

UNIVERSIDADE FEDERAL DE ALAGOAS  
CENTRO DE TECNOLOGIA  
CURSO DE ENGENHARIA QUÍMICA

ALEF TENORIO SANTOS

**DESENVOLVIMENTO DE ESTRATÉGIA PARA PROJETO DE CONTROLADOR  
PREDITIVO BASEADA EM *DEEP LEARNING***

Maceió  
2020

ALEF TENORIO SANTOS

**DESENVOLVIMENTO DE ESTRATÉGIA PARA PROJETO DE CONTROLADOR  
PREDITIVO BASEADA EM *DEEP LEARNING***

Trabalho de Conclusão de Curso apresentado como requisito para a obtenção do título de Bacharel em Engenharia Química, sob orientação do Professor Dr. Frede de Oliveira Carvalho.

Maceió

2020

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

Bibliotecária: Taciana Sousa dos Santos – CRB-4 – 2062

S237d Santos, Alef Tenorio.

Desenvolvimento de estratégia para projeto de controlador preditivo baseada em *Deep Learning* / Alef Tenorio Santos. – 2020.

46 f. : il., figs. e tabs. color.

Orientador: Frede de Oliveira Carvalho.

Monografia (Trabalho de Conclusão de Curso em Engenharia Química) – Universidade Federal de Alagoas. Centro de Tecnologia. Maceió, 2021.

Bibliografia: f. 44-46.

1. Indústria 4.0. 2. Controle Preditivo baseado em Modelo (MPC). 3. *Machine Learning*. 4. *Deep Learning*. I. Título.

CDU: 66.0: 004.8

## AGRADECIMENTOS

Agradeço primeiramente a Deus por ter sonhos tão grandes para mim, ter permitido que eu chegasse até aqui e por me abençoar a cada dia. Aos meus pais José Cícero dos Santos e Selma Basto Tenório Santos por sempre acreditarem em mim, mesmo nos momentos de dificuldades e incertezas, e por estarem ao meu lado em todos os momentos. Agradeço também ao meu irmão Jônathan pelo auxílio, conselhos dados, apoio em momentos difíceis e pela companhia que me deu durante todos esses anos.

Ao Antônio Amaro Sobrinho (*In memoriam*), Severina Rita de Lima Alves e todos os seus filhos (Dulce, Whilmar, Marcos, Neymar e Niel) por terem me educado, ensinado que o respeito está entre as maiores virtudes do ser humano e por servirem de exemplo em tantos outros aspectos.

Aos meus amigos Rodrigo Farias e Professor João Inácio Soletti por terem me introduzido à ciência, me dado diferentes oportunidades de trabalho e por contribuírem grandemente para minha formação. Agradeço aos amigos que fiz em congressos, principalmente Felipe e Loyce por terem sido fonte apoio em diferentes cidades, pelas ideias de trabalho e pelos momentos de descontração.

Ao meu líder durante o período de estágio na Braskem S.A, Dorival Freire de Carvalho por me fazer enxergar que o trabalho sempre é recompensado e que não se deve aceitar a mediocridade. Agradeço também ao meu grande amigo Ewerton Amorim por ter me introduzido à linguagem *Python*, ter me auxiliado e encorajado em momentos de sobrecarga e esgotamento mental.

Ao meu Orientador Professor Frede de Oliveira Carvalho por estar sempre disponível e por investir parte de seu tempo na minha formação. Sem ele este trabalho não teria sido realizado. Agradeço também aos integrantes do Laboratório de Sistemas Inteligentes Aplicados (LABSIA): Gabriele, Lucas, Geovane, Pedro, Marcone, Bruno por terem me dado apoio técnico em vários momentos da graduação.

Agradeço ao PET Ciência e Tecnologia e seus integrantes por ter me dado a oportunidade de crescimento em tantos aspectos. Não imaginava que passaria por tantas experiências, tendo estas grande impacto na minha formação profissional e cidadã. Agradeço ao Tutor Professor Lucena por ser sempre um ponto de apoio e por encontrar alternativas em meio a problemas aparentemente insolúveis.

Aos amigos que fiz durante o curso de graduação Valdeir, Diogo, Mariany, Andreza, Filipe, Alysson, Daniel, Leonardo, Michael, Douglas e tantos outros com quem convivi intensamente durante os últimos anos, pelo companheirismo e pela troca de experiências que me permitiram crescer.

À minha cúmplice Lays por ter me dado apoio em todos os momentos, estar sempre disponível para me ouvir e me acalmar em meio a turbulências.

## RESUMO

Devido à busca pelo desenvolvimento industrial através de processos eficientes, a Indústria 4.0 vem ganhando espaço com atuação em diferentes áreas. Na indústria química pode ser destacado o desenvolvimento de processos interligados, com sistemas de controle mais estáveis, eficientes e robustos. Isso contribui para uma maior segurança de operações, adequação a especificações de produtos, assim como aumento na rentabilidade de processos. Diferentes tipos de controladores podem ser aplicados para se manter variáveis de processo nas condições desejadas, mas devido a não linearidades em sistemas de alta complexidade, tem crescido a utilização do Controle Preditivo baseado em Modelo (MPC). No MPC, basicamente, as condições do processo são aplicadas em um modelo dinâmico para a predição de comportamentos futuros e obtenção de melhores ações de controle. Algoritmos de *Machine Learning* (ML), como Redes Neurais Artificiais (RNA), têm ganho atenção no contexto Indústria 4.0 em uma gama de aplicações, em especial na utilização em modelos preditivos para MPC. Porém, para lidar com a dificuldade de generalização e a não aplicabilidade em alguns sistemas, na literatura especializada atual pode-se perceber a crescente investigação de estratégias para utilização de *Deep Learning* (DL), tipo de ML caracterizada por possuir estruturas de aprendizado com maior densidade de neurônios. Nesse contexto, o objetivo deste trabalho consiste no desenvolvimento de estratégias para construção de controlador MPC utilizando algoritmos de ML e DL como modelo interno a partir da linguagem de programação *Python*, que vem ganhando destaque na indústria e na comunidade científica por ser de fácil utilização, possuir bibliotecas que abrem leques para diferentes aplicações e por possuir licença gratuita. As estratégias construídas foram avaliadas e validadas através da aplicação em um processo reativo em reator CSTR (*continuous stirred tank reactor*) utilizado na verificação de estratégias de controle. A partir das avaliações dos controladores, sistemas com maior estabilidade foram obtidos a partir das estruturas de *Deep Learning* (redes recorrentes), contribuindo para maior estabilidade de processos no contexto Industrial 4.0.

