metodologia e o ajuste dos parâmetros do PID uma tarefa complexa, especialmente para sistemas dinâmicos não-lineares (Schwenzer *et al.*, 2021). Com isso, houve o desenvolvimento de novas abordagens de controle, denominadas de controle avançado, que tiveram como objetivo solucionar as lacunas deixadas pela metodologia clássica, como por exemplo o controle adaptativo, o controle robusto e o controle preditivo com modelo (MPC).

O MPC é um conjunto de técnicas, originadas na década de 70, que utilizam de um modelo do sistema para antecipar as saídas do processo e atuar sobre ele de modo que pelo uso de uma função custo minimizar o desvio entre os valores futuros da variável controlada e o *setpoint* (Ren *et al.*, 2022). Desse modo, ele se destacou devido a sua eficiência e capacidade de trabalhar com sistemas complexos que apresentam características como restrições operacionais, sistemas multivariáveis (MIMO), incertezas, altos tempos mortos, não-linearidades entre outras (Pon Kumar *et al.*, 2018).

A estratégia do MPC é baseada na utilização de um modelo que preveja as saídas do processo,  $\hat{y}$ , para uma determinada quantidade, P, de tempos futuros chamado de horizonte de predição. Essas predições são baseadas em duas principais informações, as informações passadas do sistema, y, e as ações futuras de controle, u. Essas ações futuras de controle são calculadas a partir da otimização de uma função objetivo que visa manter  $\hat{y}$  mais próximo possível dos valores futuros desejados para y chamados de trajetória de referência, ou seja, o *setpoint*, em um determinado número de ações de controle denominado horizonte de controle, M. Esse processo pode ser visualizado na Figura 6, e os resultados são repassados para o sistema, e o processo é repetido até que as perturbações cessam (Camacho; Bordons, 2007).

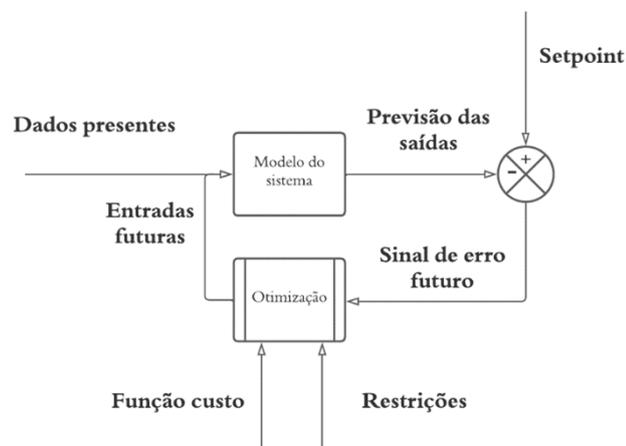
Figura 6 – Estratégia MPC



Fonte: Seaborg (2012)

Para que este resultado seja possível, a estrutura do MPC, visualizada na Figura 7, depende do algoritmo de otimização, principalmente, do modelo de predição de modo que o tipo de controlador é denominado a partir do modelo como por exemplo os MPC lineares (LMPC) e não lineares (NMPC). O modelo do sistema é a parte mais importante do MPC, pois a qualidade da previsão que será utilizada pela metodologia depende do quão bem o modelo descreve o sistema (Carvalho; Alvarez, 2019). Quanto a construção do modelo, pode ser realizada de duas maneiras principais, a fenomenológica e a determinística (empírica).

Figura 7 – Estrutura do MPC



Fonte: Autor (2024)

As abordagens fenomenológicas constroem o modelo seguindo rigidamente leis (matemá-

tica, físicas ou químicas) que descrevam o sistema estudado de modo a defini-lo com uma ou um conjunto de equações. Enquanto o segundo tipo lança mão de dados experimentais do processo para construir um modelo, que não segue nenhum tipo específico de lei do sistema. Como os modelos fenomenológicos são de difícil construção para sistema de alta complexidade, que podem conter inúmeros conjuntos de equações e parâmetros de difícil obtenção para descrevê-lo, os modelos determinísticos são mais utilizados pela sua capacidade de lidar de maneira simples com esses sistemas, já que não necessitam de nenhum tipo de informação prévia, dependendo apenas dos dados de entrada e saída dele, como por exemplo, no trabalho de Saraiva, Silva e Carvalho (2020) que utilizou modelos de aprendizado de máquina, como RNA e máquinas de vetores de suporte (redes de aprendizado de máquina baseadas na estatística) para controlar um sistema MIMO de produção de ciclopentadieno.

Em relação ao processo de otimização, ela é realizada para selecionar a melhor ação de controle que minimize o desvio da trajetória de referência. Para isso podem ser utilizados distintos métodos e funções objetivos (FO) a depender do tipo de MPC. A Equação 12, demonstra um exemplo típico de FO para MPC que relaciona no primeiro termo a discrepância entre os valores previstos e da trajetória de referência no horizonte de previsão ( $H_p$ ) e no segundo termo que representa o esforço do controlador no horizonte de controle ( $H_c$ ). Os termos  $\delta$  e  $\lambda$  são parâmetros proporcionais que relacionam a rapidez da ação de controle e a suavização do controlador (Saraiva, 2017).

$$J(N_1, H_p, H_c) = \sum_{j=N_1}^{H_p} \delta(j) [\hat{y}(t+j|t) - y(t+j)]^2 + \sum_{j=1}^{H_c} \lambda(j) [\Delta u(t+j-1)]^2 \quad (12)$$

Essa minimização, ou sintonia do controlador, pode ser realizada também pela otimização multiobjetivo de métricas de desempenho, como por exemplo as métricas ISE, IAE, ITSE e ITAE (Carvalho; Alvarez, 2019). Devido a todos os benefícios e possibilidades de arranjo que o MPC oferece, diversos estudos e aplicações industriais foram realizados, como por exemplo em plantas de energia solar, refinarias de açúcar, unidades de craqueamento catalítico, robótica, colunas de destilação, plantas de PVC, entre outros (Camacho; Bordons, 2007; Schwenzer *et al.*, 2021).

Com o avanço das aplicações da indústria 4.0, houve cada vez mais o interesse do uso dos modelos da inteligência artificial, principalmente nas subáreas de aprendizado de máquina e aprendizado profundo, no MPC para substituir os modelos fenomenológicos principalmente pela vantagem de fácil aplicação em processos complexos. Assim, a rede neural artificial (RNA), uma

famosa representante do aprendizado de máquina, foi bastante estudada para modelar sistemas de diversos processos e assim atuar como modelos do sistema no MPC como, por exemplo, na reação simples em um reator CSTR (Li; Jiang; Han, 2019), em uma coluna despropanizadora (Shin; Smith; Hwang, 2020) e em operações de espessamento de pastas (Núñez *et al.*, 2020).

O interesse na utilização das redes de aprendizado profundo como modelos internos de controladores (MPC) pode ser observado nos trabalhos de Wu, Rincon e Christofides (2020) que propuseram a utilização de uma rede neural recorrente (RNN) para controlar dois reatores CSTR não-isotérmicos acoplados em série, Santos (2020) que realizou o controle de um processo de produção do propileno glicol em um reator CSTR utilizando redes RNA, RNN, e duas outras redes chamadas de *Long-short term memory* (LSTM) e *Gated recurrent unit* (GRU), Zheng, Wang e Wu (2022) no controle de um processo de cristalização usando redes RNN, e Zarzycki e Ławryńczuk (2022) que compararam o uso de redes recorrentes do tipo LSTM e GRU no controle de dois processos químicos, reator de polimerização e um reator de neutralização.

### 3.3 Aprendizado profundo

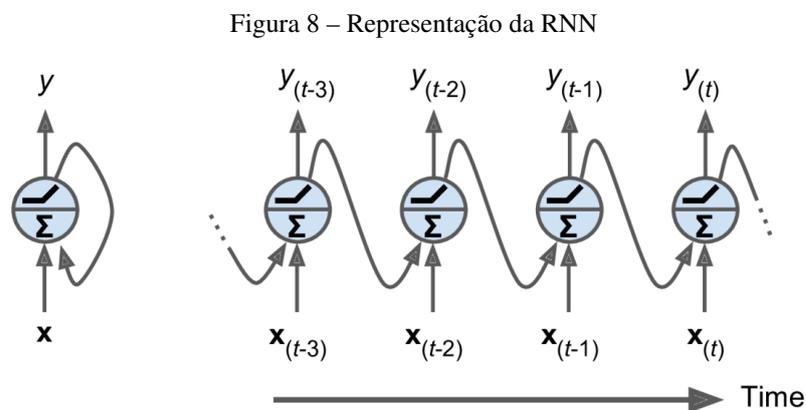
Como mencionado nos parágrafos acima, nas últimas décadas o aprendizado de máquina têm tido destaque nas mais diversas áreas, não somente relacionados ao controle de processos, mas também na engenharia, medicina e finanças (Frank; Drikakis; Charissis, 2020; Pallathadka *et al.*, 2023; Yadav, 2020). Este sucesso está relacionado a capacidade de suas redes de solucionarem problemas, como regressões e classificações, sem a necessidade de conhecimento prévio do sistema, simplesmente pela análise das relações de entrada e saída presentes nos dados.

A RNA, uma das técnicas mais difundidas do aprendizado de máquina, é um modelo computacional inspirado no funcionamento do cérebro humano. Ela é composta por uma camada de entrada que recebe os dados iniciais, pelas camadas intermediárias que realizam o processamento da informação e pela camada de saída que gera o resultado final da rede. Cada camada é composta por neurônios artificiais, que simulam as conexões existentes entre os neurônios naturais de modo que todos eles estão interligados e sua ativação, sinapse, depende de uma função de ativação que recebe como entrada a saída de outros neurônios ponderados por seus respectivos pesos. Portanto, a medida que o ajuste desses pesos é realizado a RNA consegue mapear as relações de entrada e saída e assim realizar seu aprendizado.

O aprendizado profundo é uma subclasse do aprendizado de máquina derivado do estudo de RNAs com diversas camadas intermediárias o que permitiu a habilidade de extrair, dos dados

disponíveis, representações hierárquicas mais complexas que o aprendizado de máquina é capaz de realizar. Vale salientar que essa capacidade permite a solução de problemas como regressão, processamento de linguagem natural, classificação, reconhecimento de imagens entre outros (Shrestha; Mahmood, 2019).

Dentre as técnicas de aprendizado profundo derivadas da RNA as RNN, representadas pela Figura 8, foram idealizadas para tratar dados sequenciais como séries temporais, textos, áudios, entre outros. Isto significa que sua arquitetura foi planejada para inserir na estrutura da RNA a capacidade de relacionar informações importantes com a variável tempo, ou seja, a criação de uma memória. Para isso, os dados de entrada da rede são inseridos de modo que contenham dados referentes a uma certa quantidade de tempos passados, chamados de *lag* ou atraso. Esses dados são processados conjuntamente com o resultado do processamento do tempo anterior, chamado de estado oculto ( $h$ ). E por fim, ao processar essas duas informações a rede pode atualizar o estado oculto com novas informações importantes e assim influenciar na resposta de tempos posteriores ( $y(t)$ ,  $y(t+1)$ ,  $y(t+2)$ , etc.), simulando assim a memória humana (Shrestha; Mahmood, 2019).

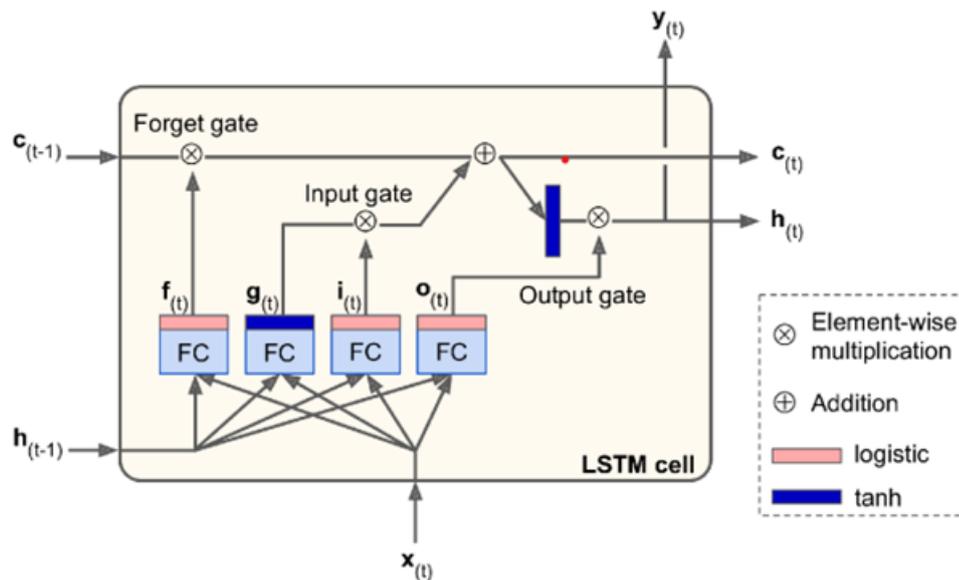


Porém, a RNN têm problemas ao lidar com séries muito longas devido a um efeito chamado *vanishing gradient* em que informações em processamentos atuais acabam possuindo uma influência maior que informações iniciais, que mesmo se importantes são esquecidas. Para solucionar os resultados insatisfatórios provocados por esse efeito foram criadas variações da RNN que possuem mecanismos para criar uma memória capaz de armazenar essas informações relevantes (Pon Kumar *et al.*, 2018; Santos, 2020).

### 3.3.1 LSTM

A rede LSTM é uma variação da RNN que visa contornar o problema do *vanishing gradient* ao acrescentar na arquitetura existente estruturas capazes de gerenciar uma memória de longo prazo. A célula LSTM, Figura 9, é composta por dois tipos de estado, e três tipos de portão que são mecanismos de controle do fluxo de informação (Shrestha; Mahmood, 2019).

Figura 9 – Representação da LSTM



Fonte: Géron (2019).

Os dois estados presentes na LSTM são o estado oculto ( $h$ ), que já estava presente na RNN, que corresponde a memória de curto prazo e o estado da célula ( $c$ ) que armazena a memória de longo prazo, ou seja a memória que retém informações importantes do início ao final da série (Géron, 2019).

Em relação aos portões, o portão de esquecimento ( $f_t$ ), descrito pela Equação 13, é a etapa responsável por determinar quais informações do estado da célula antigo se tornaram obsoletas, e assim atrapalhando o processo de aprendizado. Essa identificação ocorre através de uma função de ativação *sigmoid* ( $\varphi$ ) que interpreta os dados de entrada atuais ( $x_t$ ) e o estado oculto passado ( $h_{t-1}$ ) multiplicados pelos seus respectivos pesos ( $W_{xf}, W_{hf}$ ) e somado com um *bias*. O resultado dessa função é um vetor de valores entre 0 e 1 que significam, respectivamente, esquecer e lembrar. Esse vetor formado, posteriormente é multiplicado ao estado da célula para que assim possa atualiza-lo.

$$f_t = \varphi(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (13)$$

O portão de entrada ( $i_t$ ), representado pela Equação 14, portanto é responsável por determinar quais informações apresentadas pelos dados atuais e pelo estado oculto anterior são relevantes o suficiente para serem passadas para a memória de longo prazo. De modo análogo ao processo anterior, porém com a adição da Equação 15 que através da função de ativação tangente hiperbólica é responsável por selecionar os possíveis candidatos a serem inseridos no estado da célula.

$$i_t = \varphi(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (14)$$

$$g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \quad (15)$$

Por fim, portão de saída ( $o_t$ ) é responsável por analisar os dados de entrada e estado oculto e assim definir quais dados são necessários passar adiante como a resposta do sistema (ou novo estado oculto). Essa resposta do sistema, representada na Equação 18, é a multiplicação entre o vetor resultante do portão de saída, Equação 16, pelos candidatos selecionados a partir do novo estado da célula, Equação 17 (Géron, 2019; Van Houdt; Mosquera; Nápoles, 2020).

$$o_t = \varphi(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (16)$$

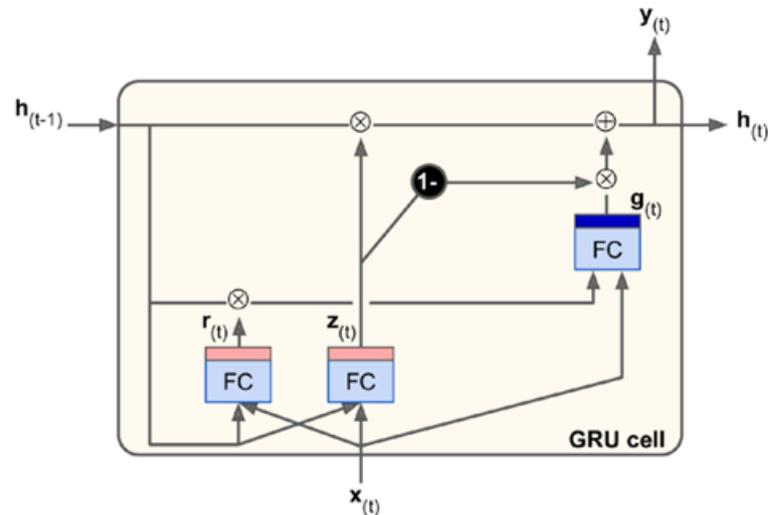
$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \quad (17)$$

$$y_t = h_t = o_t \cdot \tanh(c_t) \quad (18)$$

### 3.3.2 GRU

A rede GRU, Figura 10, é uma simplificação da rede LSTM que têm a capacidade de performar, em muitos problemas, tão bem quanto ela, porém em menos tempo (Géron, 2019; Tang *et al.*, 2020).

Figura 10 – Representação da GRU



Fonte: Géron (2019)

As principais simplificações realizadas foram a união dos dois estados anteriores para um único estado oculto ( $h_t$ ) e a simplificação dos três portões para o portão de *reset* ( $r_t$ ) e *update* ( $z_t$ ).

O portão de *reset* é responsável por determinar a quantidade do estado oculto anterior que será utilizado para o cálculo do novo estado proposto, como visto na Equação 20. Se o resultado deste portão for próximo de 0 o novo estado será calculado principalmente pelos dados de entrada, porém se for mais próximo de 1, pelos dados do antigo estado oculto.

$$r_t = \varphi(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (19)$$

$$g_t = \tanh(W_{xg}x_t + W_{hg}(h_{t-1} \cdot r_t) + b_g) \quad (20)$$

O portão *update* controla quanto do estado anterior será mantido ou substituído pelo novo estado proposto, fazendo o papel dos portões de esquecimento e entrada da LSTM. Vale salientar que, como mostrado na Equação 22, se o valor de  $z_t$  for próximo de 0, isso significa que ele atua para esquecer a informação passada pelo estado anterior e focar na informação fornecida pelo novo estado proposto. Enquanto se ele estiver próximo de 1 isso indica que deve manter a maior parte do estado anterior (Géron, 2019; Shrestha; Mahmood, 2019; Tang *et al.*, 2020).

$$z_t = \varphi(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (21)$$

$$y_t = h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot g_t \quad (22)$$

### 3.4 VMD

Para um bom funcionamento das técnicas de aprendizado de máquina e aprendizado profundo é necessário que os dados fornecidos tenham uma boa qualidade. A decomposição de sinais é uma técnica que têm como base a divisão de um sinal em um conjunto de outros sinais que representam características específicas do original. Uma aplicação direta desta técnica é o tratamento de sinais realizado a partir da retirada do ruído. Isso acontece pela extração das frequências que compõe o sinal original, retornando apenas as informações importantes para detecção de falhas. Esse tipo de aplicação pode ser utilizada, por exemplo, na detecção de falhas (Zhang *et al.*, 2020).