**Palavras-Chave:** Controle preditivo, Indústria 4.0, CSTR, *Machine Learning*.

## ABSTRACT

Thinking of the search for industrial development through efficient processes, Industry 4.0 has been gaining space with operations in different areas. In the chemical industry, the development of interconnected processes can be highlighted, with more stable, efficient and robust control systems. This contributes to greater security of operations, adaptation to product specifications, as well as increased profitability of processes. Different types of drivers can be selected to keep the process variables in the desired conditions, but due to non-linearities in highly complex systems, the use of Model-Based Predictive Control (MPC) has grown. In MPC, basically, process conditions are applied in a dynamic model to predict future behavior and obtain better control actions. Machine Learning (ML) algorithms, such as Artificial Neural Networks (RNA), have gained attention in the Industry 4.0 context in a range of applications, especially in the use of predictive models for MPC. However, to deal with generalization difficulty and the non-applicability in some systems, in the current specialized literature can perceive the growing investigation for the use of Deep Learning (DL), type of ML characterized by having structures with a higher density of neurons. In this context, the aim of this work is to develop strategies for building an MPC controller using ML and DL algorithms as an internal model based on the Python programming language, which has been gaining prominence in the industry and in the scientific community for being easy to use, it has libraries that open fans for different applications and for having a free license. The constructed strategies were evaluated and validated through the application in a reactive process in a CSTR reactor (continuous stirred tank reactor) used in the control verification. From the driver structures, systems with greater stability were sought from the Deep Learning structures (recurrent networks), contributing to greater process stability in the Industrial 4.0 context.

**Keywords:** Predictive control, Industry 4.0, CSTR, Machine Learning.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Estratégia de controle <i>feedback</i> .....	17
Figura 2 – Ações de controlador MPC .....	19
Figura 3 – Controlador preditivo utilizando algoritmos de <i>Machine Learning</i> .....	20
Figura 4 – (a)Estrutura celular de um neurônio biológico e (b)forma simplificada de um neurônio artificial (perceptron).....	21
Figura 5 – Redes Neurais em Multicamada ( <i>Multilayer Perceptron</i> - MLP). Em verde a camada de entrada de entrada; em azul, a camada de saída; e em laranja, as camadas ocultas	22
Figura 6 – Célula recorrente a cada instante de tempo .....	23
Figura 7 – Rede Recorrente a cada instante de tempo.....	23
Figura 8 – Célula da Rede Recorrente de Longa Memória (LSTM).....	24
Figura 9 – Célula Gated Recurrent Unit (GRU).....	26
Figura 10 – Reação de produção do propileno glicol.....	27
Figura 11 – Reator contínuo de mistura perfeita (CSTR) .....	27
Figura 12 – Metodologia adotada no trabalho para construção do controlador preditivo.....	28
Figura 13 – Esquema do sistema de controle estudado .....	28
Figura 14 – Dados gerados entre os instantes 1000s e 1300s .....	31
Figura 15 – Estrutura de treinamento das células de <i>Machine e Deep Learning</i> .....	32
Figura 16 - Estrutura do controlador preditivo utilizado no estudo .....	33
Figura 17 – Dados reais versus calculados nas etapas de treino e teste da rede com células MLP.....	35
Figura 18 - Dados reais versus calculados nas etapas de treino e teste da rede com células RNN.....	35
Figura 19 - Dados reais versus calculados nas etapas de treino e teste da rede com células LSTM. ....	36
Figura 20 - Dados reais versus calculados nas etapas de treino e teste da rede com células GRU.....	36
Figura 21 – Performance do controlador baseado em MLP para diferentes pesos .....	37
Figura 22 - Performance do controlador baseado em RNN para diferentes pesos.....	38
Figura 23 - Performance do controlador baseado em LSTM para diferentes pesos.....	38
Figura 24 - Performance do controlador baseado em GRU para diferentes pesos.....	39
Figura 25 – Comparação das performances dos controladores utilizando os melhores pesos .	40
Figura 26 – Visão geral da performance dos controladores com peso da variável manipulada constante .....	40
Figura 27 - Visão geral da performance dos controladores com peso da variável controlada constante .....	41

## LISTA DE TABELAS

Tabela 1 – Cinética de reação considerada no estudo .....	30
Tabela 2 – Parâmetros físico-químicos considerados no estudo .....	30
Tabela 3 – Dimensões do reator considerado no estudo .....	30
Tabela 4 – Variáveis de processo consideradas no estudo. ....	30
Tabela 5 - Limite das variáveis utilizadas na geração de dados de treinamento .....	31