A VMD (*Variational Mode Decomposition*) é uma técnica de decomposição que foi recentemente proposta por Dragomiretskiy e Zosso (2014) e vem tendo destaque na separação de sinais complexos e ruidosos devido as inovações propostas na construção de seu algoritmo. O sucesso da VMD se dá pela sua abordagem variacional, que realiza uma otimização da decomposição do sinal original em sinais centrados em frequências próprias, chamados modos, ou IMF's (*Intrinsic Mode Functions*). Esta otimização é feita para que a soma desses IMFs reconstrua o sinal original de maneira fiel, porém sem o ruído, sendo assim uma alternativa viável para o tratamento de sinais não-lineares e não-estacionários (Liu *et al.*, 2023).

Essa abordagem permite a VMD ser mais eficiente e resolver problemas recorrentes para outros algoritmos de decomposição como a sobreposição de sinais presente no EMD (*Empirical mode decomposition*) (Wei *et al.*, 2021; Zhang *et al.*, 2020). Essa capacidade torna esse modelo muito utilizado para realização do pré-tratamento de sinais complexos, e por isso, a VMD vem chamando atenção em estudos relacionados a modelos híbridos, ou seja, que utilizam mais de um tipo de algoritmo distinto para realização de uma tarefa. Exemplos de aplicações desse algoritmo em modelos híbridos são, por exemplo, modelos híbridos de decomposição-predição que utilizam do tratamento de sinais para melhorar a qualidade dos dados fornecidos para os algoritmos de previsão (Li *et al.*, 2021; Song *et al.*, 2021).

Porém, a maior limitação da utilização deste técnica, é que a capacidade dela de desconstruir o sinal corretamente está intimamente atrelada com a escolha adequada dos seus parâmetros, principalmente aqueles aos quais Dragomiretskiy e Zosso (2014) não sugerem valores padrões. Esses parâmetros são a quantidade de IMFs em que o sinal vai ser decomposto ( $K$ ) e o parâmetro  $\alpha$ . Um ajuste inadequado deles têm um impacto significativo na decomposição, pois caso  $K$  tenha um valor inferior ao adequado, o sinal será dividido em menos IMFs do que o necessário, o que

ocasionará que as informações serão dispersas em outros modos ( $\alpha$  baixo), ou descartadas por serem classificadas como ruído ( $\alpha$  elevado). De maneira contrária se  $K$  é maior do que deveria, causará problemas incorporando informação demais aos sinais, ou seja absorvendo o ruído ( $\alpha$  pequeno), ou causando a multiplicação de modos ( $\alpha$  grande) (Dragomiretskiy; Zosso, 2014).

Portanto, a sintonia desses parâmetros é de extrema importância, e diversos estudos foram realizados buscando alternativas para a otimização desses valores ( Firdaus *et al.*, 2019). Entre elas, pode-se citar a utilização do algoritmo genético para otimizar de uma vez ambos parâmetros.

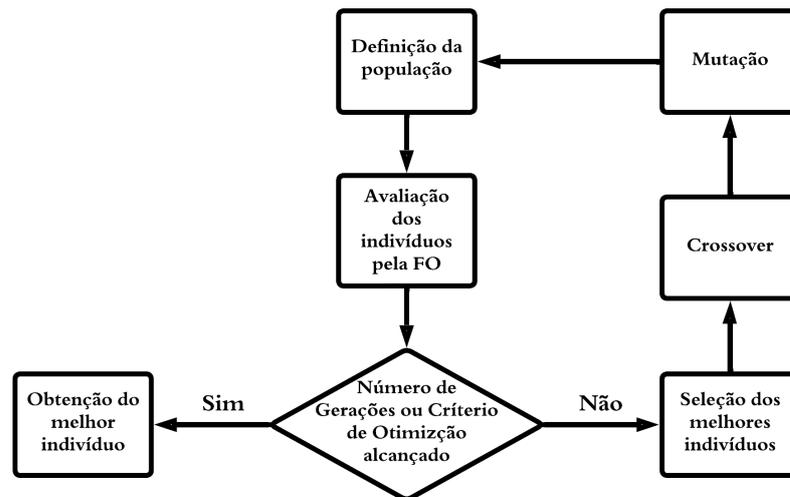
### 3.5 Algoritmo genético

A otimização é um processo que visa encontrar uma solução ótima para um problema dentre um espaço amostral vasto, algumas vezes sendo até infinito (Pereira *et al.*, 2022). Na grande maioria dos problemas abordados pela engenharia, encontrar a solução não é simples pela existência de diversos pontos no espaço de busca denominados de mínimos/máximos locais. Para problemas complexos, esses pontos referem a boas soluções em suas cercanias, porém não se aproximam da solução máxima/mínima global (a solução ótima). Portanto, muitos métodos clássicos enfrentam dificuldades ao tratar desses problemas, tendo que recorrer a um esforço computacional que não os tornam práticos ( Langazane; Saha, 2022).

Neste contexto, os métodos de otimização meta-heurísticos se tornam atrativos pela sua característica de explorar o espaço amostral da solução através de um método que utiliza as próprias soluções exploradas como pontos de partidas para determinar outros tipos de soluções. A depender de como se dá essa metodologia, os métodos de otimização meta-heurísticos podem ser classificados naqueles baseados na evolução, em fenômenos físicos, comportamento relacionados a seres humanos ou naqueles inspirados em enxames (Pereira *et al.*, 2022).

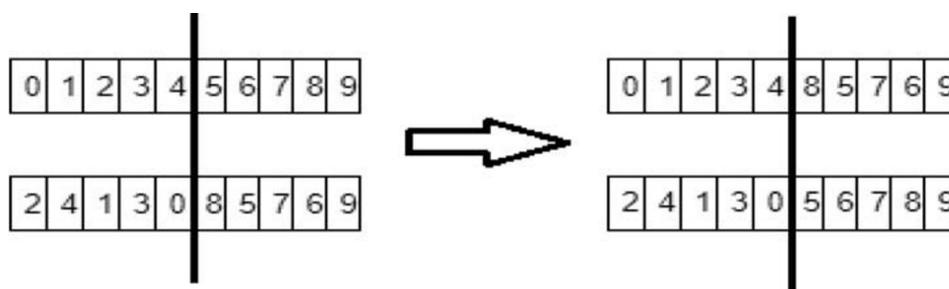
Dentre os algoritmos evolutivos, o mais famoso é o algoritmo genético que é um método de otimização inspirada na seleção natural de Charles Darwin, em que os indivíduos que possuem os genes que lhes dão características que os beneficiem no ambiente em que estão, têm mais chances de sobreviver e perpetuar esses genes. A estrutura desse algoritmo pode ser visualizada no diagrama disposto na Figura 11.

Figura 11 – Estrutura do algoritmo genético



Fonte: Autor (2024)

A primeira etapa deste processo é a definição da população de indivíduos, em que cada indivíduo corresponde a um gene que carrega consigo as informações das variáveis que serão otimizadas no processo, ou seja, as possíveis soluções. Após isso, cada indivíduo é avaliado pela função objetivo e pelo método de seleção que pode ocorrer de diversas maneiras, como por exemplo roleta, *ranking* e torneio. Ao ser selecionado o mesmo passa pela etapa de *crossover*, representado na Figura 12, que é a etapa responsável por gerar os novos indivíduos da próxima geração, essa reprodução pode ser realizada por diferentes métodos como por exemplo, um-ponto, dois-pontos e k-pontos (Katoch; Chauhan; Kumar, 2021).

Figura 12 – Método de *crossover* de um-ponto

Fonte: (Katoch; Chauhan; Kumar, 2021)

Por fim, a mutação dos indivíduos é gerada para inserir uma diversidade genética na população e permitir que o algoritmo saia de máximos/mínimos locais e explore outras soluções possíveis além daquelas que tiveram uma alta probabilidade de cruzamento nas gerações

anteriores. E então este processo é realizado até que o número de gerações definido seja alcançado ou que a função objetivo atinja algum critério de otimização.

As aplicações deste método são diversas abrangendo campos como por exemplo engenharia, processamento multimídia (imagens e vídeos), agricultura de precisão, jogos e localização (Katoch; Chauhan; Kumar, 2021). Mais focado no escopo deste trabalho é possível citar a sua aplicação na otimização de parâmetros importantes, como *lag*, para redes GRU e LSTM (Bouktif *et al.*, 2020; Dhake; Kashyap; Kosmopoulos, 2023; Surakhi *et al.*, 2021), para a otimização dos valores de  $\alpha$  e  $K$  na VMD (Wang *et al.*, 2022) e na otimização de modelos híbridos, como no caso de Huang *et al.* (2021) que realizou a otimização dos parâmetros da VMD, e após isso usou os dados decompostos de uma série temporal do preço de fechamento de ações como entradas para uma rede LSTM.

### 3.6 Processo estudado

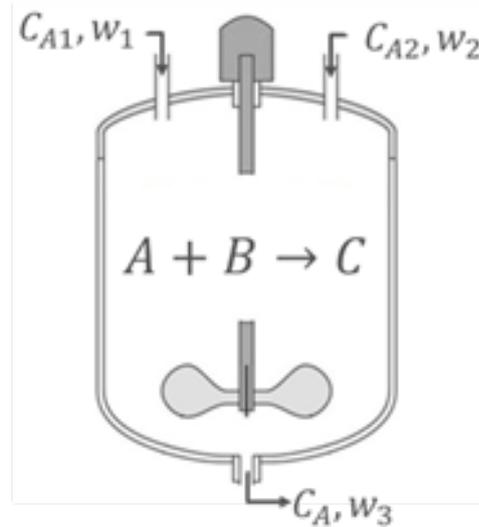
As estratégias de controle desenvolvidas neste trabalho foram utilizadas para controlar a concentração de saída da hidrogenação catalítica de etileno a etano, Equação 23. Esta reação é uma reação de cinética não-linear, Equação 24, comum para reações catalíticas assim como descrito em Sussman e Potter (1954).



$$r_a = \frac{k_1 \cdot c_a}{(1 + k_2 \cdot c_a)^2} \quad (24)$$

Para este sistema a reação foi realizada em um reator do tipo tanque de mistura perfeita (CSTR) isotérmico, representado na Figura 13 como uma reação genérica  $A + B$  formando  $C$ , em que  $A$  representa o etileno,  $B$  hidrogênio e  $C$  o etano. O sistema escolhido têm como configuração duas linhas de entrada (1 e 2) com vazões representadas por  $w_1$  e  $w_2$  e uma de saída com vazão  $w_3$  controlada por uma válvula.

Figura 13 – Reator CSTR para o sistema estudado.



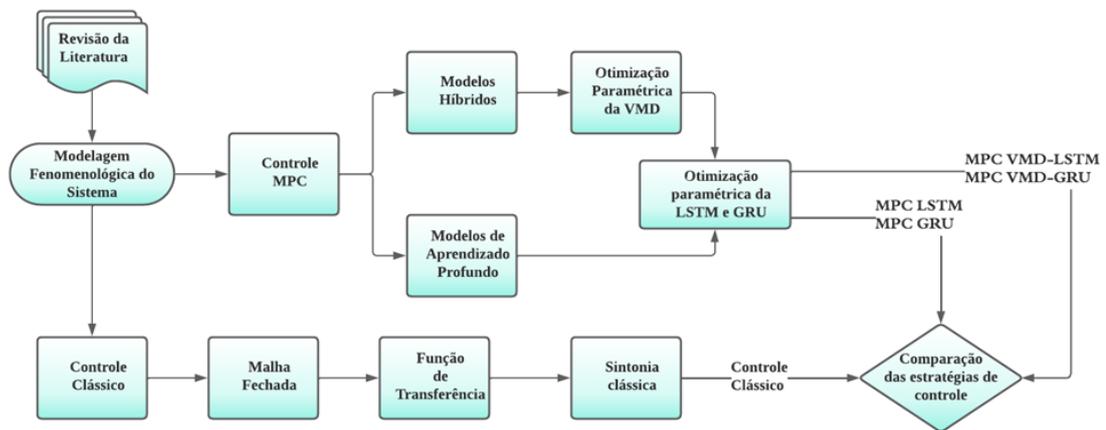
Fonte: Silva (2020).

A característica não linear do processo catalítico incentivou diversos estudos de controle que trataram de aspectos, como a estabilidade da reação, ao ser realizada por diferentes reatores (Matsuura; Kato, 1967), diferentes estratégias de controle e a influência causada pela restrição de variáveis (Li; Biegler, 1988), a utilização de controle preditivo de múltiplos passos (Bregel; Sieder, 1989) e o uso de técnicas de aprendizado de máquina como modelo para o controle preditivo (Silva, 2020).

## 4 METODOLOGIA

A metodologia realizada neste trabalho é expressa no fluxograma da Figura 14 e detalhes de cada passo são desenvolvidos durante as próximas subseções.

Figura 14 – Fluxograma de metodologia do trabalho.



Fonte: (Autor,2024)

O presente trabalho foi realizado a partir da construção de rotinas computacionais programadas em linguagem *Python*. Esta linguagem de programação é bastante popular devido a facilidade de uso e o fato de ser uma linguagem de código aberto. Isso permite que o *Python* seja utilizado livremente para diversas aplicações. Assim, os usuários (pessoas físicas, órgãos e empresas) podem ter acesso a diversas funcionalidades que são disponibilizadas aos demais como bibliotecas. Dentre as inúmeras, este trabalho utiliza, por exemplo, a *Control* para controle clássico, *Keras* para a programação das redes de aprendizado profundo, *Pandas* e *Numpy* para o tratamento de manipulação de dados, *Matplotlib* para geração de gráficos e modelagem e *Odeint* para a simulação de sistemas dinâmicos com a solução de EDOs.

### 4.1 Modelagem e simulação

A modelagem do problema foi realizada baseado no processo de hidrogenação catalítica do etileno, descrito na seção 3.6. Este foi escolhido por ter, como apresentado por Sussman e Potter (1954), uma cinética não-linear, Equação 24, típica para reações catalíticas.

Para modelar as equações do sistema, foram realizadas três principais considerações simplificadoras. A primeira é que o sistema é isotérmico, a segunda delimita que a mistura no

reator CSTR é perfeita e por fim supõe-se que os parâmetros, como densidade e capacidade calorífica, permaneçam constantes ao decorrer da reação.

#### 4.1.1 Obtenção das equações do sistema

Primeiramente foi realizado o balanço de massa global do sistema, representado pela Equação 25.

$$\begin{aligned} \text{Acúmulo} &= \text{Entrada} + \text{Saída} - \text{Consumo} \\ \frac{dV}{dt} &= w_1 + w_2 - w_3 \end{aligned} \quad (25)$$

A partir da Equação 25 podem ser feita duas considerações. A primeira está relacionada com a vazão de saída ( $w_3$ ). Esta é definida pela equação da válvula, ou seja, é dependente da constante da válvula ( $Cv$ ) e da altura de líquido no sistema. A segunda é que a área do reator é constante, e igual a  $10 \text{ m}^2$ , deste modo pode-se deixar a equação em termos da única variável, a altura, ao substituir  $V = A \cdot h$ .

Essas considerações resultam na equação diferencial ordinária (EDO) visualizada na Equação 26 que representa a variação da altura de líquido no sistema.

$$\begin{aligned} \frac{d(A \cdot h)}{dt} &= w_1 + w_2 - Cv\sqrt{h} \\ \frac{dh}{dt} &= \frac{w_1 + w_2 - Cv\sqrt{h}}{A} \end{aligned} \quad (26)$$

De maneira análoga para o balanço de massa do etileno temos que:

$$\begin{aligned} \frac{dN_a}{dt} &= w_1 \cdot c_{a1} + w_2 \cdot c_{a1} - w_3 \cdot c_a - r_a \cdot V \\ \frac{dN_a}{dt} &= w_1 \cdot c_{a1} + w_2 \cdot c_{a1} - Cv\sqrt{h} \cdot c_a - r_a \cdot V \end{aligned} \quad (27)$$

Em que  $N_a$  é o número de mols de A,  $c_a$  é a concentração de saída de a e  $r_a$  é cinética definida por Sussman e Potter (1954) na Equação 24. Considerando que o número de mols pode ser definido como a multiplicação da concentração com o volume temos que:

$$\begin{aligned} \frac{dN_a}{dt} &= w_1 \cdot c_{a1} + w_2 \cdot c_{a1} - Cv\sqrt{h} \cdot c_a - \frac{k_1 c_a}{(1 + k_2 c_a)^2} \cdot V \\ \frac{d(A \cdot h \cdot c_a)}{dt} &= w_1 \cdot c_{a1} + w_2 \cdot c_{a1} - Cv\sqrt{h} \cdot c_a - \frac{k_1 c_a}{(1 + k_2 c_a)^2} \cdot h \cdot A \\ \frac{dh}{dt} \cdot c_a + \frac{dc_a}{dt} \cdot h &= \frac{w_1 \cdot c_{a1}}{A} + \frac{w_2 \cdot c_{a2}}{A} - \frac{Cv\sqrt{h} \cdot c_a}{A} - \frac{k_1 c_a}{(1 + k_2 c_a)^2} \cdot h \end{aligned} \quad (28)$$

Aplicando o valor da Equação 26 na Equação 28, obtemos a Equação 29 para a EDO que representa a variação de  $c_a$ .

$$\frac{dc_a}{dt} = \frac{w_1(c_{a1} - c_a)}{h \cdot A} + \frac{w_2(c_{a2} - c_a)}{h \cdot A} - \frac{k_1 c_a}{(1 + k_2 c_a)^2} \quad (29)$$

Deste modo, pode-se perceber que para realizar a simulação dinâmica do sistema é preciso resolver simultaneamente o conjunto de EDOs, Equações 26 e 29, devido a dependência da variação de  $c_a$  com a altura. A técnica utilizada para resolver a integração deste conjunto de EDOs foi o método numérico de Runge-Kutta de 4ª Ordem.

#### 4.1.2 Definição do ponto de operação do sistema

Neste sentido, para que se possa realizar a simulação, é necessário obter os valores dos parâmetros das EDOs ( $A, C_v, k_1, k_2$ , entre outros) como também as condições iniciais para a simulação.

Os parâmetros neste trabalho foram adaptados do trabalho de Brengel e Sieder (1989) e podem ser visualizados na Tabela 3. Dentre esses parâmetros, a vazão  $w_1$  é particularmente importante, pois realizará a função de variável manipulada no controle da concentração de saída.