## LISTA DE ABREVIATURAS E SIGLAS

CSTR	<i>Continuous Stirred Tank Reactor</i> (Reator Agitado de Fluxo Contínuo)
CV	<i>Controlled Variable</i> (variável controlada)
DL	<i>Deep Learning</i>
GRU	<i>Gated Recurrent Unit</i> (GRU)
LSTM	<i>Long short-term memory</i> (Rede neural de longa memória)
ML	<i>Machine Learning</i>
MLP	<i>Multilayer Perceptron</i> (Perceptrons em Multicamadas)
MPC	Modelo de controle preditivo
P	Proporcional (controle)
PD	Proporcional-derivativo (controle)
PI	Proporcional-integral (controle)
PID	Proporcional, integral e derivativo (controle).
RNA	Redes Neurais Artificiais
RNN	<i>Recurrent Neural Network</i> (Redes Neurais Recorrentes)
SP	Trajectoria de referência – <i>setpoint</i>
PSO	<i>Particle Swarm Optimization</i> (Otimização por Enxame de Partículas)
IAE	<i>Integral of the absolute value of error</i> (Integral do Erro Absoluto)
ITAE	<i>Integral of time-weighted absolute error</i> (Integral do Erro Absoluto Ponderado pelo Tempo)
ITSE	<i>Integral of time-weighted squared error</i> (Integral do Erro Quadrático Ponderado pelo Tempo)
SISO	<i>Simple Input Simple Output</i> (Entrada e saída únicas)

## LISTA DE SÍMBOLOS

### **Processo**

$C_a$	Concentração de reagente no reator
$q$	Vazão de alimentação
$V$	Volume do reator
$C_{Alim}$	Concentração do reagente na alimentação
$k_0$	Constante cinética da reação
$g$	Razão entre a energia de ativação e a constante dos gases ideais
$T$	Temperatura do reator
$T_{Alim}$	Temperatura do fluido de alimentação
$\alpha$	Coefficiente de troca térmica
$A$	Área de troca térmica
$\rho$	Massa específica do conteúdo do reator
$C_p$	Calor específico do conteúdo do reator
$T_c$	Temperatura do fluido da camisa do reator
$\Delta H_r$	Entalpia de reação
$q_c$	Vazão de alimentação de fluido da camisa
$V_c$	Volume da camisa
$T_0$	Temperatura de fluido de alimentação da camisa
$\rho_c$	Massa específica do fluido da camisa
$C_{pc}$	Calor específico do fluido da camisa.

### **Controle MPC**

$C_a$	Concentração de reagente no reato
$J$	Função objetivo ou custo
$k$	Instante de amostragem
$H_c$	Horizonte de controle
$H_p$	Horizonte de predição
$r_i$	Trajectoria de referência i
$y$	Variável de saída
$u$	Ação do controle
$w_i$	Peso da variável manipulada i
$\gamma_i$	Peso da variável controlada i

### ***Rede Neural***

$\gamma_i$	Peso da variável controlada $i$
$\varphi$	Função de ativação do neurônio
$x$	Vetor com entradas
$w$	Vetor com os pesos
$i$	Número de entradas e pesos

### ***Rede Neural Recorrente***

$w_x$	Pesos referente às entradas
$w_y$	Pesos referente às saídas anteriores
$x_t$	Entradas
$y_{t-1}$	Saída do instante anterior
$t$	Instante de tempo
$\varphi$	Função de ativação do neurônio
$W_x$	Matrizes com parâmetros das entradas de toda a rede
$W_y$	Matrizes com parâmetros das saídas do momento anterior de toda a rede

### ***Long short-term memory***

$f_t$	Função de controle do ponto de esquecimento
$i_t$	Função de controle do ponto de entrada
$o_t$	Função de controle do ponto de saída
$h_t$	Termo de curta memória
$c_t$	Termo de longa memória
$W_{xi}$	Matriz referente aos pesos do ponto de entrada
$W_{xf}$	Matriz referente aos pesos do ponto de esquecimento
$W_{xo}$	Matriz referente aos pesos do ponto de saída
$b_i$	Bias do ponto de entrada
$b_f$	Bias do ponto de esquecimento
$b_o$	Bias do ponto de saída
$W_{hi}$	Matriz com os pesos da camada de entrada para conexão com o termo $h_{t-1}$
$W_{hf}$	Matriz com os pesos da camada de esquecimento para conexão com o termo
$h_{t-1}$	
$W_{ho}$	Matriz com os pesos da camada de saída para conexão com o termo $h_{t-1}$
$\varphi$	Função de ativação do neurônio

### ***Gated Recurrent Unit***

$z_t$	Controlador de porta única
$W_{xz}$	Peso referente à conexão com o vetor de entrada $x_t$
$W_{xr}$	Peso referente à conexão com o vetor de entrada $x_t$
$W_{xg}$	Peso referente à conexão com o vetor de entrada $x_t$
$W_{hz}$	Matriz com os pesos das camadas para conexão com o termo $h_{t-1}$
$W_{hr}$	Matriz com os pesos das camadas para conexão com o termo $h_{t-1}$
$W_{hg}$	Matriz com os pesos das camadas para conexão com o termo $h_{t-1}$
$\varphi$	Função de ativação do neurônio
$b_z$	Bias de cada conexão
$b_r$	Bias de cada conexão
$b_g$	Bias de cada conexão

## SUMÁRIO

1	INTRODUÇÃO.....	13
2	OBJETIVOS.....	16
2.1	Objetivo Geral.....	16
2.2	Objetivos específicos .....	16
3	REVISÃO BIBLIOGRÁFICA.....	17
3.1	Controle de processos .....	17
3.2	Controle preditivo com modelo .....	19
3.3	Machine Learning .....	21
3.4	Deep Learning.....	23
3.4.1	Redes Neurais Recorrentes (RNN).....	23
3.4.2	Long Short-Term Memory (LSTM).....	24
3.4.3	Gated Recurrent Unit (GRU).....	26
3.5	Descrição do processo: reação de produção do propileno glicol.....	27
4	METODOLOGIA.....	28
4.1	Modelo do processo .....	29
4.2	Geração de dados para treinamento das redes .....	31
4.3	Treinamento das redes neurais.....	32
4.4	Aplicação do controle preditivo não linear.....	33
5	RESULTADOS E DISCUSSÃO .....	35
5.1	Treinamento das redes .....	35
5.2	Implementação do controle.....	37
5.3	Avaliações gerais dos controladores.....	40
6	CONCLUSÕES .....	42
7	REFERÊNCIAS BIBLIOGRÁFICAS .....	43

## 1 INTRODUÇÃO

Devido à crescente busca pelo desenvolvimento industrial através de processos mais eficientes, a Indústria 4.0 vem ganhando espaço com atuação em diferentes áreas. Na indústria química pode ser destacado o desenvolvimento de processos interligados, com sistemas de controle mais estáveis, eficientes e robustos. Isso contribui para uma maior segurança de operações, adequação a especificações de produtos, assim como aumento na rentabilidade de processos (E.LEBLANC STEVEN, 2009; OGUNNAIKE, BABATUNDE A.; RAY, 1994; STEPHANOPOULOS, 1985).