Tabela 3 – Parâmetros utilizados no trabalho

Parâmetro	Valor
A	10 m <sup>2</sup>
$k_1$	1 s <sup>-1</sup>
$k_2$	1 L/mol
$c_{a1}$	24,9 mol/L
$c_{a2}$	0,1 mol/L
$w_1$	0,1 L/s
$w_2$	0,1 L/s
$Cv$	0,2 L/( $\sqrt{h} \cdot s$ )

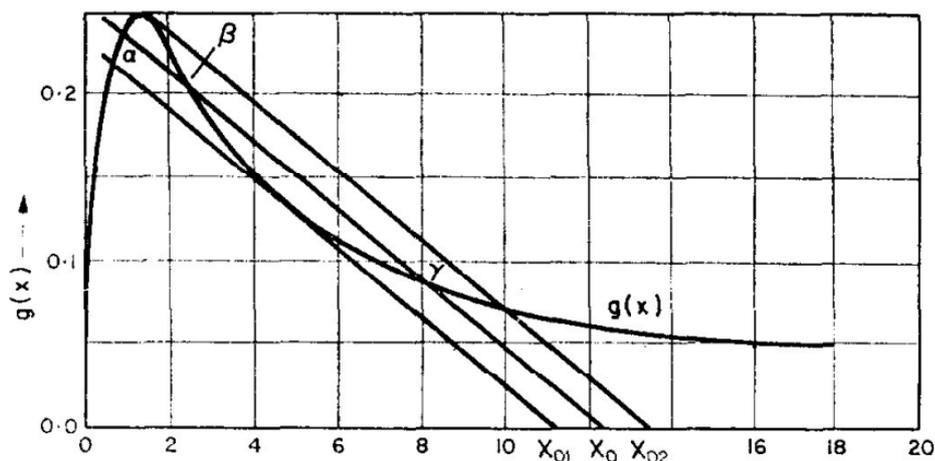
Fonte: Brengel e Sieder (1989)

A condição inicial do sistema também foi definida baseado no trabalho de Brengel e Sieder (1989) que utilizam do trabalho de Matsuura e Kato (1966) para definir pontos de

estabilidade e instabilidade do processo, inclusive para o sistema hidrogenação catalítica utilizado neste trabalho.

Esta metodologia alcança isto através da igualdade do regime difusional com o reacional para diferentes valores de concentração. Destacando os pontos em que os mesmos se igualam, como representado na Figura 15.

Figura 15 – Pontos de equilíbrio para a reação de hidrogenação catalítica de etileno.



Fonte: Matsuura e Kato (1966)

Dentre os três pontos visualizados na Figura 15, diferentes para cada linha, têm-se que  $\alpha$  e  $\gamma$  são pontos denominados estáveis e o  $\beta$  chamado de instável. O ponto  $\beta$  é denominado assim pois o sistema tende a algum dos outros dois pontos de equilíbrio a partir dele (Matsuura; Kato, 1966).

Portanto, neste trabalho o ponto  $\alpha$  foi escolhido como estado estacionário de partida, e assim como Brengel e Sieder (1989), o ponto  $\beta$  foi definido como o *setpoint* para o valor de  $c_a$ . Os pontos  $\alpha$ ,  $\beta$  e  $\gamma$  encontrados neste trabalho foram respectivamente 0,6326, 2,7926 e 7,0746 mol/L.

## 4.2 Controle clássico

Neste contexto, a estratégia de controle clássica utilizada para solução deste problema foi a *feedback*. O sistema SISO deste trabalho tratou do controle da concentração de etileno ( $c_a$ ) na corrente de saída do reator utilizando a vazão de entrada na corrente 1 ( $w_1$ ) como variável manipulada.

Conforme a revisão bibliográfica, na seção 3.1.2, para realizar a ação do controle clássico é necessário a obtenção da FT do processo estudado. Porém, o conjunto de EDOs do sistema,

Equações 26 e 29, detêm termos não lineares. Isto é um problema para o controle clássico e assim é necessário que estas equações sejam linearizadas primeiramente.

#### 4.2.1 Linearização

Este processo foi realizado pela aplicação da série de Taylor, que pode ser descrita como o somatório de uma sequência de infinitos termos que aproximam uma função desejada entorno de um ponto arbitrário. Para a definição da FT, é necessário que este ponto seja calculado a partir das variáveis desvios, aplicadas para o valor do estado estacionário determinado na seção 4.1.2 como mostrado na Equação 30.

$$H = h - h_s; \quad C_a = c_a - c_{as}; \quad W_1 = w_1 - w_{1s} \quad (30)$$

Como as EDOs dependem de mais de uma variável ( $c_a, w$  e  $h$ ) deve-se realizar uma linearização multivariável, que nada mais é que a adição de derivadas parciais desses termos na série de Taylor. Ao definir a EDO de  $h$  como  $f_h(h, w_1)$  e a de  $c_a$  como  $f_c(c_a, h, w_1)$  obtém-se as seguintes equações.

$$\frac{df_h}{dt} = \frac{\partial f_h}{\partial h} \cdot H + \frac{\partial f_h}{\partial c_a} \cdot C_a + \frac{\partial f_h}{\partial w_1} \cdot W_1 \quad (31)$$

$$\frac{df_c}{dt} = \frac{\partial f_c}{\partial h} \cdot H + \frac{\partial f_c}{\partial c_a} \cdot C_a + \frac{\partial f_c}{\partial w_1} \cdot W_1 \quad (32)$$

Portanto, a partir da escolha do estado estacionário, definido como  $c_{as} = 0,6326$  mol/L,  $w_{1s} = 0,1$  L/s e  $h_s = 1$  m, pode-se realizar os cálculos das derivadas parciais das Equações 31 e 32 com os valores citados para obter as Equações 33 e 34.

$$\frac{df_h}{dt} = -0,1 \cdot H + W_1 \quad (33)$$

$$\frac{df_c}{dt} = -0,2373 \cdot H - 0,1044 \cdot C_a + 2,4267 \cdot W_1 \quad (34)$$

#### 4.2.2 Cálculo da função de transferência

Uma vez em posse das equações linearizadas, é possível realizar a aplicação da transformada de Laplace para obtenção das FTs. A primeira a ser calculada foi a da Equação 33, isto porque como não há uma dependência da EDO em relação a variável  $c_a$  ( $\frac{\partial f_h}{\partial c_a} = 0$ ) ela permite que as variáveis desvio H e W sejam relacionadas diretamente, Equação 35.

$$\frac{H(s)}{W_1(s)} = \frac{1}{s + 0,1} \quad (35)$$

Com a Equação 35 à disposição, foi aplicada Laplace na Equação 32. Ao reorganizar os termos e substituir o valor encontrado da Equação 35 obtém-se a seguinte equação para FT do sistema:

$$\frac{C_a(s)}{W_1(s)} = \frac{2,4267 \cdot s + 0,0053}{s^2 + 0,2043 \cdot s + 0,0104} \quad (36)$$

#### 4.2.3 Estudo da estabilidade do sistema de malha fechada

Após a obtenção da FT do sistema em malha aberta, pode-se realizar a análise da malha fechada como exposto na seção 3.1.2. Primeiramente, como o problema estudado é baseado na alteração do *setpoint* da concentração do sistema, o mesmo é classificado como do tipo servo, e como os distúrbios,  $\bar{d}(s)$ , não são levados em consideração, a Equação 2 toma a seguinte forma.

$$\frac{\bar{y}(s)}{\bar{y}_{SP}(s)} = \frac{G_p(s)G_f(s)G_c(s)}{1 + G_p(s)G_f(s)G_c(s)G_m(s)} \quad (37)$$

Pode-se observar que na Equação 37 consta quatro FTs distintas que precisam ser definidas. A FT do processo,  $G_p(s)$ , foi definida na etapa anterior, Equação 36. Já a FT do controlador,  $G_c(s)$ , depende do tipo de controlador (P, PI, PID) implementado como visto na seção 3.1.3.

No caso deste estudo, as funções de transferência do sistema de medição,  $G_m(s)$ , e do elemento final de controle,  $G_f(s)$ , foram definidas da seguinte forma.

Primeiramente foi considerado que as FTs que as descrevem sejam sistemas de primeira ordem assim como visualizado nas equações abaixo.

$$G_m(s) = \frac{K_m}{\tau_m \cdot s + 1} \quad (38)$$

$$G_f(s) = \frac{K_f}{\tau_f \cdot s + 1} \quad (39)$$

Em que  $K_m$  e  $K_f$  são os ganhos de medição e de elemento final. Neste trabalho é suposto que ambos sejam iguais a 1, e que  $\tau_m$  e  $\tau_f$  sejam as constantes de tempo para os respectivos sistemas de medição e atuação. Portanto, supondo-se que estes sistemas são significativamente mais velozes que a resposta do processo, ou seja, que os valores de  $\tau_m$  e  $\tau_f$  sejam próximos de zero, podemos supor que as respectivas FTs sejam:

$$G_m(s) = K_m \quad (40)$$

$$G_f(s) = K_f \quad (41)$$

Com todas as FTs devidamente definidas, pode-se realizar o estudo da estabilidade do sistema de malha fechada a partir da solução da equação característica, Equação 42.

$$0 = 1 + G_p(s)G_f(s)G_c(s)G_m(s) \quad (42)$$

O estudo de estabilidade realizado neste trabalho contou com a aplicação de Routh para analisar os intervalos de valores de  $K_c$  em que o sistema é estável. Além da visualização pelo método do lugar das raízes.

#### 4.2.4 Sintonia dos controladores

Ao obter o intervalo de valores de  $K_c$  que garantam a estabilidade do sistema, pode-se realizar o processo de sintonia dos controladores utilizados. Para isso, foram testados dois métodos, o primeiro foi o ZN, baseando-se nos valores  $K_{cr}$  e  $P_{cr}$  encontrados a partir do método de Routh e avaliado a partir das integrais do erro.

O outro método aplicado foi o de otimização meta-heurística por algoritmo genético. Neste, foram utilizados os parâmetros descritos na Tabela 4 para realizar a minimização das quatro integrais do erro a partir dos valores dos ganhos correspondentes para cada controlador.

Tabela 4 – Parâmetros utilizados na otimização por algoritmo genético.

Parâmetro	Valor
População	200
Gerações	50
Probabilidade de cruzamento	90 %
Probabilidade de mutação	10%

Fonte: Autor (2024)

#### 4.3 Controle avançado

Como dito anteriormente, o tipo de controle avançado utilizado neste trabalho é o MPC. Esta técnica foi aplicada com dois tipos de modelo interno, divididos em modelos de aprendizado profundo e modelos híbridos.

Neste contexto foram utilizados na primeira categoria os modelos LSTM e GRU, que gerou os controladores denominados MPC-LSTM e MPC-GRU. Enquanto para os modelos híbridos foi realizada a interação entre o modelo de decomposição VMD e os modelos de aprendizado profundo, que gerou os controladores MPC-VMD-LSTM e MPC-VMD-GRU.

#### 4.3.1 Geração de dados

Independente das categorias escolhidas, os modelos utilizados necessitam de dados representativos para poderem aprender de maneira adequada. Para isto, foi realizada a simulação do sistema, devido a impossibilidade de adquirir dados reais do processo.

Os dados foram simulados solucionando as EDOs obtidas no item 4.1.1 pelo método numérico de Runge-Kutta de 4ª ordem. Para que fosse o mais representativo possível, foi realizada uma série de perturbações na variável manipulada (vazão) para avaliar como a variável controlada responderia a isto.

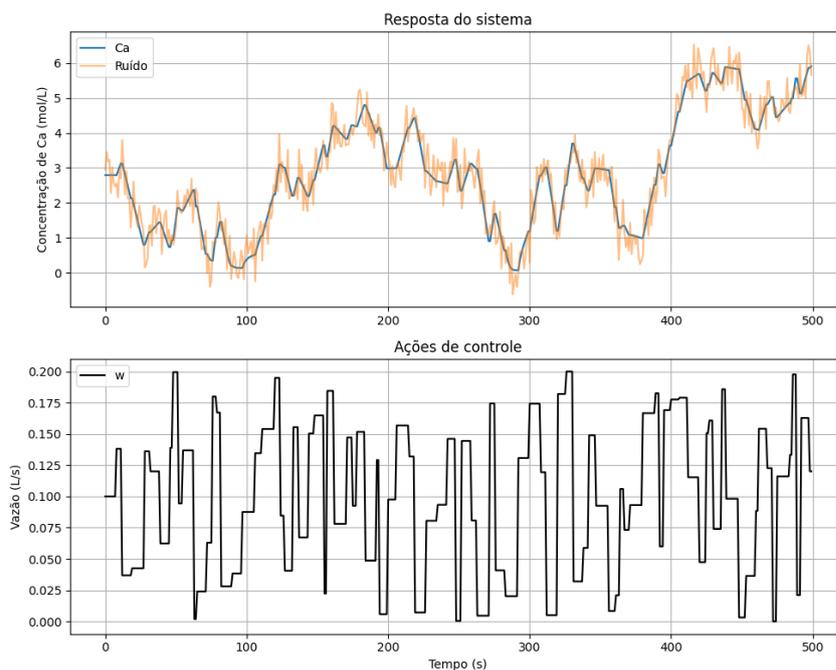
As perturbações na variável manipulada ocorreram entre 0 e 0,6 L/s e foram realizadas pelo método PRNBS (*Pseudo-random non binary signal*) que realiza perturbações com valores aleatórios de  $w$  dentro do intervalo escolhido, que têm a duração também aleatória entre dois e dez segundos.

Dessa forma foram simulados 450 mil dados contendo as informações de tempo, vazão e concentração. Porém, nem todos os dados obtidos refletem exatamente o valor da variável em uma medição real no processo. Isto pois fatores como flutuações aleatórias das variáveis do processo ou interferência externa na medição podem mascarar o verdadeiro valor. Saber lidar com ruído presente nos dados é um fator importante. Visto que a depender da sensibilidade do sistema de controle, o ruído pode influenciar demais na escolha das ações de controle e assim deixar as respostas instáveis (Woolf, 2024).

Sob esta perspectiva, a VMD se mostra uma ferramenta interessante, dado a sua capacidade de filtrar o ruído do sinal original (Jing *et al.*, 2023). Para investigar essa capacidade aplicada ao MPC, foi inserido no conjunto de dados simulados um ruído branco. Esse tipo de ruído é um tipo de sinal estocástico independente do tempo, e que em geral possui média zero, variância constante e obedece geralmente a uma distribuição normal. Portanto, o ruído foi inserido ao somar o valor original a uma série de somas e subtrações de valores aleatórios (Woolf, 2024).

Neste contexto, uma amostra dos dados obtidos tanto da simulação direta do sistema, quanto da inserção dos erros pode ser visto na Figura 16.

Figura 16 – Simulação dos dados de entrada das redes.



Fonte: Autor (2024)

#### 4.3.2 Treinamento das redes

Uma vez que os dados do sistema estejam simulados, eles precisam passar por transformações, denominadas de pré-tratamento, para aumentar o desempenho do treinamento das redes.

A primeira etapa a ser realizada é a separação do *dataset* em dois conjuntos, um para treino e outro para teste. Isso evita que o modelo tenha acesso aos dados que serão utilizados para avaliá-lo, e assim que tenha tendências a memorizá-los. Para modelos adaptados a séries temporais, como a LSTM e a GRU, essa separação é feita sequencialmente. Neste trabalho foi utilizada a divisão de 80% dos dados para treino e 20% dos dados para teste.

Após isso, foi realizado o processo de normalização dos dados. Neste caso, a normalização utilizada foi a *StandardScaler* que transforma os dados de maneira que obedecem a uma distribuição normal centrada em zero e de variância unitária.

Ao fim do pré-tratamento, a rede pode ser enfim treinada. No treinamento, essas redes ajustam parâmetros internos para poderem se adaptar aos padrões inerentes nos dados, e assim, aumentarem seu desempenho de previsão. Porém, existem outros parâmetros, chamados de hiperparâmetros, que impactam na eficiência das redes e que não são ajustados no processo de treino.

Neste trabalho, os principais hiperparâmetros que foram ajustados da LSTM e GRU foram o número de células, a quantidade de camadas, a função de ativação, a taxa de aprendizado e o *lag*. A metodologia de ajuste utilizada neste trabalho foi a *RandomSearch* da biblioteca *Keras Tuner*. Ela realiza uma sequência de testes com os valores de cada variável escolhidos de maneira randômica, dentro de um espaço de busca pré-determinado. Os intervalos utilizados para cada uma das variáveis estão dispostos na Tabela 5.

Tabela 5 – Hiperparâmetros utilizados no *RandomSearch*

Hiperparâmetro	Intervalo/Opções
Nº de células	Entre 10 a 150
Nº camadas intermediárias	1 ou 2
<i>Learning rate</i>	Entre $1 \cdot 10^{-4}$ a $1 \cdot 10^{-2}$
<i>Lag</i>	Entre 1 a 7
Função de ativação	Sigmoid, tanh ou relu

Fonte: Autor (2024)

Durante o processo de ajuste, o método divide aleatoriamente os dados de treino, reservando uma parte para avaliação, chamada de dados de validação. O objetivo dessa separação é evitar o viés que ocorreria se os dados de teste fossem utilizados no processo de ajuste. Ao final de cada etapa de treinamento, o modelo prevê os valores correspondentes aos dados de validação e os compara com os valores reais que deveriam ser alcançados. Por fim, o conjunto de variáveis com melhor resultado é selecionado, e assim utilizado para treinar com o *dataset* completo.

Após isso, este modelo é utilizado para prever os valores dos dados de teste. Neste trabalho a avaliação do resultado das previsões foi feita pelas métricas RMSE, MAE e o  $R^2$ .

$$RMSE = \sqrt{\frac{\sum_{t=1}^T (y - \hat{y})^2}{N}} \quad (43)$$

$$MAE = \frac{\sum_{t=1}^T |y - \hat{y}|}{N} \quad (44)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (\bar{y} - \hat{y})^2}{\sum_{i=1}^n (y - \hat{y})^2} \quad (45)$$

O processo realizado para o treinamento dos modelos híbridos, VMD-LSTM e VMD-GRU, é análogo. A única diferença é a adição da etapa de tratamento de ruído na etapa de pré-tratamento. E para fazer isso de maneira eficiente é necessário ajustar dois principais parâmetros da VMD, o número de IMFs (K) e o  $\alpha$ .

Este trabalho seguiu a metodologia utilizada por Huang *et al.* (2021), que utilizou o algoritmo genético para realizar esse ajuste. Neste trabalho os intervalos utilizados para K e  $\alpha$  são respectivamente 2 a 5 e 1000 a 10000. A função objetivo utilizada neste método é o erro de reconstrução do sinal Equação 46, em que N é o número de dados,  $S_N[n]$  é o sinal original e  $S'_N[n]$  é o sinal reconstruído .

$$FO = \sqrt{\frac{\sum_{t=1}^T (S_N[n] - S'_N[n])^2}{N}} \quad (46)$$

#### 4.3.3 Aplicação do MPC

Desta forma, os modelos treinados são utilizados como modelos internos dos seguintes controladores MPC-LSTM, MPC-GRU, MPC-VMD-LSTM e MPC-VMD-GRU. Da mesma maneira que o controle clássico, eles foram utilizados para o controle servo SISO da concentração, ao utilizar a vazão como variável manipulada.

A estrutura do MPC é formada basicamente pela otimização, pelo modelo e pela resposta do sistema. O método de otimização utilizado neste trabalho para realizar a escolha das ações de controle foi o algoritmo genético. Este tinha como entrada o estado anterior do sistema e o modelo de previsão e tinha como saída as vazões que melhor ajustavam o sistema ao *setpoint*. Neste trabalho, foi utilizado um AG com 5 gerações e com uma população de 200.

A resposta do sistema, foi obtida a partir das EDOs utilizadas na geração dos dados, visto que não foi possível utilizar protótipos para estimar a resposta real referente as ações de controle propostas.

Da mesma maneira que ocorre com os controladores clássicos, a sintonização dos parâmetros do MPC é essencial para que o controlador possa gerar ações de controle adequadas. Neste sentido, os principais parâmetros de sintonia do MPC vem da função objetivo expressa na Equação 12, sendo o  $H_p$ ,  $H_c$ ,  $\delta$  e  $\lambda$ . Deste modo, é fundamental entender como a alteração dessas variáveis altera o funcionamento do controlador, assim como investigar se existe alguma interação entre parâmetros de eficiência do modelo, como o *lag*, e os do MPC.

Para isso foi realizada um planejamento experimental  $2^6$  com as seguintes variáveis: tipo de modelo,  $lag$ ,  $H_p$ ,  $H_c$ ,  $\delta$  e  $\lambda$ . Os valores dos intervalos para cada uma das variáveis estão descritos pela Tabela 6. Vale salientar que o valor de  $H_c$  deve ser menor ou igual que o de  $H_p$ , desse modo foi retirado qualquer configuração que desrespeitasse essa condição. Portanto, apenas 48 dos 64 ensaios iniciais foram realizados. A comparação desses resultados foi feita pela análise individual como também pela média dos indicadores descritos anteriormente: IAE, ITAE, ISE e ITSE.

Tabela 6 – Níveis do planejamento experimental

Variável	Nível baixo	Nível alto
$Lag$	1	7
$H_p$	2	7
$H_c$	1	7
$\delta$	0,1	100
$\lambda$	0,1	100
Modelo	LSTM	GRU

Fonte: Autor (2024)

Após isso, os melhores ensaios obtidos são utilizados para analisar o impacto da inserção da ruído no processo de controle. Para isso, primeiramente estes modelos foram treinados com dados ruidosos e após isso seus desempenhos foram avaliados.

Ao fim deste processo, foi realizada a substituição do modelo interno pelo modelo híbrido, que utilizou a VMD para filtrar o ruído dos dados que então foram usados para treinar o modelo de aprendizado profundo.

## 5 RESULTADOS E DISCUSSÃO

Nesta seção, são apresentados e discutidos os principais resultados obtidos da aplicação dos passos descritos na metodologia.

A seção 5.1 apresenta a construção, escolha e desempenho do controlador clássico entre as três opções descritas na seção 3.1.3 (P, PI e PID). A seção 5.2 descreve o estudo da utilização das duas categorias de controladores avançados escolhida, o que utiliza apenas os modelos de aprendizado profundo (LSTM e GRU) e aqueles que utilizam os modelos híbridos (VMD-LSTM e VMD-GRU).

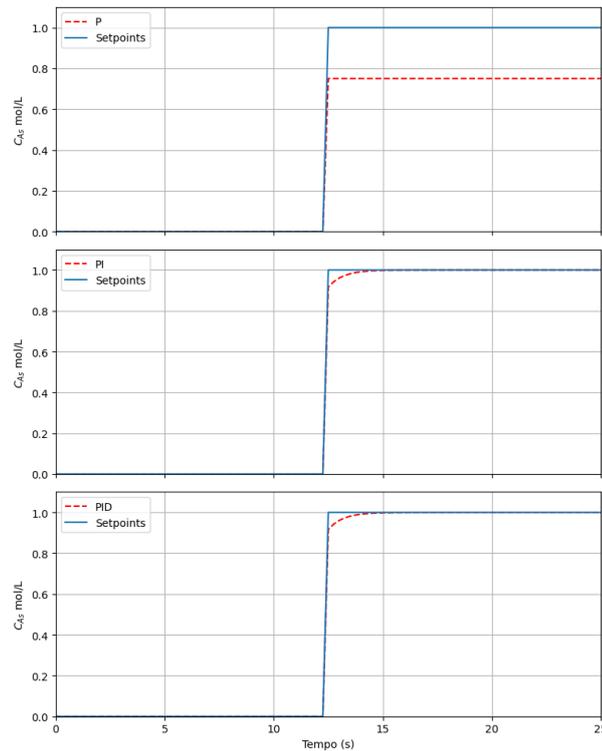
Por fim, a seção 5.2.3 analisa de maneira mais detalhada os resultados obtidos nas seções 5.1 e 5.2 e realiza a comparação de todos os controladores. Esta é dividida nos casos em que os controladores foram expostos ou não a dados ruidosos.

### 5.1 Controle clássico

Conforme descrito na metodologia deste trabalho, a FT do sistema de malha fechada é definida pela Equação 37 e é composta por quatro funções de transferência. A função  $G_p(s)$  é definida pela Equação 36, enquanto  $G_m(s)$  e  $G_f(s)$  definidos como 1. Por fim, assim como exposto na seção 3.1.3, a função  $G_c(s)$  dependerá do tipo de controlador a ser utilizado.

Neste sentido, a escolha do tipo de controlador (P, PI ou PID), impacta significativamente no comportamento da malha fechada do sistema. Para proporcionar uma visão geral do desempenho de cada tipo de controlador, foram realizadas perturbações no *setpoint* da variável desejada, a fim de avaliar as respostas obtidas. O desempenho de cada controlador pode ser observado na Figura 17. Vale salientar que foram atribuídos valores arbitrários para  $K_c$ ,  $K_i$  e  $K_d$  sendo todos iguais a 10.

Figura 17 – Comparação dos controladores



Fonte: Autor (2024)

Ao analisar a Figura 17, é possível notar que as respostas dos controladores PI e PID foram similares. Ambos controladores conseguiram eliminar o erro no novo estado estacionário do sistema. Em contraste, o controlador P apresenta um *offset*, o que é esperado, uma vez que é uma de suas características inerentes.

Considerando essas diferenças de desempenho, é necessário melhorar a escolha dos ganhos ( $K_c$ ,  $K_i$  e  $K_d$ ). Desta forma, a primeira etapa para fazer uma escolha adequada é o estudo de estabilidade do sistema.

Primeiramente foi analisado o método de Routh aplicado ao sistema com controlador proporcional. A equação característica para esse sistema está disposta na Equação 47 abaixo.

$$s^2 + (2,4267 \cdot K_c + 0,2044)s + 0,0053 \cdot K_c + 0,01043 = 0 \quad (47)$$

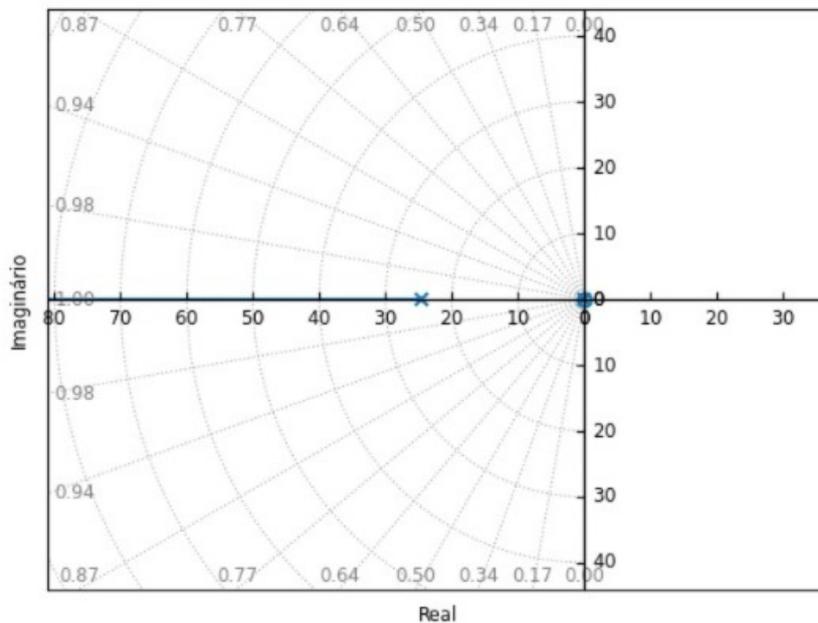
Aplicando o método de Routh para esta equação característica obtém-se a matriz representada pela Equação 48, em que para que o sistema seja considerado como estável, todos os valores da primeira coluna devem ser positivos.

$$\begin{bmatrix} 1 & (0,0053 \cdot K_c + 0,01043) & 0 \\ (2,4267 \cdot K_c + 0,2044) & 0 & 0 \\ (0,0053 \cdot K_c + 0,01043) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (48)$$

Portanto, ao resolver as inequações referentes aos termos da primeira coluna, encontra-se que o valor de  $K_c$  deve ser maior que -0,08 para que o sistema seja estável. Nota-se que o  $K_c$  não obteve nenhuma restrição quanto ao seu valor máximo, indicando que o aumento exagerado do  $K_c$  não causaria oscilações duradouras no sistema.

Além da análise das inequações, outra forma de avaliar a estabilidade do sistema é por meio dos gráficos do lugar das raízes. O sistema de controlador proporcional, representado pela Figura 18, apresentou um zero,  $z_1 = -0,0022$  e dois polos,  $p_1 = -24,69$  e  $p_2 = -2,603617 \cdot 10^{-3}$ , todos situados na região negativa do eixo dos reais, o que cumpre os critérios para estabilidade do sistema.

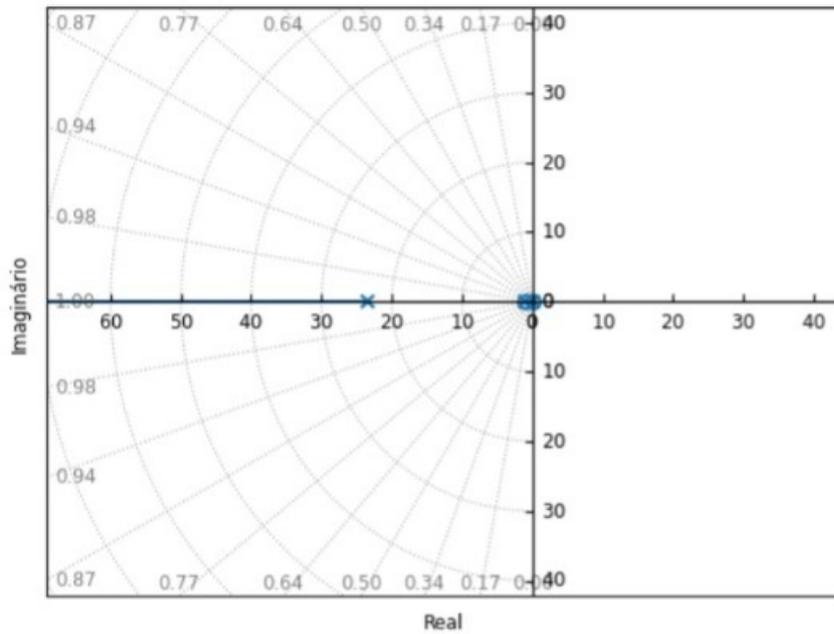
Figura 18 – Lugar das raízes para controlador proporcional



Fonte: Autor (2024)

O sistema de controle PI, representado pela Figura 19, apresentou dois zeros,  $z_1 = -1$  e  $z_2 = -0,0022$  e três polos  $p_1 = -23,433$ ,  $p_2 = -1,036$  e  $p_3 = -2,1941 \cdot 10^{-3}$ . Como todos os polos estão presentes no lado negativo dos reais, o sistema também é estável.

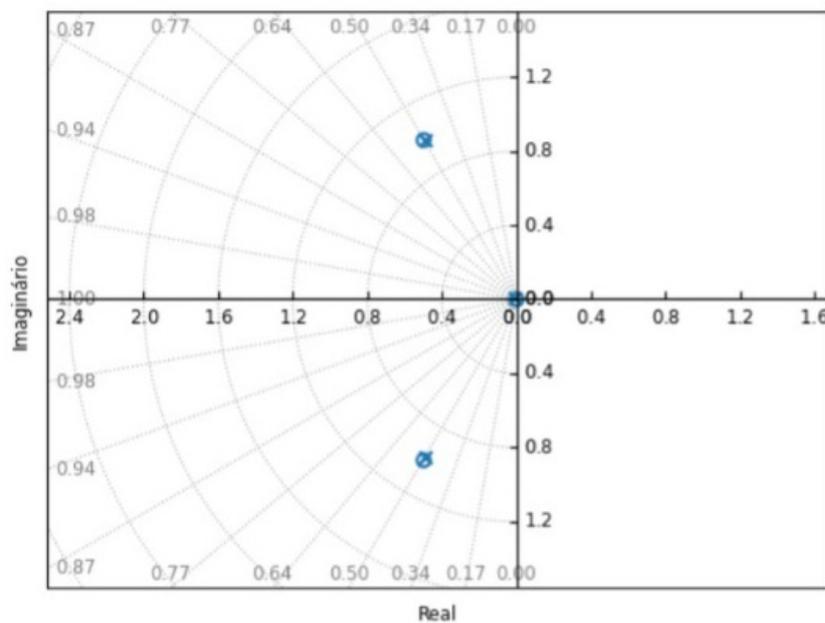
Figura 19 – Lugar das raízes para controlador proporcional-integral



Fonte: Autor (2024)

Por fim, o sistema de controle PID, representado pela Figura 20, apresentou comportamento semelhante ao controle PI, com três zeros  $z_1 = -0,5 + 0,8660 j$ ,  $z_2 = -0,5 - 0,8660 j$ ,  $z_3 = -0,0021$  e três polos, também todos com parcela real negativa,  $p_1 = -0,4842 + 0,8522 j$ ,  $p_2 = -0,4842 - 0,8522 j$ ,  $p_3 = -0,00219$ . E assim como os outros apresentou um comportamento estável.

Figura 20 – Lugar das raízes para controlador proporcional-integral-derivativo



Fonte: Autor (2024)

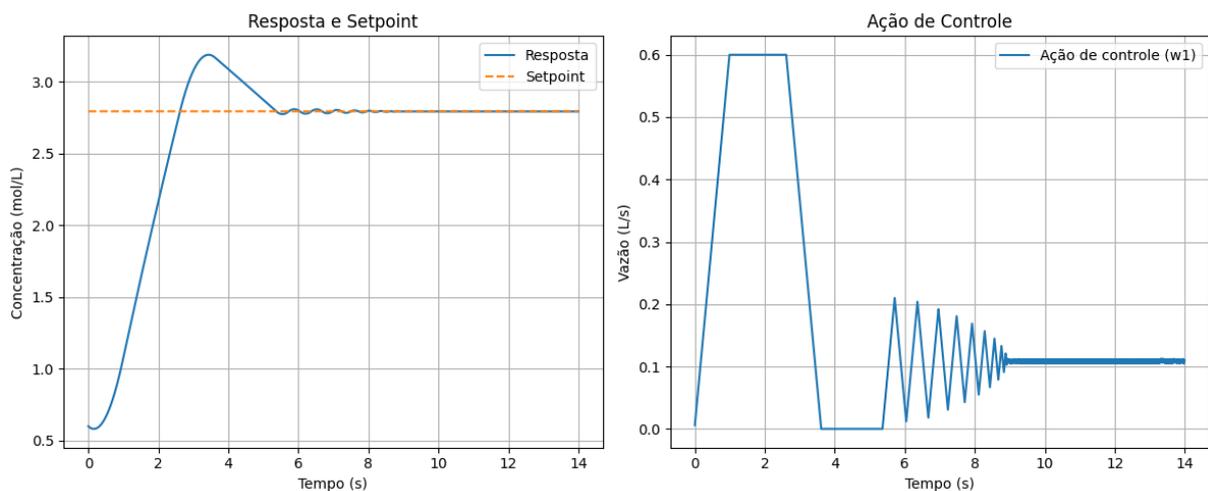
No caso estudado, percebeu-se que  $K_p$  não está restrito, desse modo poderia-se aumentar o seu valor para agilizar a resposta do sistema a perturbação realizada. Entretanto, o aumento indiscriminado do ganho, mesmo que reduza o valor do erro no estado estacionário, ainda não é algo recomendável já que eleva proporcionalmente o esforço do controlador e pode adicionar instabilidade ao sistema (Ogata, 2010). Neste contexto, o processo de sintonia dos parâmetros é essencial para aplicação correta do controlador.

Entretanto, como visto, a malha de controle não apresenta um comportamento oscilatório duradouro com o aumento de  $K_c$ . Assim, a versão do método ZN que necessita de um  $K_{cr}$  que imponha um estado oscilatório no sistema não foi aplicada. Além disso, como o sistema não foi simulado com tempo morto, o outro método ZN também foi impossibilitado.

Assim, para obter os melhores parâmetros dos controladores foi necessário recorrer a outra metodologia. Neste trabalho, o método escolhido foi a otimização meta-eurística com algoritmo genético. Isto, pois é um método de conhecimento do autor, que inclusive é utilizado neste trabalho. Desse modo, foi feita a variação dos parâmetros ( $K_c$ ,  $K_i$ ,  $K_d$ ) em um intervalo escolhido de maneira arbitrária entre 0 a 200.

Portanto, foi realizada uma otimização para cada tipo de controlador. Para o controlador P, o AG definiu  $K_c = 153,23$  como melhor valor. Nota-se que o valor encontrado é consideravelmente alto, muito provável para proporcionar uma convergência rápida ao *setpoint*. Porém, como pode ser visto na Figura 21, mesmo que este valor tenha diminuído o *offset*, também causou certa oscilação no processo de convergência.