Até meados dos anos 1940, unidades industriais eram operadas manualmente, sendo muitos operadores necessários para manter processos nas condições ideais de operação, visto a grande quantidade de variáveis operacionais em uma unidade industrial de grande porte. O aumento contínuo no preço da energia elétrica nos anos 1960 fez com que engenheiros químicos intensificassem a aplicação de teorias de análises dinâmicas de controle aos processos químicos (BENENATI, 1973; OGUNNAIKE, BABATUNDE A.; RAY, 1994).

Sistemas de controle atuam de forma a garantir a estabilidade de processos através da supressão de perturbações externas (mudanças em condições do sistema que não são decorrentes de ações do operador ou do sistema de controle) ou através da adequação de todo o sistema a partir de mudanças nos valores de referência (*set-points*). Um processo estável é caracterizado por ser capaz de manter a variável controlada em determinada faixa de operação. Já um sistema instável é algo indesejável dentro do processo, possuindo esta alta tendência a oscilação (geralmente crescente) (STEPHANOPOULOS, 1985).

Para se buscar a estabilidade, diferentes estratégias de controle podem ser adotadas. Em um sistema com controle *feedforward*, o elemento de medição se encontra à montante do processo de modo que o controle se dá de forma antecipativa para a suavização de efeitos das ocasionais perturbações. Já no controle *feedback*, o elemento de medição é localizado na saída do processo, fazendo com que o controlador possa agir a partir dos efeitos da perturbação no sistema (E.LEBLANC STEVEN, 2009).

Os controladores, responsáveis pela comparação do valor da variável medida com o valor de referência para atuação no elemento final de controle, podem ser baseados em técnicas clássicas. Controladores clássicos podem atuar de forma proporcional (P), proporcional-integral (PI), proporcional-derivativo (PD) ou proporcional-integral-derivativo (PID) aos desvios do valor de referência (E.LEBLANC STEVEN, 2009).

Devido a não linearidades em sistemas de alta complexidade, tem crescido a utilização do Controle Preditivo baseado em Modelo (MPC), desenvolvido nos anos 1970 por duas empresas de pesquisas industriais. No MPC as ações do controlador são tomadas através de um modelo dinâmico do processo e medições de determinado instante são utilizadas para uma estimativa do comportamento futuro do sistema (SEBORG, 2012).

Algoritmos de *Machine Learning* (ML), como Redes Neurais Artificiais (RNA), têm ganho atenção no contexto Indústria 4.0 em uma gama de aplicações, em especial na utilização em sensores virtuais e modelos preditivos para MPC. Algoritmos de ML são capazes de aprender padrões em um determinado conjunto de dados para maior acurácia na predição de valores futuros. Porém, para lidar com a dificuldade de generalização e a não aplicabilidade em determinados sistemas, na literatura especializada atual pode-se perceber a crescente investigação de estratégias utilizando *Deep Learning* (DL). DL são estratégias de ML caracterizado por possuir estruturas de aprendizado com maior densidade de neurônios para representação mais efetiva de funções de alta complexidade (HERMIYANTY, WANDIRA AYU BERTIN, 2017; KITTISUPAKORN *et al.*, 2009; PON KUMAR *et al.*, 2018).

Estruturas de *Deep Learning* com maior capacidade preditiva em dados sequenciais são as Redes Neurais Recorrentes (do inglês RNN), em que há a retroalimentação da saída pelo modelo a cada instante. Devido a mudanças nos dados através da RNN, informações podem ser esquecidas a cada unidade de tempo. Para lidar com esse problema, foi proposta uma rede recorrente de longa memória chamada LSTM (*long short-term memory*) com capacidade de detectar dependências de longos termos nos dados através do filtro de informações no decorrer do tempo (GÉRON, 2019).

Uma célula recorrente de longa memória que tem sido bastante aplicada atualmente é a *Gated Recurrent Unit* (GRU) que difere da LSTM pela necessidade de se apagar dados do local de armazenamento antes da adição de novas informações. Esse comportamento favorece a adoção de critérios exigentes na seleção, sendo um filtro de informações mais robusto (GÉRON, 2019).

Atualmente a aplicação dos algoritmos de *Deep Learning* citados são majoritariamente realizadas utilizando a linguagem de programação *Python*, que vem ganhando destaque na indústria e na comunidade científica com crescente utilização a partir dos anos 2000 por ser de fácil utilização, gratuita e por possuir bibliotecas com código aberto que possibilitam sua aplicação em diferentes tipos de problemas (TOBERGTE; CURTIS, 2013).

A garantia da estabilidade de reatores CSTR (*continous stirred-tank reactor*) é um problema presente na literatura e altamente utilizado em estudos de estratégias de controle, principalmente quando são empregados em reações complexas e não lineares como a de produção do propileno glicol. A não linearidade do sistema de controle faz com que esse problema seja de alta complexidade de modo que controladores clássicos não garantem a estabilidade do sistema. (FOGLER, 2011).

Desta forma, neste trabalho serão investigadas estratégias para construção de controlador MPC utilizando algoritmos de *Deep Learning* como modelo interno a partir da linguagem de programação *Python* e avaliação das estratégias a partir da aplicação no *benchmarking* de produção do propileno glicol em reator CSTR devido a sua vasta utilização industrial e à importância do controle de temperatura para a reação.