Figura 21 – Resposta do controlador P com  $K_c = 153,23$

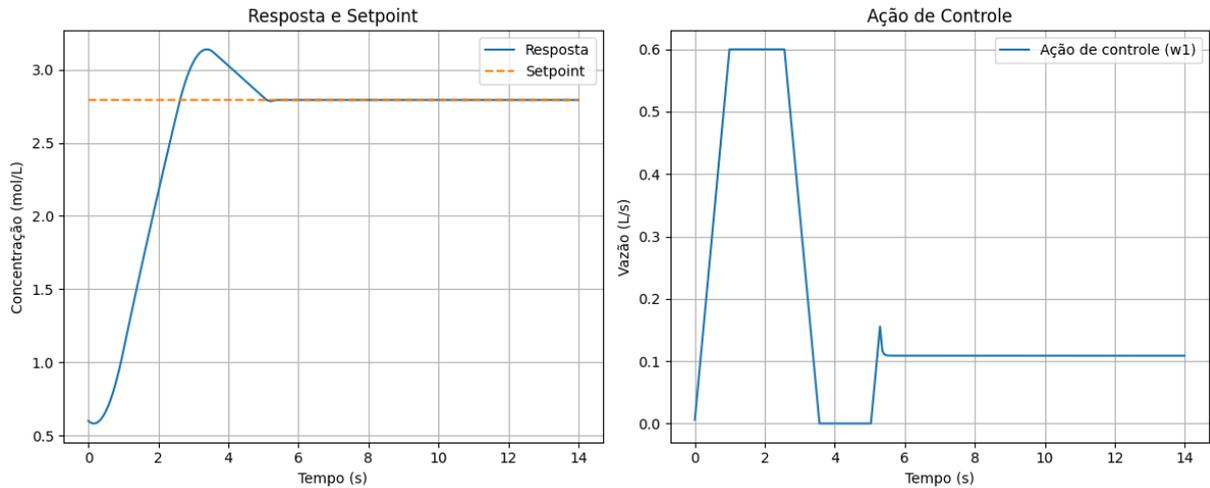


Fonte: Autor (2024)

Em relação ao controlador PI, o AG definiu os seguintes parâmetros:  $K_c = 9,623$  e

$K_i = 0.0368$ . Nota-se pela Figura 22, que as oscilações são suprimidas, e que com a inserção da ação integrativa o valor de  $K_c$  caiu drasticamente. Porém, o *overshoot*, parcela da resposta que ultrapassa o *setpoint* antes de se estabilizar, não foi eliminado.

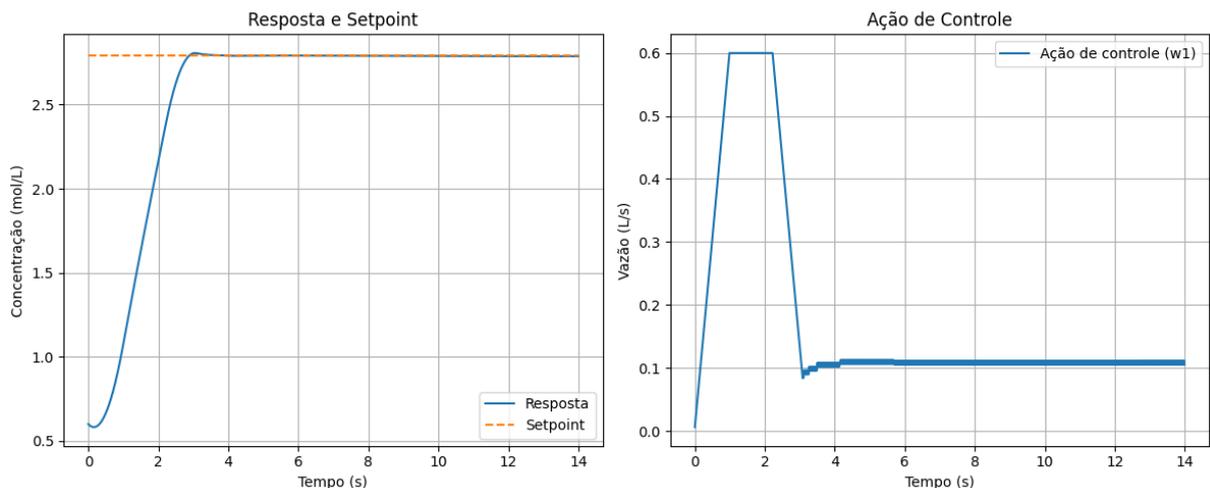
Figura 22 – Resposta do controlador PI com  $K_c = 9,623$  e  $K_i = 0.0368$



Fonte: Autor (2024)

Para o controlador PID, o AG delimitou os seguintes valores  $K_c = 53,2882$  e  $K_i = 0$ ,  $K_d = 20,2567$ . Ao avaliar a Figura 23, nota-se a robustez desse tipo de controlador, já que ele foi capaz de eliminar tanto o *overshoot* quanto as oscilações.

Figura 23 – Resposta do controlador PID, com  $K_c = 53,2882$  e  $K_i = 0$ ,  $K_d = 20,2567$



Fonte: Autor (2024)

Portanto, é visível que o controlador PID obteve um melhor desempenho e uma resposta mais suave para a mudança de *setpoint*. Para dar ênfase a esta conclusão, pode-se observar

os valores das integrais do erro dispostos na Tabela 7, em que ele demonstra um desempenho superior em todas as quatro métricas utilizadas.

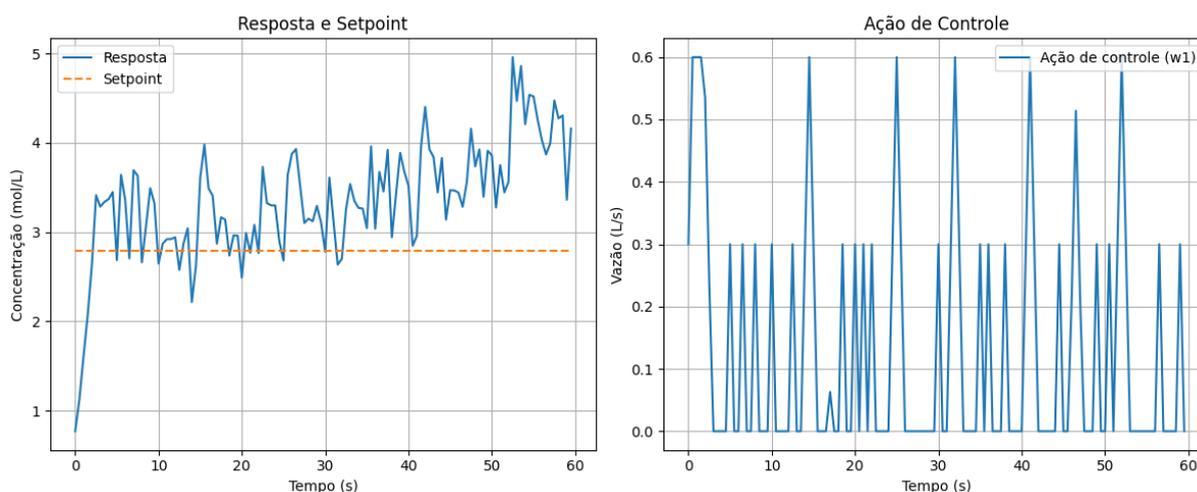
Tabela 7 – Métricas de erro para os controladores otimizados por AG

	IAE	ISE	ITAE	ITSE
P	4,1312	5,9605	5,6930	5,0413
PI	3,9640	5,9142	4,9170	4,8654
PID	<b>3,4778</b>	<b>5,7878</b>	<b>3,4267</b>	<b>4,4489</b>

Fonte: Autor (2024)

Porém, nem sempre os resultados das medições retornarão dados verdadeiros, então neste trabalho, também realizou-se uma simulação do comportamento do PID na presença de ruído, como representado pela Figura 24 .

Figura 24 – Resposta do controlador PID, com  $K_c = 53,2882$  e  $K_i = 0$ ,  $K_d = 20,2567$  e dados ruidosos



Fonte: Autor (2024)

Ao observar a figura percebe-se que a inserção do ruído nos dados de medição do controlador causou a desestabilização das ações de controle. Isso pode ser evidenciado pelo aumento significativo das integrais do erro que foram respectivamente para IAE, ISE, ITAE e ITSE 9,8896. 10,0453, 52,6292 e 38,0456.

Desse modo, fica nítido que o controlador PID é uma boa alternativa pela sua capacidade de levar o sistema ao *setpoint* de maneira precisa e suave. Porém, o mesmo não foi adequado para lidar com ruídos intensos nas medições, que podem causar suas ações de controle a serem instáveis.

## 5.2 Controle avançado

Desse modo, é pertinente buscar novas estratégias de controle mais robustas, para assim avaliar sua implementação nos casos aplicados ao PID no tópico passado. No que diz respeito ao controle avançado, a primeira etapa realizada na construção dos controladores MPC é o treinamento das redes que serão utilizadas como modelos do sistema.

### 5.2.1 MPC com aprendizado profundo

Assim, como descrito na metodologia, após o processo de pré-tratamento dos dados, foi realizado o processo de otimização dos hiperparâmetros para as redes recorrentes LSTM e GRU.

Em relação a LSTM, o *RandomSearch* escolheu os seguintes valores: 135 células, 1 camada, *sigmoid* como função de ativação, 0,00016 como taxa de aprendizado e o *lag* de 3. Esse conjunto de hiperparâmetros resultou nas métricas dispostas na Tabela 8.

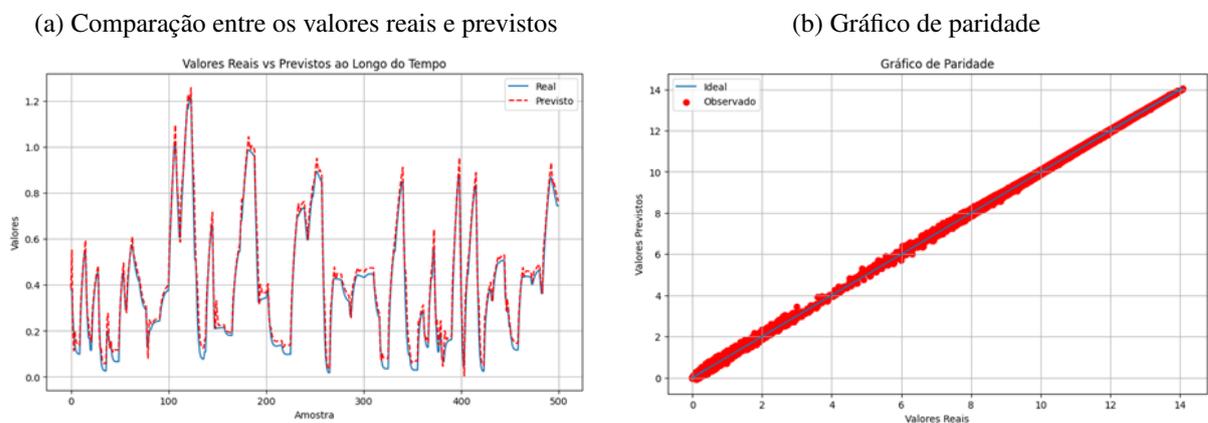
Tabela 8 – Resultados dos hiperparametros para a LSTM

Métrica	Resultado
$R^2$	0,99989
RMSE (mol/L)	0,04970
MAE (mol/L)	0,03802

Fonte: (Autor, 2024)

Observando os resultados da Tabela 8, percebe-se que as métricas obtiveram valores próximos aos ótimos. Por exemplo,  $R^2$  próximo de 1 e as demais próximas de 0. Esse ótimo desempenho é evidenciado quando se visualiza a comparação dos valores reais e previstos, Figura 25a, e o gráfico de paridade, Figura 25b.

Figura 25 – Resultados do modelo LSTM



Fonte: (Autor, 2024)

Ao observar o gráfico comparativo entre os valores reais e previstos e o de paridade percebe-se que o modelo foi capaz de acompanhar fielmente o comportamento dos dados. É possível observar na Figura 25a que os dados de previsão estão próximos aos valores reais, mesmo quando o processo apresenta mudanças bruscas. Isso denota a precisão do modelo, que é evidenciada pela Figura 25b, já que os dados estão agrupados próximos a linha de idealidade, que indica o ponto em que a previsão é igual ao valor real.

No caso da GRU, o *RandomSearch* definiu praticamente os mesmos valores escolhidos para a LSTM, diferindo apenas na taxa de aprendizado, 0,00012, e no *lag* de 7. Do mesmo modo, a Tabela 9, mostra o resultado da aplicação do melhor conjunto de hiperparâmetros.

Tabela 9 – Resultados dos hiperparâmetros para a GRU

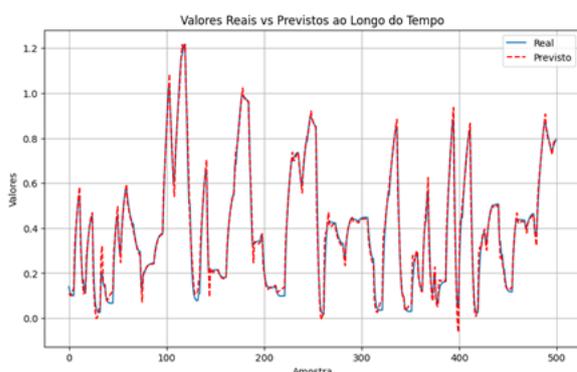
Métrica	Resultado
$R^2$	0,9999
RMSE (mol/L)	0,0436
MAE (mol/L)	0,0322

Fonte: (Autor, 2024)

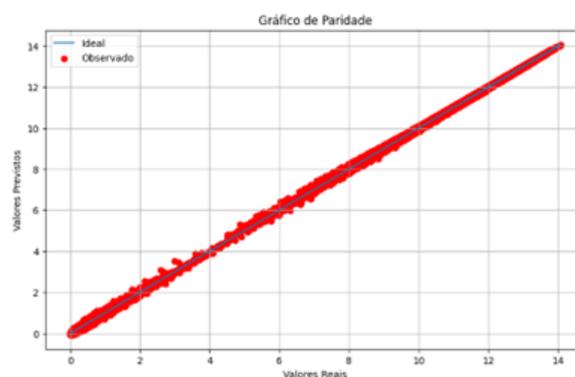
Nota-se que a GRU também foi capaz de prever bem o conjunto de dados, obtendo resultados próximos o bastante a LSTM para não indicar uma diferença significativa entre as duas redes. De maneira análoga, pode-se ver a atuação da GRU em relação a comparação dos valores previstos e reais, Figura 26a, e com o gráfico de paridade, Figura 26b.

Figura 26 – Resultados do modelo GRU

(a) Comparação entre os valores reais e previstos



(b) Gráfico de paridade



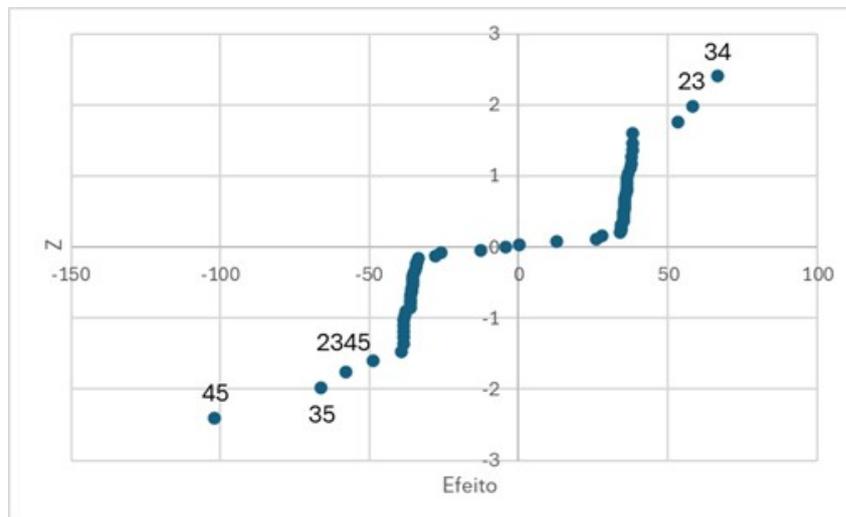
Fonte: (Autor, 2024)

Dessa maneira, ambos modelos apresentaram uma ótima eficiência. Assim, os dois são adequados para serem utilizados como modelos internos do MPC. Para investigar mais a fundo o desempenho desses modelos no contexto do controle avançado, bem como para observar

sua interação com as variáveis de sintonia específicas do MPC, foi realizado o planejamento experimental descrito na metodologia deste trabalho.

Como citado anteriormente, o planejamento contou com 48 ensaios distintos para analisar um conjunto de seis variáveis  $lag$ ,  $H_p$ ,  $H_c$ ,  $\delta$ ,  $\lambda$  e modelo, descritas respectivamente como variáveis 1, 2, 3, 4, 5 e 6. O gráfico das 63 interações de efeitos pode ser visualizado na Figura 27. Ao fim da análise estatística o teste t de Student indicou que 5 interações, mostradas na figura, foram significativas.

Figura 27 – Gráfico dos efeitos



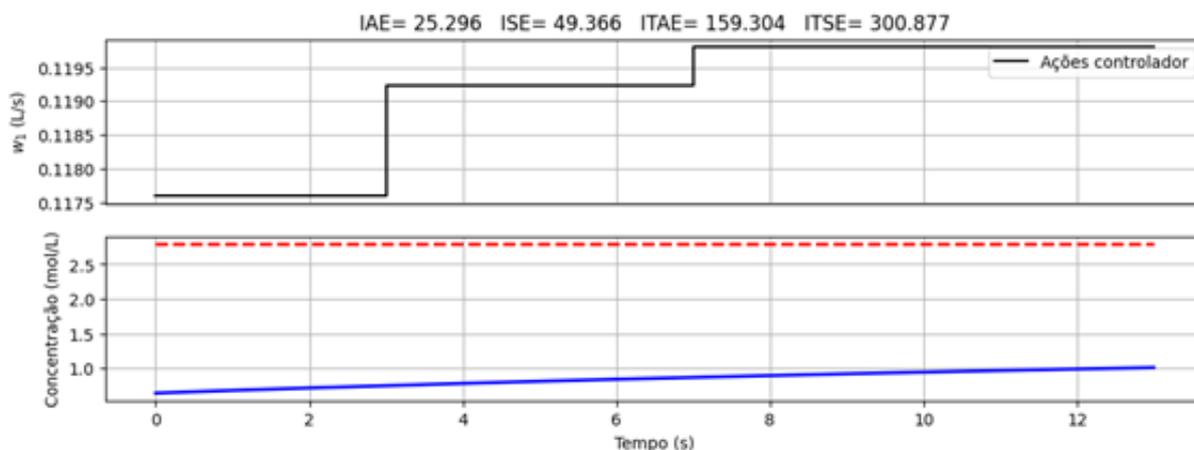
Fonte: Autor (2024)

Percebe-se que todos os efeitos significativos foram interações entre as variáveis de sintonia do MPC. Isto indica que, desde que o modelo seja adequado, tanto os parâmetros de sintonia (como o  $lag$ , variável 1) quanto a escolha do modelo (variável 6) não apresentam mudanças ou interações significativas.

Dentre as variáveis de sintonia do MPC, a interação 2345 reúne as relações combinadas das outras interações, desse modo, não irá ser comentada neste trabalho. De todas as interação entre os parâmetros, a interação 45, que se refere as constantes da função objetivo,  $\delta$  (variável 4) e  $\lambda$  (variável 5), foi a mais significativa.

Nos ensaios realizados, todos os controladores que utilizaram o valor de  $\lambda$  maior do que o de  $\delta$  ( $\lambda = 100$  e  $\delta = 0,1$ ) obtiveram resultados insatisfatórios, como exemplificado na Figura 28. Esta figura consta na parte superior o acompanhamento das ações de controle e na parte inferior a resposta do sistema (linha azul) em relação ao  $setpoint$  (linha tracejada vermelha).

Figura 28 – Resultado do MPC-GRU com lag = 7,  $H_p = 7$ ,  $H_c = 1$ ,  $\delta = 0,1$  e  $\lambda = 100$



Fonte: (Autor, 2024)

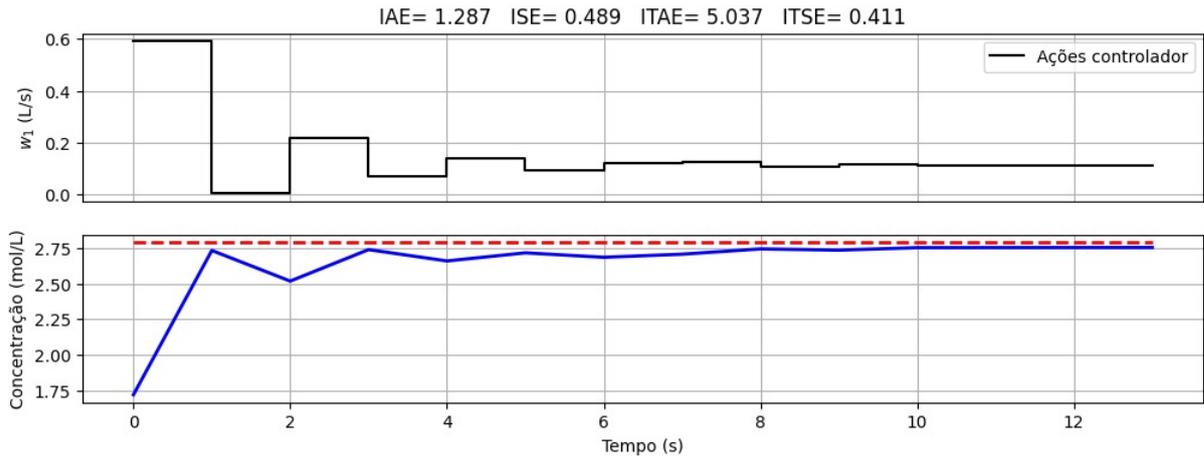
Ao avaliar a Figura 28, nota-se que o motivo desse baixo desempenho é que o MPC prioriza reduzir o esforço do controlador em detrimento da minimização do erro. Isto pois  $\delta$  e  $\lambda$  são respectivamente as constantes do sinal de erro e do esforço do controlador. E portanto com um alto valor de  $\lambda$ , o controlador reduz a mudança na variável manipulada, que neste caso foram da ordem de  $2 \cdot 10^{-3} L/s$ .

As relações 34 e 35 são análogas já que relacionam respectivamente o  $H_c$  aos pesos  $\delta$  e  $\lambda$ . Em ambos os casos, o valor escolhido delimita a prioridade da precisão das respostas do sistema em relação a suavidade da convergência. Valores pequenos de  $\delta$ , ou maiores de  $\lambda$ , resultam em uma baixa variação das vazões do  $H_c$ . Isso pode ser um ponto positivo, ao causar uma convergência estável e suave, ou negativo, caso o sistema não alcance o *setpoint* em um tempo aceitável. Enquanto valores mais elevados de  $\delta$  e menores de  $\lambda$ , exigem uma resposta mais rápida do sistema, e assim permite chegar ao *setpoint* logo, porém pode gerar mudanças bruscas nas vazões, e assim, causar instabilidade.

De forma semelhante ao discutido anteriormente, a relação entre os  $H_p$  e  $H_c$  influencia diretamente a suavidade da resposta no controle preditivo. Horizontes menores para ambos os parâmetros tendem a direcionar a otimização para uma convergência mais rápida em relação aos distúrbios e mudanças de *setpoint*. Em contrapartida, horizontes mais longos proporcionam uma resposta mais gradual e suavizada, pois as ações de controle são ajustadas com maior moderação e distribuídas em um maior número de vazões. Isto contribui para a estabilidade do sistema e evita oscilações abruptas nas variáveis de controle, porém como dito antes pode não atender urgência da variável controlada voltar ao *setpoint*.

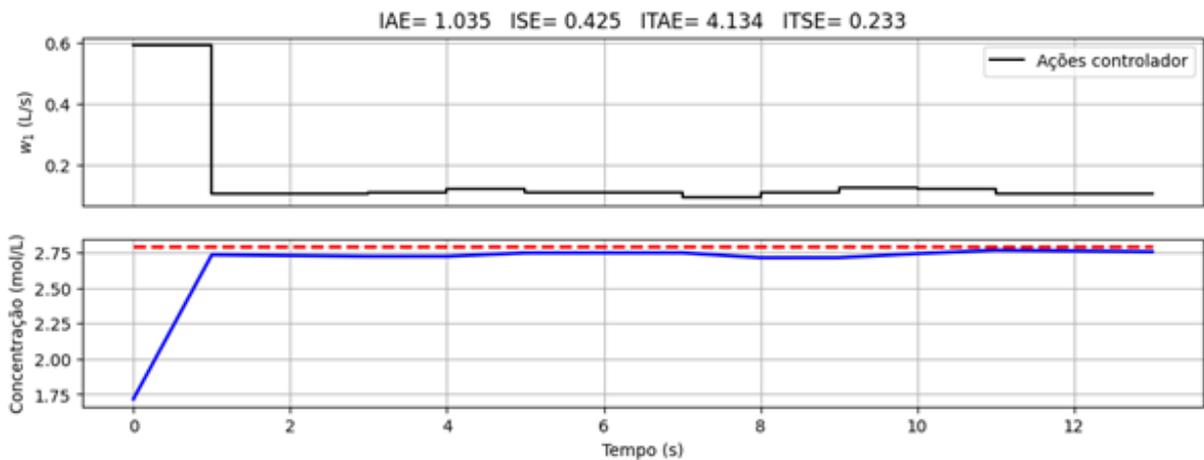
Neste contexto, é possível perceber que a sintonia desses parâmetros é uma tarefa complexa devido à necessidade de balancear a celeridade de resposta com a estabilidade do sistema. Ao levar isto em consideração, os melhores resultados obtidos nos ensaios realizados, para este sistema, utilizaram parâmetros de  $H_p$  e  $H_c$ , como também de  $\delta$  e  $\lambda$ , próximos uns dos outros, como representado nas Figuras 29 e 30.

Figura 29 – Resultado do MPC-LSTM com lag = 7,  $H_p = 2$ ,  $H_c = 1$ ,  $\delta = 100$  e  $\lambda = 0,1$



Fonte: (Autor, 2024)

Figura 30 – Resultado do MPC-GRU com lag = 7,  $H_p = 7$ ,  $H_c = 7$ ,  $\delta = 100$  e  $\lambda = 100$



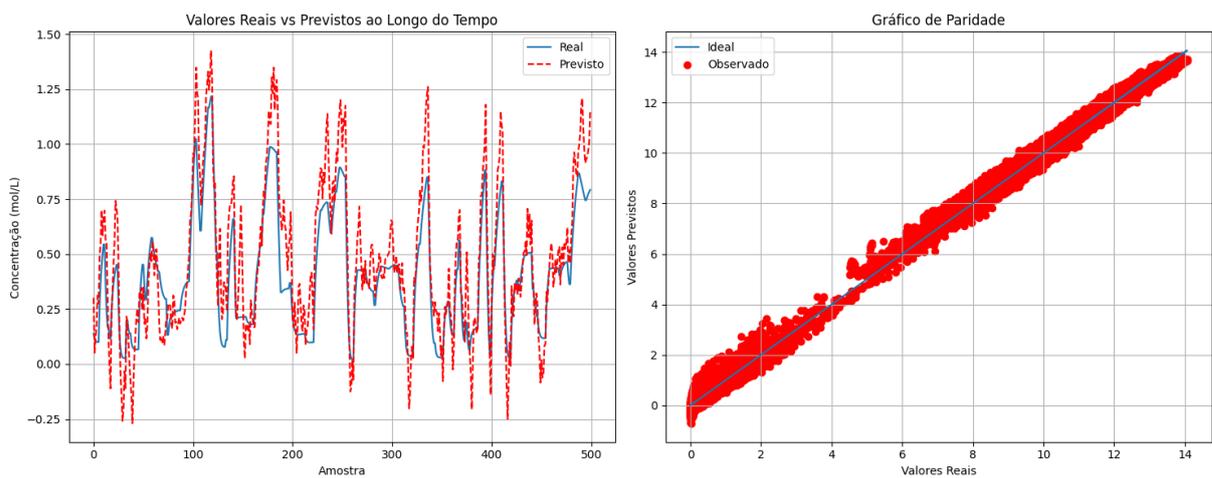
Fonte: (Autor, 2024)

Ao observar os gráficos acima, percebe-se um paralelo interessante da discussão anterior. Note que ambos os gráficos possuem um valor de  $\delta$  maior ou igual a  $\lambda$ . Isto para que a redução do erro seja priorizada, porém a Figura 30 apresenta um conjunto de parâmetros que beneficiam uma convergência com maior estabilidade, como o  $H_p$ ,  $H_c$  e  $\lambda$  no nível superior.

De fato, pode-se perceber que a Figura 29 possui uma maior oscilação antes de chegar ao *setpoint* se comparada com a Figura 30. Porém, essa escolha de valores para os horizontes também significou um tempo total de uma hora e dezessete minutos para a Figura 30 em relação a vinte dois minutos da Figura 29, já que no primeiro caso a quantidade de vazões a serem otimizadas e a quantidade de concentrações a serem previstas é maior.

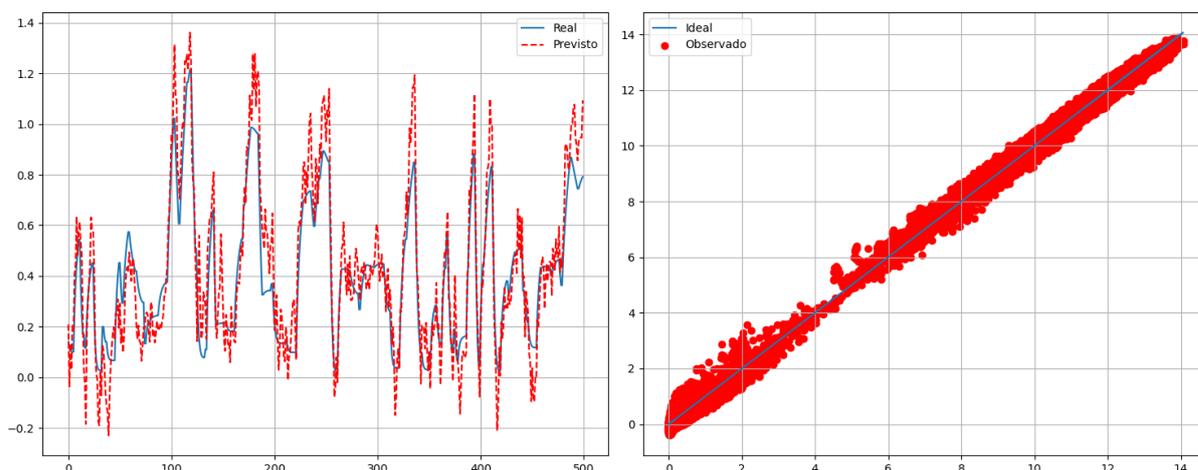
Essas observações, juntamente com os valores das métricas expostas nas figuras, indicam que esses MPCs apresentam um ótimo desempenho no controle do processo estudado. Contudo, para uma análise complementar, é importante investigar se essa metodologia consegue lidar com a presença de ruído nos dados. Dessa maneira, cada uma dessas duas redes foram retreinadas com objetivo de preverem dados reais a partir dos valores da vazão e dos ruídos na concentração. O resultado desse experimento pode ser visualizado nas Figuras 31 e 32.

Figura 31 – Desempenho da LSTM com dados ruidosos



Fonte: Autor (2024)

Figura 32 – Desempenho da GRU com dados ruidosos



Fonte: Autor (2024)

Pelas figuras é possível notar que ambas as redes perderam desempenho devido a inserção dos ruídos. No gráfico de comparação das previsões é possível perceber que a eficiência das redes é prejudicada principalmente nas mudanças bruscas, prevendo extremos de concentração muito maiores, ou muito menores dos que os dados do processo. Essa conclusão é evidenciada com o aumento da dispersão em ambos os gráficos de paridade que também pode ser percebido com a diminuição dos valores de  $R^2$  e aumento considerável das médias de erro, mostradas na Tabela 10.

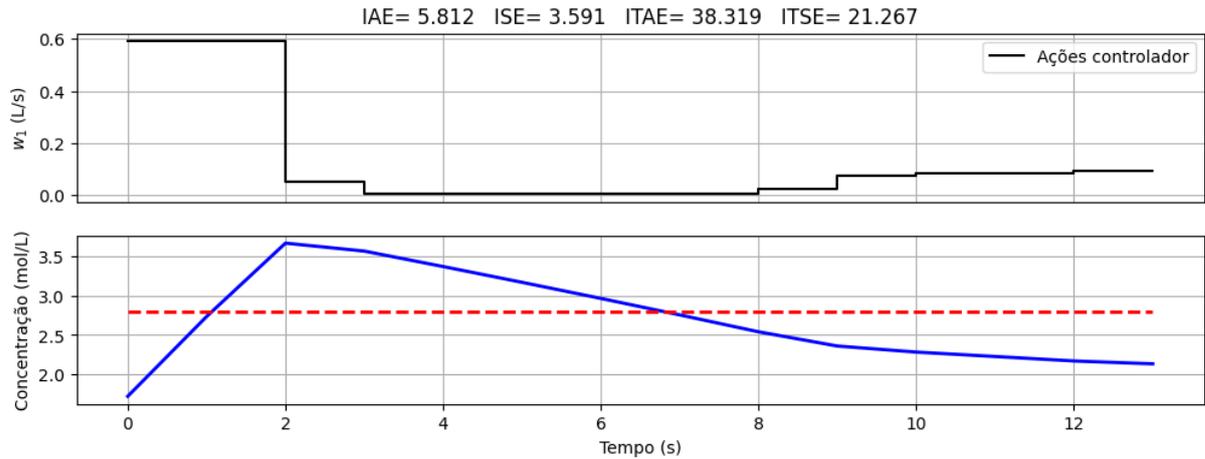
Tabela 10 – Comparação do desempenho das redes após a inserção do ruído

Modelos	$R^2$	MAE (mol/L)	RMSE (mol/L)
LSTM	0,9999	0,0380	0,0497
GRU	0,9999	0,0322	0,0436
LSTM-Ruído	0,9984	0,1543	0,1954
GRU-Ruído	0,9988	0,1343	0,1726

Fonte: Autor (2024)

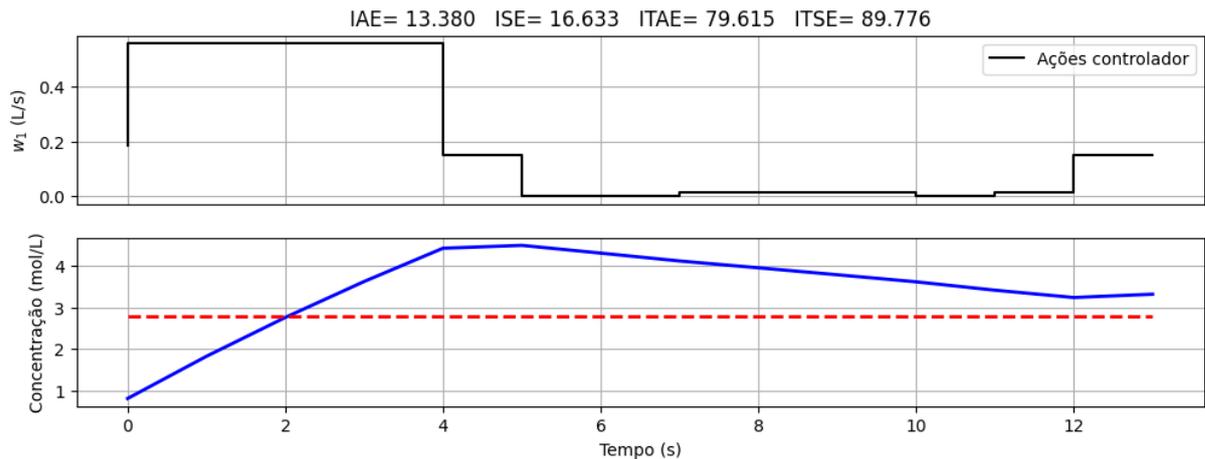
Dessa forma, como os modelos obtiveram uma redução em suas métricas espera-se que a aplicação deles como modelo interno do MPC também reduza o seu desempenho. Assim como o esperado, pode-se observar que as Figuras 33 e 34 apresentam desempenhos inferiores em relação aos dados sem ruído.

Figura 33 – Resultado do MPC-LSTM com  $lag = 7$ ,  $H_p = 2$ ,  $H_c = 1$ ,  $\delta = 100$  e  $\lambda = 0,1$  e dados ruidosos



Fonte: Autor (2024)

Figura 34 – Resultado do MPC-GRU com  $lag = 7$ ,  $H_p = 7$ ,  $H_c = 7$ ,  $\delta = 100$  e  $\lambda = 100$  e dados ruidosos



Fonte: Autor (2024)

Ao analisar as figuras, pode-se inferir que a presença de erro nos dados de treino dos modelos, além de provocar *overshoot* introduziu instabilidade nas previsões, resultando em respostas do sistema que, embora se aproximem do *setpoint*, tendem a se desviar dele.

### 5.2.2 MPC com modelos híbridos

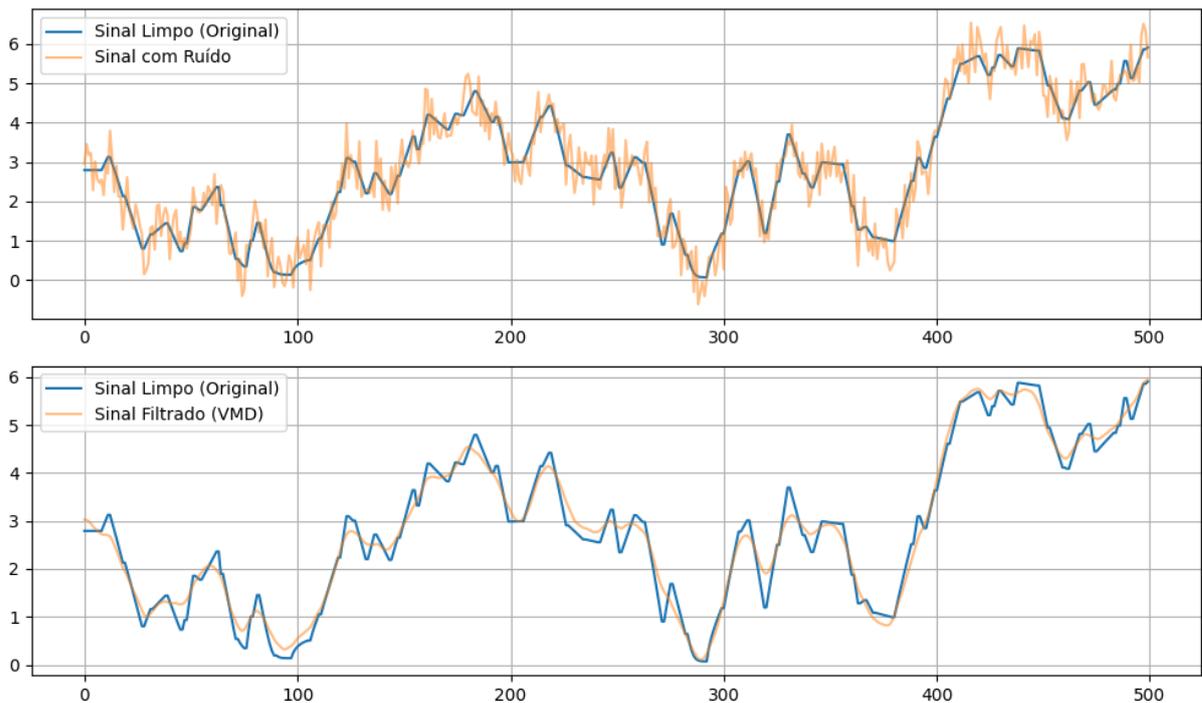
Assim, é possível notar que a presença de ruídos nos dados é um aspecto determinante na qualidade dos modelos, e assim na qualidade do controle preditivo. Para endereçar esse problema, é interessante investigar o impacto da inserção de métodos de tratamento de ruído como a VMD, antes do treinamento de modelos, com os chamados de modelos híbridos.