## 2 OBJETIVOS

### 2.1 Objetivo Geral

Desenvolver, implementar e avaliar estratégias de controle utilizando MPC baseado em *Deep Learning* utilizando a linguagem de programação *Python*. Investigar a funcionalidade das estratégias desenvolvidas e comparar as performances através de aplicação em um sistema reativo não linear.

### 2.2 Objetivos específicos

- Modelagem fenomenológica do sistema reativo para a produção do propileno glicol em CSTR através de balanços de massa e energia;
- Desenvolvimento de estratégias de controle preditivo baseadas em *Deep Learning* utilizando Redes Neurais Recorrentes (RNN), *Long Short-term Memory* (LSTM) e *Gated Recurrent Unit* (GRU) a partir da linguagem de programação *Python*;
- Implementação das estratégias a um sistema de controle de temperatura para a reação de produção do propileno glicol utilizando a linguagem de programação *Python*;
- Investigação e avaliação da aplicabilidade das estratégias de controle preditivo desenvolvidas;
- Verificação das condições de estabilidade do sistema a partir da aplicação das estratégias de controle;
- Comparação da performance das estruturas de *Deep Learning* e dos pesos do controlador;
- Capacitação de conteúdos ligados à área de controle de processos.

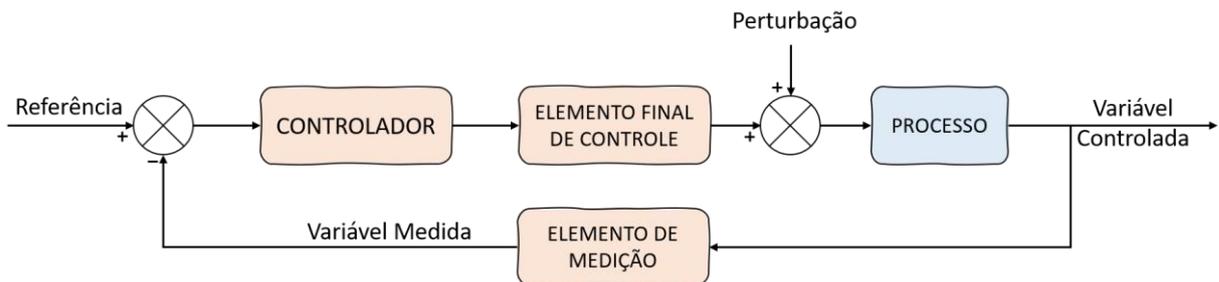
### 3 REVISÃO BIBLIOGRÁFICA

#### 3.1 Controle de processos

Para que a operação de processos químicos industriais aconteça de forma estável, estratégias de controle são aplicadas, tratando assim desvios do estado estacionário (SEBORG, DALE E.; *et. al*, 2019). Em um sistema de controle automático do tipo *feedback*, são essenciais elementos de medição, utilizados para a amostragem das variáveis de processo; controladores, que recebem as informações medidas, realizam comparações com o valor de referência (*set-point*) e enviam informações de tomadas decisão para o elemento final de controle – Figura 1 (E.LEBLANC STEVEN, 2009).

Quando o sistema de controle atua em malha aberta, o sinal enviado ao elemento final de controle é ajustado manualmente. Já em malha fechada, o valor de saída (variável controlada) é comparado a um valor de referência para o ajuste a partir das variáveis manipuladas (E.LEBLANC STEVEN, 2009; SILVA, 2018).

**Figura 1– Estratégia de controle *feedback***



Fonte: Coughanour (2009)

Quando há alteração no valor de referência (*set-point*), o sistema de controle deve agir de modo a minimizar o erro (diferença entre o valor de referência e o valor obtido a partir do elemento de medição), caracterizando um problema do tipo “servo”. O mesmo ocorre em caso de perturbações do sistema (mudança em condições do sistema que não são fruto de ações do operador ou do controlador), sendo caracterizado como um problema regulatório (STEPHANOPOULOS, 1985).

As ações dos controladores podem se dar de acordo com técnicas clássicas através do envio de informações para o elemento final de controle de forma proporcional, integral ou derivativa ao erro. Porém as técnicas clássicas apresentam limitações em sistemas com não linearidades e com interações multivariáveis, além da alta complexidade de sintonia (obtenção de parâmetros ótimos para se atingir condições de estabilidade). A fim de contornar tais condições, podem ser

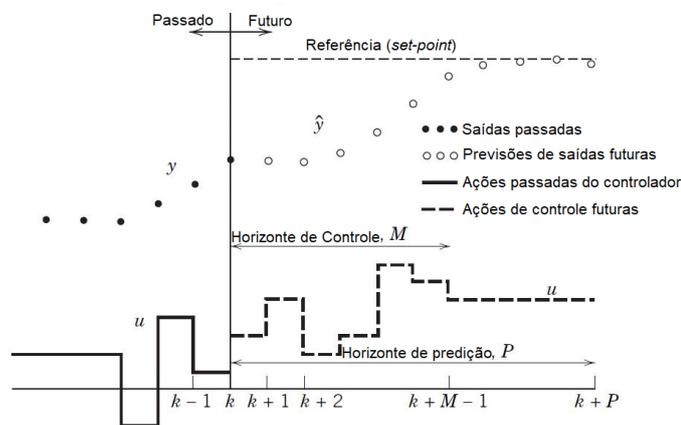
utilizados controladores preditivos com modelo (MPC) ( KITTISUPAKORN et al., 2009; SEBORG et al., 2012).

### 3.2 Controle preditivo com modelo

Os controladores preditivos com modelo (MPC) fazem parte de um conjunto de métodos que utilizam modelos de processo, atuando a partir da obtenção de estimativas da saída do sistema (variável controlada) em um horizonte de tempo futuro para que o controlador atue com maior assertividade (SEBORG, 2012; SILVA, 2019, 2018).