Para utilizar a melhor configuração da VMD foi-se utilizado um AG para otimizar a escolha do número de IMF (K) e o parâmetro  $\alpha$ .

Porém é importante destacar que a otimização da VMD foi limitada devido a grande demanda computacional que ela impõe ao ser aplicada a um grande número de dados. Para contornar esse problema, foi necessário reduzir o tamanho dos dados aplicados a VMD, o que restringiu a aplicação da otimização da reconstrução de sinais a apenas metade dos dados originais.

Os valores escolhidos pelo AG foram  $K = 3$  e  $\alpha = 1666$ , e o resultado deste tratamento pode ser visualizado na Figura 35. No gráfico da parte superior é mostrado a dimensão das flutuações em relação a concentração real. Enquanto que na parte inferior é mostrada o resultado do filtro da VMD.

Figura 35 – Tratamento do ruído com a VMD



Fonte: Autor (2024)

A análise da figura revela que a VMD obteve um desempenho satisfatório na remoção do ruído nos dados de concentração do reator. Ao observar o sinal reconstruído pela VMD, percebe-se que as flutuações foram significativamente suavizadas. Este sinal, portanto, acompanha de forma mais fiel o sinal limpo original. Isso sugere que a VMD foi bem-sucedida em decompor o sinal ruidoso com objetivo de evidenciar apenas a principal tendência dos dados.

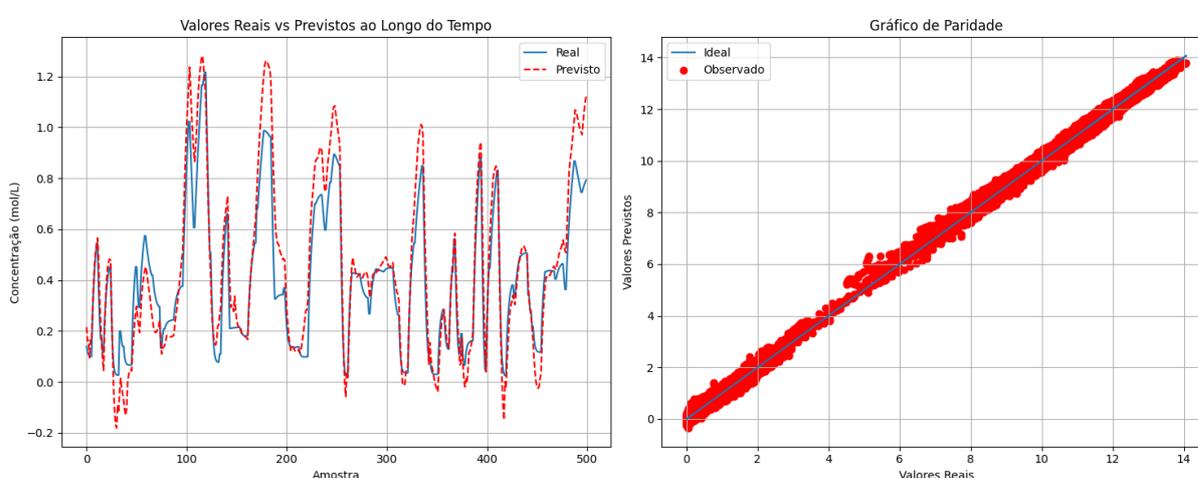
Desse modo, essa configuração foi utilizada para realizar a comparação entre os modelos

utilizados no item anterior. Assim, aplicou-se a VMD aos dados de ruído, e o resultado desse tratamento foi utilizado para treinar os mesmo modelos LSTM e GRU utilizados na etapa passada.

Ao aplicar os dados reconstruídos às duas melhores redes identificadas anteriormente, observa-se que o desempenho obtido pelos modelos híbridos foi bastante similar entre si.

Os gráficos de desempenho dos modelos híbridos VMD-LSTM e VMD-GRU mostram resultados praticamente idênticos. Assim, a Figura 36 apresenta o desempenho do modelo VMD-GRU como uma representação eficaz do comportamento observado em ambos os modelos.

Figura 36 – Resultado do modelo híbrido VMD-GRU



Fonte: Autor (2024)

Assim como nas Figuras 31 e 32, percebe-se pela Figura 36 que os modelos híbridos também tiveram dificuldades em estimar com precisão os valores em que a variação brusca leva a picos. Esses pontos podem não ter sido bem representados na suavização imposta pela VMD. Dessa forma, para avaliar melhor o desempenho desses modelos pode-se analisar a Tabela 11 que dispõe as métricas comparativas.

Tabela 11 – Comparação do desempenho dos modelos híbridos com ruído

Modelos	R <sup>2</sup>	MAE (mol/L)	RMSE (mol/L)
LSTM-Ruído	0,9984	0,1543	0,1954
GRU-Ruído	0,9988	0,1343	0,1726
VMD-LSTM	<b>0,9993</b>	<b>0,0996</b>	<b>0,1276</b>
VMD-GRU	<b>0,9993</b>	0,1001	0,1284

Fonte: Autor (2024)

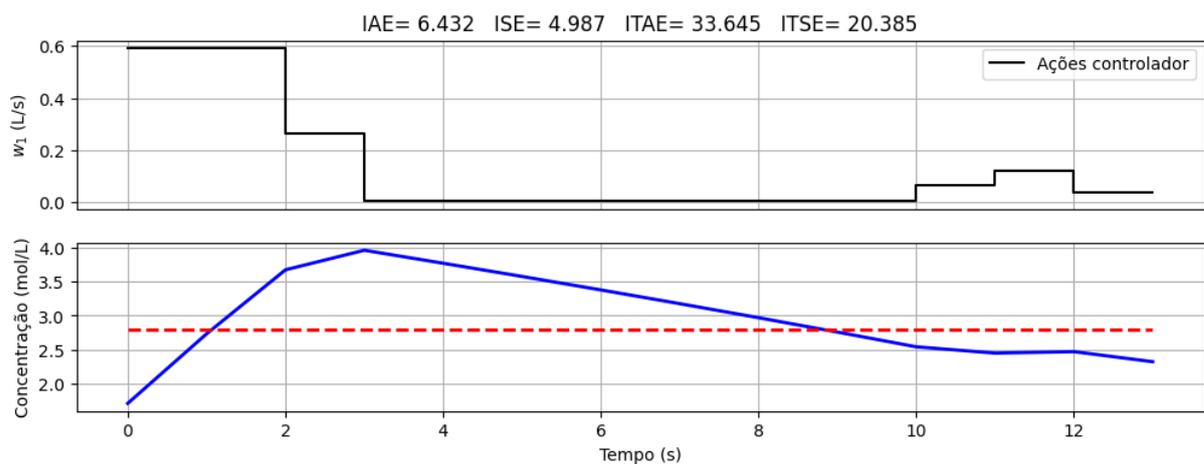
Percebe-se primeiramente que as métricas não estão significativamente tão distantes umas das outras, principalmente quando se avalia os erros (MAE e RMSE). Isso pode estar

relacionado a dificuldade, em geral, de previsão dos valores dos picos de variação, aspecto que não foi solucionado pelo filtro de VMD. Provavelmente, esta variação brusca foi interpretada como parte do erro, e assim também foi mascarada pelo filtro da VMD, que dessa maneira interferiu na previsão dos modelos.

Entretanto, vale salientar que houve uma melhora no valor de  $R^2$ , que pode ser percebida pela diminuição da dispersão ao redor da linha de idealidade no gráfico de paridade. Isso indica que ocorreram mais previsões próximas dos valores ideais. Estas devem estar relacionadas com o aumento do desempenho no intervalo entre os valores de pico de concentração, como mencionado anteriormente, devido a suavização proporcionada pela VMD.

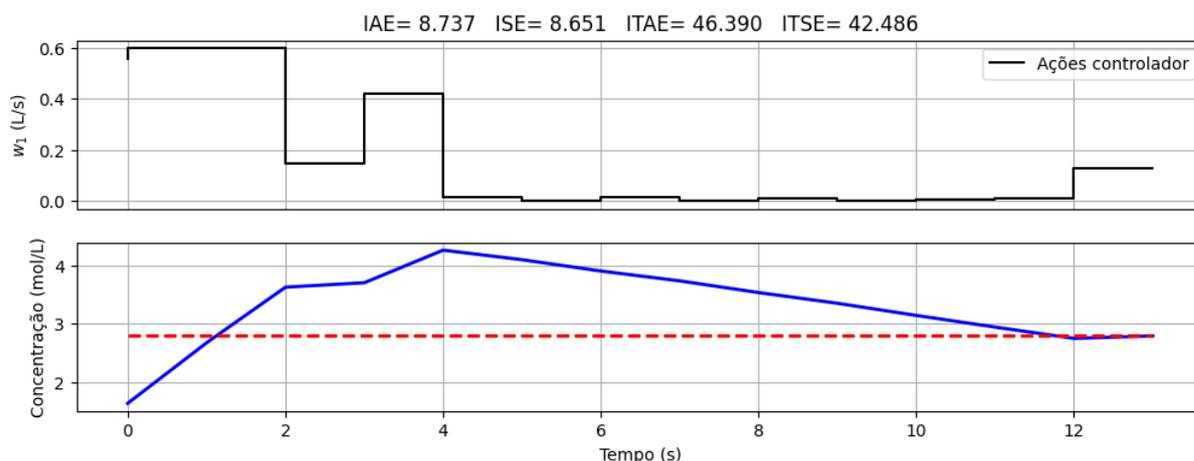
Desse modo, como os resultados dos modelos híbridos foram similares aos modelos GRU e LSTM, espera-se que seus desempenhos no MPC também sejam, assim como mostrado nas Figuras 37 e 38 .

Figura 37 – Resultado do MPC-VMD-LSTM com  $lag = 7$ ,  $H_p = 2$ ,  $H_c = 1$ ,  $\delta = 100$  e  $\lambda = 0,1$  e dados ruidosos



Fonte: Autor (2024)

Figura 38 – Resultado do MPC-VMD-GRU com lag = 7,  $H_p = 7$ ,  $H_c = 7$ ,  $\delta = 100$  e  $\lambda = 100$  e dados ruidosos



Fonte: Autor (2024)

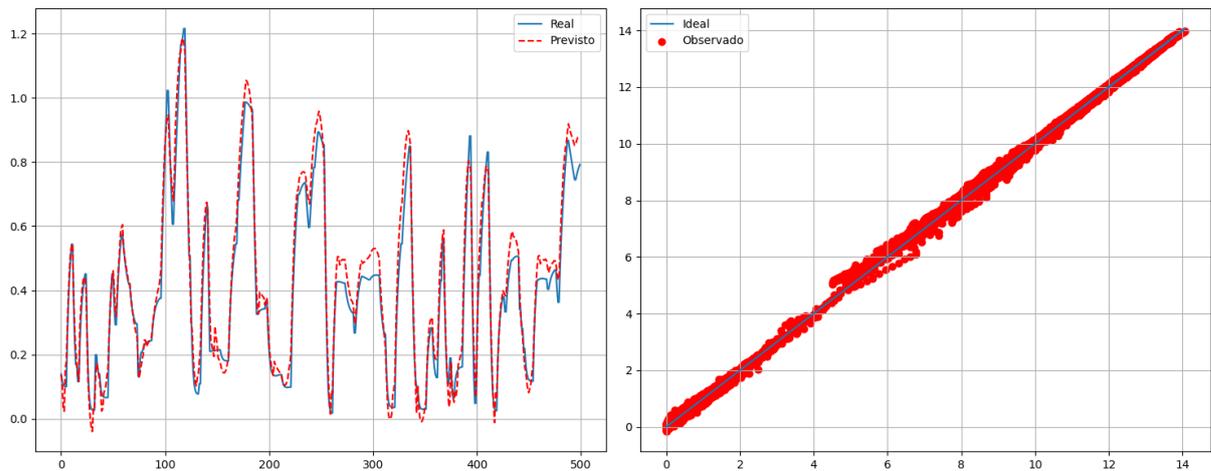
Ao comparar os resultados dos MPC-LSTM e MPC-GRU com as suas respectivas versões com modelo híbrido, nota-se que o comportamento do controlador foi de fato similar. Entretanto, é possível perceber que estas diferenças não foram consistentes.

Os valores das métricas para os controladores MPC-LSTM e MPC-VMD-LSTM foram bastante próximos, apresentando um aumento médio de 8,19% nas integrais do erro para o caso com VMD. Em contrapartida, para os controladores MPC-GRU e MPC-VMD-GRU, a aplicação da VMD resultou em uma redução de 44,95% nas métricas de desempenho. Esse resultado pode ser observado nas figuras, onde a Figura 38 demonstra que o controlador conseguiu atingir o *setpoint* de maneira mais suave.

Uma possível explicação, é o fato do modelo MPC-LSTM ter obtido um desempenho consideravelmente melhor que o MPC-GRU para os dados de ruído. Dessa maneira, a VMD pode ter tido impacto apenas no controlador que não estava adaptado ao erro, MPC-GRU. Isso provavelmente se deve à escolha dos parâmetros de sintonia do MPC, que podem ter sido melhor adaptados ao ruído na sintonia do MPC-LSTM do que do MPC-GRU.

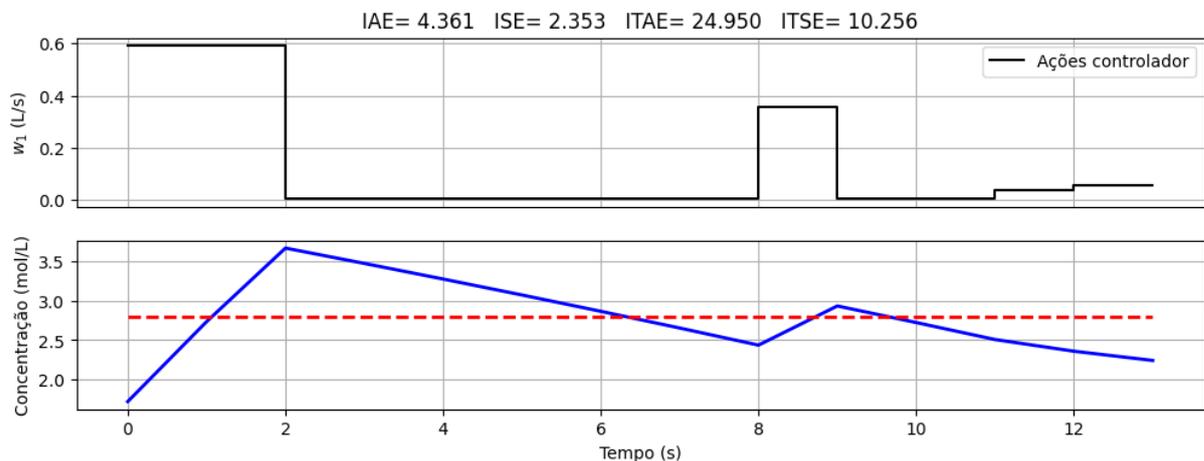
Como critério de comparação, também foi investigado o desempenho da VMD no caso em que os dados utilizados não possuam ruído. Ambos os modelos híbridos neste caso obtiveram em média um  $R^2 = 0,9998$ , um MAE = 0,0445 mol/L e um RMSE = 0,0591 mol/L. A Figura 39 mostra o resultado para o modelo VMD-LSTM.

Figura 39 – Resultado do modelo híbrido VMD-LSTM para dados sem ruído



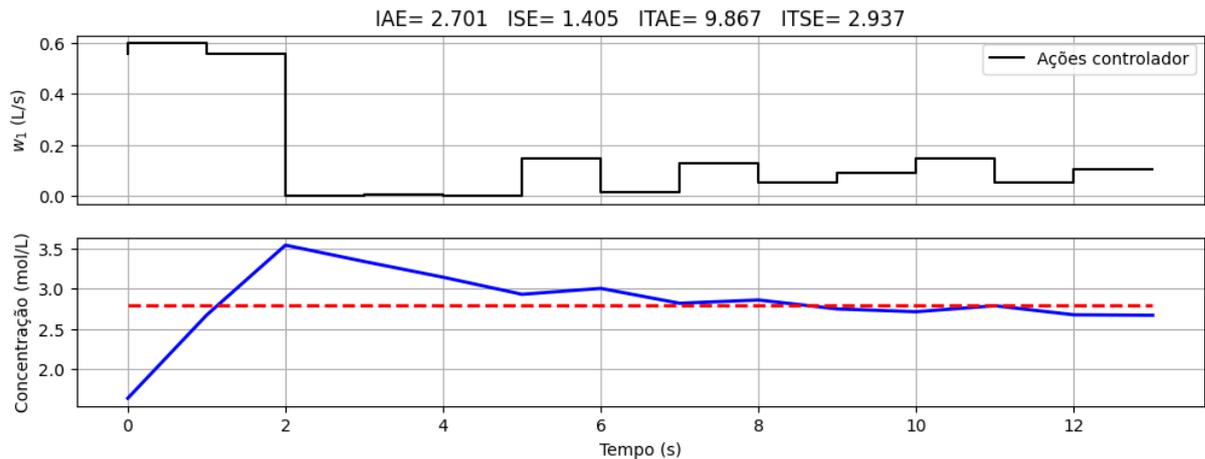
Fonte: Autor (2024)

Observa-se que o desempenho dos modelos melhorou consideravelmente, especialmente nas previsões dos picos de variação de concentração. No entanto, esse desempenho ainda é inferior ao dos modelos GRU e LSTM sem a presença de ruído. Dessa forma, como a eficiência dos modelos melhoraram, os resultados de seus respectivos MPCs também, como pode ser visto nas Figuras 40 e 41.