As ações futuras do controlador para o horizonte tempo  $M$  (horizonte de predição) são previstas a cada instante  $k$  usando um modelo do processo - Figura 2. As ações de controle previstas são obtidas a partir de otimização (minimização do erro) mantendo assim o sistema próximo da trajetória de referência (CAMACHO; BORDONS, 1999; SEBORG, 2012).

**Figura 2 – Ações de controlador MPC**



Fonte: Seborg (2012).

De acordo com SARAIVA (2017) e SILVA (2014), controladores MPC são compostos por:

- **Trajétória de referência:** composta pelo conjunto de saídas desejadas do sistema (variável controlada) a cada instante de tempo.
- **Modelo de Predição:** Estima a dinâmica futura do processo ao longo de um horizonte determinado com base em seu comportamento passado.
- **Algoritmo de otimização:** Minimiza uma função objetivo a cada instante de tempo, de forma a tomar a ação mais adequada para garantir a proximidade da variável controlada (do inglês – CV) em relação à trajetória de referência.

As ações de controle são tomadas de forma a manter o processo o mais próximo possível da trajetória de referência. A lei de controle é obtida a partir da minimização da função objetivo – (1. A função objetivo verifica a diferença entre a saída predita ( $\hat{y}$ ) e a trajetória de referência ( $r$ ) para enviar sinal ótimo ao controlador ( $u$ ).

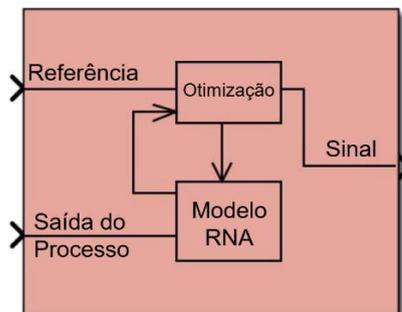
$$J = w \sum_{j=1}^{Hp} [\hat{y}(k+j) - r(k+j)]^2 + \gamma \sum_{i=1}^{Hc} \Delta u^2(k+i-1) \quad (1)$$

Onde  $w$  e  $\gamma$  representam os pesos atribuídos ao erro de previsão e variação de ação de controle, respectivamente. O parâmetro  $\gamma$  está ligado à suavização da ação de controle e o  $w$  à agilidade dessa ação.  $Hp$  e  $Hc$  representam os horizontes de controle e de previsão.

Segundo SEBORG (2012), a eficácia do MPC depende principalmente da acurácia do modelo de processo utilizado. Predições errôneas podem fazer com que os resultados deixem a desejar, gerando um comportamento oscilatório no sistema (E.LEBLANC STEVEN, 2009).

Atualmente é encontrado uma grande quantidade de trabalhos com foco em MPC a partir da utilização de modelos preditivos com base em algoritmos de aprendizado de máquina (*Machine Learning*). KITTISUPAKORN *et al.* (2009) e SILVA (2018) abordaram problemas de controle utilizando MPC baseados em Redes Neurais Artificiais (algoritmo de *Machine Learning*) - Figura 3 - aplicados a problemas de controle em reatores CSTR, obtendo assim comportamentos estáveis nos diferentes estudos de caso.

**Figura 3 – Controlador preditivo utilizando algoritmos de *Machine Learning***



Fonte: Mathworks (2017).

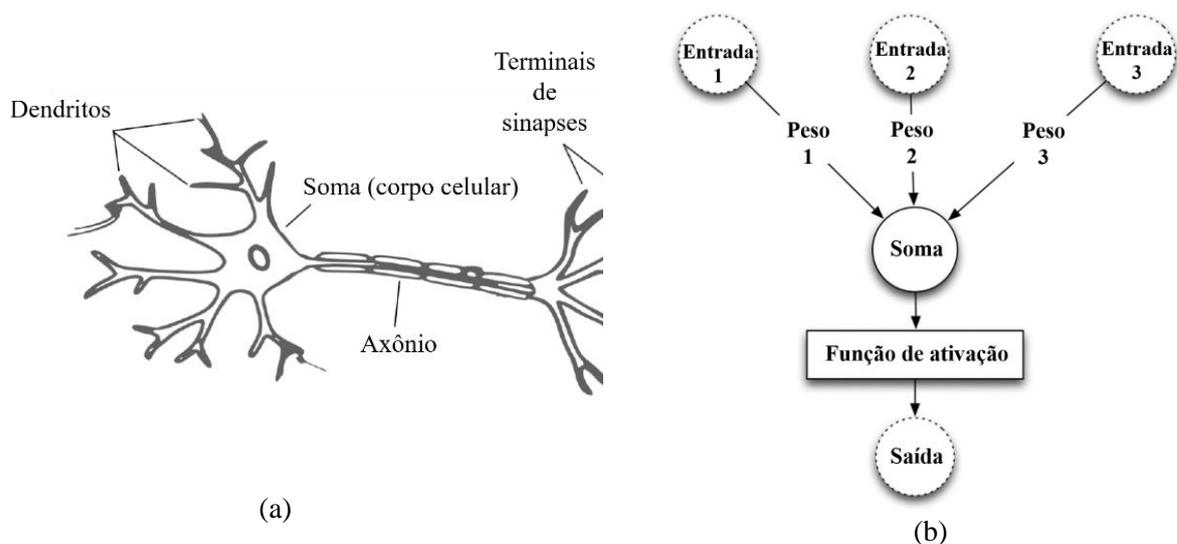
### 3.3 Machine Learning

Algoritmos de *Machine Learning* têm se tornado cada vez mais populares, sendo possível o aprendizado de padrões a partir de características peculiares em um conjunto de dados. Isso acontece a partir da programação para o aprendizado com relação a experiências de um sistema ou processo, tendo uma forma de medição de performance desse aprendizado (GÉRON, 2019).

As principais formas de aprendizado adotadas em um treinamento de algoritmos de *Machine Learning* são o aprendizado não-supervisionado, quando os dados de saída (*target*) não estão definidos, de modo que o algoritmo busca padrões e segregações a partir dos dados de entrada apresentados; e supervisionado, quando o algoritmo é treinado utilizando os dados de entrada e os valores de saída correspondentes (*target*) (GÉRON, 2019).

Um dos algoritmos de *Machine Learning* mais utilizados para problemas de regressão e análise de problemas com dependência temporal (aprendizado supervisionado) são as Redes Neurais Artificiais (RNA). As RNA's são representações através de modelos matemáticos de neurônios biológicos (células encontradas no cérebro de animais). Os neurônios são compostos de dendritos, responsáveis pela recepção dos sinais; corpo celular, onde todos os sinais recebidos pelos dendritos são somados para a formação de um sinal de entrada e axônios, que transmitem os sinais gerados aos terminais de sinapse - Figura 4a. Os terminais de sinapse se conectam a outros dendritos formando assim uma rede neural Figura 5 - (GÉRON, 2019; SILVA, 2018).

**Figura 4 – (a) Estrutura celular de um neurônio biológico e (b) forma simplificada de um neurônio artificial (perceptron)**



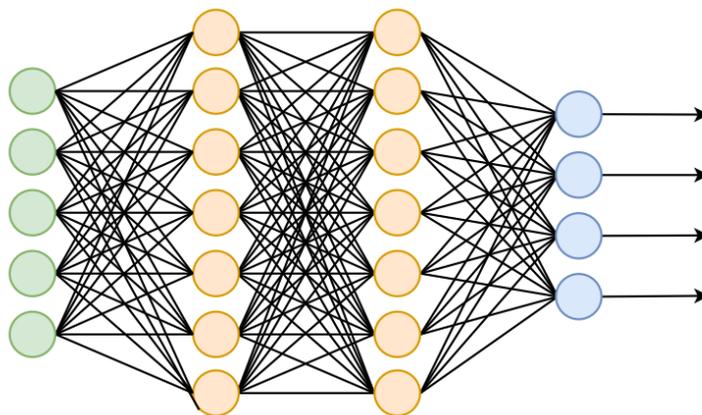
Fonte: Silva (2018) e Heaton (2012)

A forma mais simples de representação matemática de um neurônio biológico é chamada de perceptron - Figura 4b. Nele, as entradas fornecidas são multiplicadas por pesos para serem inseridas em uma função de ativação. Na Equação (2) há a representação de como as saídas são obtidas. As variáveis  $x$  e  $w$  representam as entradas e os pesos (respectivamente). A variável  $i$  corresponde ao número de entradas e pesos. Cada entrada é multiplicada pelo seu peso correspondente e o resultado da multiplicação é alimentado a uma função de ativação, representada pela letra  $\varphi$ . Esse processo resulta em uma única saída do neurônio (GÉRON, 2019; GRAUPE, [s.d.]; HEATON, 2012).

$$f(x_i, w_i) = \varphi \left( \sum_i (x_i \cdot w_i) \right) \quad (2)$$

Como o perceptron demonstrava pouca acurácia em problemas lógicos simples, foi proposta a união de vários perceptrons para a formação de uma Rede Neural Artificial chamada Perceptrons em Multicamadas (MLP - *Multilayer Perceptron*). Essa estrutura permitiu a resolução de diferentes tipos de problemas com boa acurácia. Porém, devido à necessidade da utilização de estratégias mais robustas para problemas com maior complexidade, podem ser utilizadas estruturas com muitas camadas, caracterizando uma estrutura de aprendizado profundo (*Deep Learning*) (GÉRON, 2019).

**Figura 5 – Redes Neurais em Multicamada (*Multilayer Perceptron* - MLP). Em verde a camada de entrada de entrada; em azul, a camada de saída; e em laranja, as camadas ocultas**



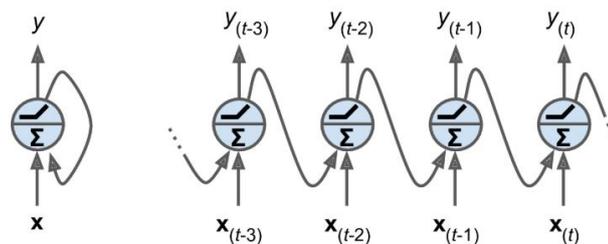
Fonte: SILVA (2018)

### 3.4 Deep Learning

#### 3.4.1 Redes Neurais Recorrentes (RNN)

Estruturas de *Deep Learning* são altamente versáteis, poderosas e escaláveis, sendo possível para aplicação em problemas de *Machine Learning* de alta complexidade. Um tipo de célula de aprendizado que pode ser utilizado em multicamadas para aplicação de técnicas de *Deep Learning* são as Redes Recorrentes (RNN). As RNN's são utilizadas para estimativa de condições futuras em um conjunto de dados sequenciais. Nelas, as saídas da estrutura de aprendizado dos instantes anteriores  $y_{t-1}$  dão entrada nas células junto com os dados de entrada  $x_t$  a cada instante de tempo, levando consigo informações de toda a cadeia de dados (dependência temporal) - Figura 6 (GÉRON, 2019).