Figura 40 – Resultado do MPC-VMD-LSTM com  $lag = 7$ ,  $H_p = 2$ ,  $H_c = 1$ ,  $\delta = 100$  e  $\lambda = 0,1$ 

Fonte: Autor (2024)

Figura 41 – Resultado do MPC-VMD-GRU com lag = 7,  $H_p = 7$ ,  $H_c = 7$ ,  $\delta = 100$  e  $\lambda = 100$



Fonte: Autor (2024)

Assim, a aplicação da VMD como método de pré-processamento dos dados mostrou-se promissora ao suavizar as variações ruidosas nos sinais de concentração, como visto na Figura 35. Porém, a comparação entre os controladores com e sem VMD indicou resultados mistos tanto para o caso sem e, principalmente, com ruído. Para o caso com ruído, o desempenho dos modelos híbridos foi considerável apenas no caso do MPC-VMD-GRU, que teve uma melhora notável. Porém, ainda sim não obteve um resultado comparável aos dos MPC-LSTM e MPC-VMD-LSTM. Estes em que o desempenho do MPC-LSTM com dados ruidosos foi até melhor que o MPC-VMD-LSTM.

### 5.2.3 Comparação dos controladores

Portanto, a fim de sintetizar todas as discussões realizadas a respeito dos controladores estudados neste trabalho, é necessário compara-los em relação as métricas alcançadas em cada um dos dois tipos de caso descritos, com e sem ruído.

A Tabela 12 expõe os valores das métricas para o caso dos dados sem ruído. Pode-se perceber que as métricas se encontram em um intervalo relativamente próximo, o que indica o bom resultado que essas técnicas obtiveram.

Tabela 12 – Comparação dos controladores utilizados

Controlador	IAE	ISE	ITAE	ITSE	Tempo
PID	3,4778	5,7878	<b>3,4267</b>	4,448	-
MPC-LSTM	1,287	0,489	5,037	0,411	<b>22min</b>
MPC-GRU	<b>1,035</b>	<b>0,425</b>	4,134	<b>0,233</b>	1h17min
MPC-VMD-LSTM	4,361	2,353	24,950	10,256	25min
MPC-VMD-GRU	2,701	1,405	9,867	2,937	1h07min

Fonte: Autor (2024)

Ao avalia-las é possível observar que aquelas que obtiveram em geral o melhor resultado foram as que utilizaram a estratégia MPC com modelos de aprendizado de máquina. Entretanto, a metodologia de controle PID mostrou um desempenho satisfatório, principalmente no que tange as métricas acumulativas (ITAE e ITSE). Pode-se notar ainda, que devido a sua simplicidade, o tempo de execução dele é praticamente instantâneo em relação as outras técnicas. Isto o torna uma alternativa interessante já que proporciona um equilíbrio entre desempenho e tempo de execução. E por isso é o controlador mais utilizado em aplicações que envolvam processos com restrições em relação ao tempo de resposta.

Em relação aos controladores MPC-LSTM e MPC-GRU, pode-se perceber que suas métricas são muito próximas, com a única diferença que o MPC-GRU tenha conseguido chegar ao *setpoint* primeiro e de maneira mais estável. Entretanto, pode-se notar que o tempo de execução do MPC-GRU é três vezes maior que o do MPC-LSTM. Essa agilidade na execução, assim como dito no parágrafo anterior, pode ser um fator determinante na escolha da aplicação, principalmente se somado a um processo de sintonia mais adequado. Dessa maneira, por mais que ambos tenham sido os melhores controladores dentre os comparados neste quesito, não se pode afirmar qual dos dois é o melhor.

Por fim, em relação aos controladores MPC com modelos híbridos, pode-se dizer que a inserção da VMD não atuou de modo significativo no processo de controle, já que as métricas encontradas após a sua inserção ligeiramente piores. Isto pode ter acontecido devido a sua aplicação no conjunto de dados original. Pois pode ter causado uma suavização das variações causadas pela perturbação da vazão pela VMD.

Dessa forma, pode-se analisar pela Tabela 13 como os modelos se comportaram quando o ruído estava presente nos dados. Primeiramente percebe-se que a inserção do ruído causou um impacto negativo significativo no desempenho de todos os modelos, até mesmo para os modelos híbridos. Isso pode ser evidenciado pelo fato de que mesmo os melhores valores obtidos na Tabela 13 ainda estão abaixo das métricas alcançadas pelo pior controlador da Tabela 12, o

## MPC-VMD-LSTM.

Tabela 13 – Comparação dos controladores com dados ruidosos

Controlador	IAE	ISE	ITAE	ITSE	Tempo
PID	9,8896	10,0452	52,6292	38,0456	-
MPC-LSTM	<b>5,812</b>	<b>3,591</b>	38,319	21,267	21min
MPC-GRU	13,380	16,633	79,615	89,776	1h17min
MPC-VMD-LSTM	6,432	4,987	<b>33,645</b>	<b>20,285</b>	25min
MPC-VMD-GRU	8,737	8,651	46,390	42,486	1h10min

Fonte: Autor (2024)

Dessa forma, mesmo que a VMD tenha conseguido suavizar com sucesso o ruído inserido nos dados, o resultado obtido utilizando os dados tratados não foi consistente, como discutido no item anterior.

Dessa forma, pode-se notar que a sintonia dos parâmetros é um dos principais desafios de se utilizar o MPC e principalmente a VMD. Por um lado, a VMD é uma ferramenta poderosa para tratamento de ruído em diversos tipos de sinais. Por outro, a otimização dos seus parâmetros, como mencionado anteriormente, é uma tarefa complexa, árdua e que necessita um poder computacional relevante. Porém, sem a sintonia correta de seus parâmetros a decomposição dos sinais será ineficiente, e o ruído irá se misturar com os IMFs e assim, fazer com que as informações relevantes se percam (Huang *et al.*, 2021).

No que se diz respeito ao MPC, pode-se perceber pelas Tabelas 12 e 13 que essa estratégia foi a que obteve o melhor desempenho dentre todas as utilizadas. Porém, também observa-se o impacto que a variação em qualquer um dos seus parâmetros de sintonia causam no desempenho (Giraldo; Melo; Secchi, 2022).

Neste trabalho o processo de sintonia desses parâmetros foi limitado ao estudo realizado no planejamento experimental. Deste modo, com um processo de sintonia adequado essa estratégia pode obter resultados ainda melhores, e adaptados as necessidades do processo em si (Nebeluk; Ławryńczuk, 2021).

Por fim, vale salientar que o processo utilizado foi um processo relativamente simples, com apenas uma variável controlada e uma manipulada (SISO). Desse modo, a vantagem de se utilizar controladores MPC em relação ao PID aumentaria a medida que a complexidade do processo aumentasse. Seja por controlar variáveis que não são facilmente medidas, com a utilização de sensores virtuais, ou por realizar tipos de controles mais exigentes, como o MIMO (Shan *et al.*, 2023).

## 6 CONCLUSÃO

Portanto, os resultados obtidos se mostraram convincentes sobre os benefícios da utilização das estratégias MPC e PID no cenário industrial. Ambas as estratégias têm vantagens e desafios que de certa maneira se complementam e direcionam essas técnicas para diferentes tipos de aplicações.

O controlador PID é uma estratégia renomada, confiável e de bom custo-benefício. Dessa forma, devido à sua notoriedade, é facilmente implementado na indústria, que possui estruturas e equipamentos que operarem de maneira otimizada com essa técnica.

Portanto, em casos como o que foi estudado por este trabalho, é possível entender o receio de se utilizar novas técnicas de controle que necessitarão de adaptação, manutenção e mão de obra específicas em detrimento a estratégias que já apresentam um bom desempenho, como o PID. Porém, à medida que surgem aplicações mais complexas, que necessitem de estratégias mais flexíveis e robustas, a diferença de desempenho do PID para estas técnicas se torna mais perceptível.

Nestes casos, o uso de estratégias de controle avançado, como o MPC, se mostra de grande auxílio. Pois, como descrito por este trabalho, possui uma estrutura que propicia a esta técnica condições de obter desempenhos satisfatórios seja em aplicações com ou sem ruído.

Em relação a aplicação dos modelos híbridos, percebeu-se a capacidade promissora da VMD de tratar ruídos presentes nos dados. Porém, este trabalho não obteve mudanças significativas de resultados após a inserção da VMD nos controladores.

Uma das principais causas foi justamente o processo complexo de otimização dos parâmetros dos modelos, principalmente, em relação a otimização da VMD para o conjunto completo de dados. Isso se mostrou uma tarefa demorada e que demandava mais potência computacional do que disponível.

No que se trata dos parâmetros da MPC em ambos os casos (modelos de aprendizado de máquina e híbridos) foi possível notar o impacto que a sintonia desses parâmetros causam no desempenho. Porém, devido a dificuldade mencionada acima, este trabalho não entrou em detalhes sobre a otimização dos parâmetros mais significantes:  $H_p$ ,  $H_c$ ,  $\delta$  e  $\lambda$ .

Assim, estudos focados na otimização dos parâmetros dos controladores MPC-GRU, MPC-LSTM, MPC-VMD-LSTM e MPC-VMD-GRU são promissores, especialmente para aprimorar o desempenho desses modelos e explorar sua aplicabilidade em sistemas de controle mais complexos.

## REFERÊNCIAS

- BOUKTIF, S. et al. Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting. **Energies**, v. 13, n. 2, 2020.
- BRENGEL, D. D.; SIEDER, W. D. Multistep nonlinear predictive controller. **Industrial & Engineering Chemistry Research**, v. 28, n. 12, p. 1812–1822, 1989.
- CAMACHO, E. F.; BORDONS, C. **Model Predictive Control**. Segunda ed. London: Springer London, 2007.
- CARVALHO, R.; ALVAREZ, L. Sintonia ótima de um controlador MPC aplicado à planta de Williams-Otto. [s.l.: s.n.].
- DHAKE, H.; KASHYAP, Y.; KOSMOPOULOS, P. Algorithms for Hyperparameter Tuning of LSTMs for Time Series Forecasting. **Remote Sensing**, v. 15, n. 8, 1 abr. 2023.
- DRAGOMIRETSKIY, K.; ZOSSO, D. Variational mode decomposition. **IEEE Transactions on Signal Processing**, v. 62, n. 3, p. 531–544, 1 fev. 2014.
- FIRDAUS ISHAM, M. et al. A Review on Variational Mode Decomposition for Rotating Machinery Diagnosis. **MATEC Web of Conferences**, 2019.
- FRANK, M.; DRIKAKIS, D.; CHARISSIS, V. Machine-Learning Methods for Computational Science and Engineering. **Computation**, v. 8, n. 1, p. 15, 3 mar. 2020.
- GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems**. 2. ed. [s.l.]: O’Reilly, 2019.
- GIRALDO, S. A. C.; MELO, P. A.; SECCHI, A. R. Tuning of Model Predictive Controllers Based on Hybrid Optimization. **Processes**, v. 10, n. 2, 1 fev. 2022.
- HUANG, Y. et al. A new financial data forecasting model using genetic algorithm and long short-term memory network. **Neurocomputing**, v. 425, p. 207–218, 15 fev. 2021.
- JING, G. et al. Noise Reduction Based on Improved Variational Mode Decomposition for Acoustic Emission Signal of Coal Failure. **Applied Sciences (Switzerland)**, v. 13, n. 16, 1 ago. 2023.

KATOCH, S.; CHAUHAN, S. S.; KUMAR, V. A review on genetic algorithm: past, present, and future. **Multimedia Tools and Applications**, v. 80, n. 5, p. 8091–8126, 1 fev. 2021.

KRAVARIS, C.; KOOKOS, I. K. **Understanding Process Dynamics and Control**. Cambridge: Cambridge University Press, 2021.

LANGAZANE, S. N.; SAHA, A. K. Effects of Particle Swarm Optimization and Genetic Algorithm Control Parameters on Overcurrent Relay Selectivity and Speed. **IEEE Access**, v. 10, p. 4550–4567, 2022.

LI, S. et al. Sub-Bottom Sediment Classification Using Reliable Instantaneous Frequency Calculation and Relaxation Time Estimation. **Remote Sensing**, v. 13, n. 23, p. 4809, 27 nov. 2021.

LI, S.; JIANG, P.; HAN, K. RBF Neural Network based Model Predictive Control Algorithm and its Application to a CSTR Process. In: 38th **Chinese Control Conference**. Anais... Guangzhou: jul. 2019.

LI, W. C.; BIEGLER, L. T. Process Control Strategies for Constrained Nonlinear Systems. **Ind. Eng. Chem. Res**, v. 27, n. 8, p. 1421–1433, 26 fev. 1988.

LIU, W. et al. A Review of Variational Mode Decomposition in Seismic Data Analysis. **Surveys in Geophysics**, Springer Science and Business Media B.V., 1 abr. 2023.

MATSUURA, T.; KATO, M. Concentration stability of the isothermal reactor. **Chemical Engineering Science**, v. 22, p. 171–184, 1967.

MOHAMMADI, A. et al. Multiobjective and Interactive Genetic Algorithms for Weight Tuning of a Model Predictive Control-Based Motion Cueing Algorithm. **IEEE Transactions on Cybernetics**, v. 49, n. 9, p. 3471–3481, 1 set. 2019.

NEBELUK, R.; ŁAWRYŃCZUK, M. Tuning of multivariable model predictive control for industrial tasks. **Algorithms**, v. 14, n. 1, 1 jan. 2021.

NÚÑEZ, F. et al. Neural Network-Based Model Predictive Control of a Paste Thickener over an Industrial Internet Platform. **IEEE Transactions on Industrial Informatics**, v. 16, n. 4, p. 2859–2867, 1 abr. 2020.

OGATA, K. **Engenharia de controle moderno**. 5. ed. [s.l.]: PEARSON, 2010.

- PALLATHADKA, H. et al. Applications of artificial intelligence in business management, e-commerce and finance. **Materials Today: Proceedings**, v. 80, p. 2610–2613, 1 jan. 2023.
- PANERATI, J. et al. Experimental methods in chemical engineering: Artificial neural networks–ANNs. **Canadian Journal of Chemical Engineering**, Wiley-Liss Inc., 1 set. 2019.
- PEREIRA, J. L. J. et al. A Review of Multi-objective Optimization: Methods and Algorithms in Mechanical Engineering Problems. **Archives of Computational Methods in Engineering**, Springer Science and Business Media B.V., 1 jun. 2022.
- PON KUMAR, S. S. et al. A Deep Learning Architecture for Predictive Control. **IFAC-PapersOnLine**, v. 51, n. 18, p. 512–517, 2018.
- QI, X.; LI, X.; HE, S. FMPC-VMD BASED CONTROL STRATEGY FOR TRACKING GENERATION PLANS OF MULTI-TYPE ENERGY STORAGE SYSTEM. **China International Conference on Electricity Distribution**, 2018.
- RAMASAMY, V. et al. Optimal tuning of model predictive controller weights using genetic algorithm with interactive decision tree for industrial cement kiln process. **Processes**, v. 7, n. 12, 1 dez. 2019.
- REN, Y. M. et al. A tutorial review of neural network modeling approaches for model predictive control. **Computers and Chemical Engineering**, v. 165, 1 set. 2022.
- RODRIGUES, L. L. et al. Tuning a model predictive controller for doubly fed induction generator employing a constrained genetic algorithm. **IET Electric Power Applications**, v. 13, n. 6, p. 793–801, 1 jun. 2019.
- SANTOS, A. T. Desenvolvimento de estratégia para projeto de controlador preditivo baseada em deep learning. Maceió: Universidade Federal de Alagoas, 2020.
- SARAIVA, S. V. Estudo de um controlador preditivo baseado em sistemas inteligentes. Maceió: Universidade Federal de Alagoas, 2017.
- SARAIVA, S. V.; SILVA, F. V.; CARVALHO, F. O. Comparative analysis of machine learning models for predictive control of the cyclopentadine production process. **Chemical Product and Process Modeling**, v. 15, n. 3, 1 set. 2020.

SCHWENZER, M. et al. Review on model predictive control: an engineering perspective. **International Journal of Advanced Manufacturing Technology**, Springer Science and Business Media Deutschland GmbH, 1 nov. 2021.

SEBORG, D. E.; EDGAR, T. F.; MELLICHAMP, D. A. **Process Dynamics and Control**. 4. ed. [s.l.]: John Wiley & Sons, 2016.

SHAHID, N.; RAPPON, T.; BERTA, W. Applications of artificial neural networks in health care organizational decision-making: A scoping review. **Plos one**, Public Library of Science, 1 fev. 2019.

SHAN, B. et al. Soft sensor model predictive control for azeotropic distillation of the separation of DIPE/IPA/water mixture. **Journal of the Taiwan Institute of Chemical Engineers**, v. 152, 1 nov. 2023.

SHIN, Y.; SMITH, R.; HWANG, S. Development of model predictive control system using an artificial neural network: A case study with a distillation column. 2020.

SHRESTHA, A.; MAHMOOD, A. Review of Deep Learning Algorithms and Architectures. **IEEE Access**, v. 7, p. 53040–53065, 2019.

SILVA, G. D. DA. Machine Learning no controle preditivo de processos. Maceió: Universidade Federal de Alagoas, 2020.

SONG, C. et al. A water quality prediction model based on variational mode decomposition and the least squares support vector machine optimized by the sparrow search algorithm (VMD-SSA-LSSVM) of the Yangtze River, China. **Environmental Monitoring and Assessment**, v. 193, n. 6, 1 jun. 2021.

STEPHANOPOULOS, G. Chemical Process Control. [s.l.: s.n.].

SUSSMAN, M. V; POTTER, C. Kinetics of Propylene and Ethylene Hydrogenation. **Industrial and Engineering Chemistry**, v. 46, n. 3, p. 457–465, mar. 1954.

TANG, D. et al. A n-Gated Recurrent Unit with review for answer selection. **Neurocomputing**, v. 371, p. 158–165, 2 jan. 2020.

VAN HOUTT, G.; MOSQUERA, C.; NÁPOLES, G. A review on the long short-term memory model. **Artificial Intelligence Review**, v. 53, n. 8, p. 5929–5955, 13 dez. 2020.

WANG, Y. et al. A denoising method for mining cable PD signal based on genetic algorithm optimization of VMD and wavelet threshold. **Sensors**, v. 22, n. 23, 1 dez. 2022.

WEI, H. et al. Comparative research on noise reduction of transient electromagnetic signals based on empirical mode decomposition and variational mode decomposition. **Radio Science**, v. 56, n. 10, 1 out. 2021.

WOOLF, P. **Chemical process dynamics and controls**. Michigan: LibreTexts, 2024.

WU, Z.; RINCON, D.; CHRISTOFIDES, P. D. Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes. **Journal of Process Control**, v. 89, p. 74–84, 2020.

YADAV, R. K. PSO-GA based hybrid with Adam optimization for ANN training with application in medical diagnosis. **Cognitive Systems Research**. 2020.

ZARZYCKI, K.; ŁAWRYŃCZUK, M. Advanced predictive control for GRU and LSTM networks. **Information Sciences**, v. 616, p. 229–254, 1 nov. 2022.

ZHANG, J. et al. Intelligent fault diagnosis of rolling bearings using variational mode decomposition and self-organizing feature map. **Journal of Vibration and Control**, v. 26, n. 21–22, p. 1886–1897, 1 nov. 2020.

ZHENG, Y.; WANG, X.; WU, Z. Machine learning modeling and predictive control of the batch crystallization process. **Industrial and Engineering Chemistry Research**, v. 61, n. 16, p. 5578–5592, 27 abr. 2022.