**Figura 6 - Célula recorrente a cada instante de tempo**



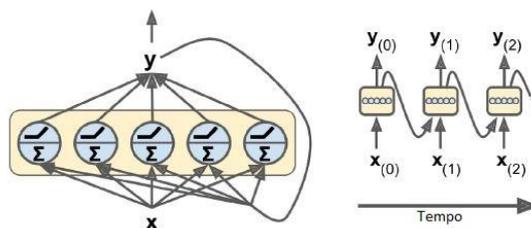
Fonte: Géron (2019)

Nas redes recorrentes, cada neurônio é composto por dois conjuntos de parâmetros ( $w_x$  e  $w_y$ ): um para as entradas ( $x_t$ ) e outros para a saída do instante anterior ( $y_{t-1}$ ). A saída da rede no instante  $t$  é calculada a partir da Equação (3).

$$y_t = \varphi(W_x^T x_t + W_y^T y_{t-1} + b) \quad (3)$$

Sendo  $b$  um vetor com bias,  $\varphi$  a função de ativação,  $W_x$  e  $W_y$  matrizes com os parâmetros das entradas e saídas de toda a rede (não apenas de um neurônio). A seguir é apresentado uma estrutura de rede recorrente - Figura 7.

**Figura 7 – Rede Recorrente a cada instante de tempo**



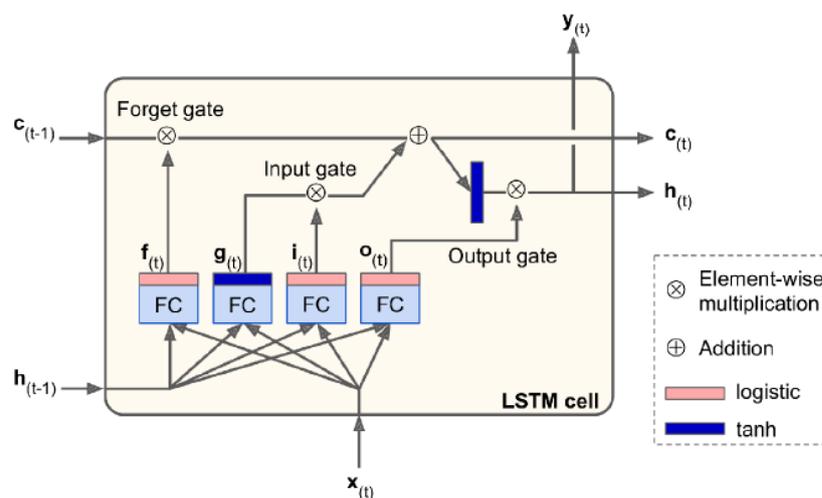
Fonte: Géron (2019)

### 3.4.2 Long Short-Term Memory (LSTM)

Quando as redes recorrentes processam longas sequências de dados, informações das primeiras entradas são perdidas a cada instante de tempo devido a transformações nos dados. A fim de contornar a situação, foram introduzidas células de longa memória (*Long Short-Term Memory* - LSTM) em 1997 por Sepp Hochreiter. A estrutura de realimentação é muito similar à RNN, porém à medida que a estrutura é treinada, informações menos importantes são perdidas através de uma conexão para esquecimento (*forget gate*) e há a manutenção de informações relevantes através da conexão de entrada para o próximo instante (*input gate*) - Figura 8. A cada instante, novas memórias são adicionadas e outras (menos importantes) são esquecidas da rede, sendo possível uma predição de comportamentos futuros através dessa estrutura (GÉRON, 2019).

Observando a Figura 8, pode-se verificar que há grande semelhança nas entradas e saídas da célula (ignorando o interior da “caixa”) em relação à RNN, porém com a adição dos termos de curta  $h_t$  e de longa memória  $c_t$  (GÉRON, 2019).

**Figura 8 – Célula da Rede Recorrente de Longa Memória (LSTM)**



Fonte: Géron (2019)

- O ponto de esquecimento (controlada por  $f_t$ ) controla quais partes dos termos de longa memória deverão ser apagados;
- O ponto de entrada (controlada por  $i_t$ ) controla quais partes de  $g_t$  deverão ser adicionadas ao termo de longa memória;
- O ponto de saída (controlada por  $o_t$ ) controla quais partes do termo de longa memória deverão ser lidos e direcionados à saída da célula sob a forma dos ramos  $y_t$  e  $h_t$  (GÉRON, 2019).

Os cálculos das saídas e dos ramos de longa e curta memória são feitos de acordo com as Equações (4 a (9).

$$i_t = \varphi(W_{xi}^T x_t + W_{hi}^T h_{t-1} + b_i) \quad (4)$$

$$f_t = \varphi(W_{xf}^T x_t + W_{hf}^T h_{t-1} + b_f) \quad (5)$$

$$o_t = \varphi(W_{xo}^T x_t + W_{ho}^T h_{t-1} + b_o) \quad (6)$$

$$g_t = \tanh(W_{xg}^T x_t + W_{hg}^T h_{t-1} + b_g) \quad (7)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \quad (8)$$

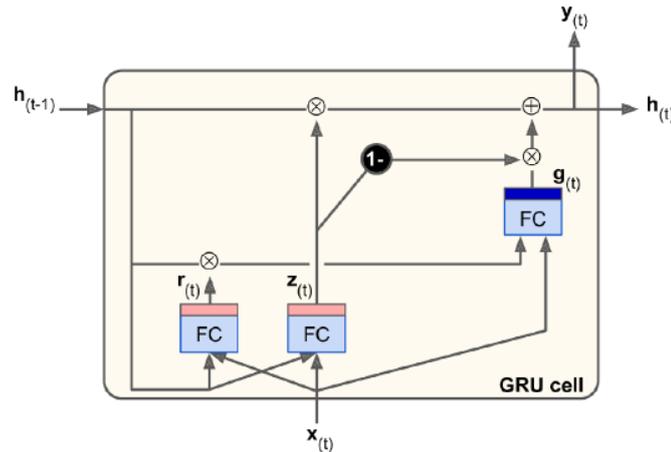
$$y_t = h_t = o_t \otimes \tanh(c_t) \quad (9)$$

Sendo,  $W_{xi}, W_{xf}, W_{xo}, W_{xg}$  matrizes referentes aos pesos de cada uma das quatro conexões ao vetor de entrada  $x_t$ .  $W_{hi}, W_{hf}, W_{ho}, W_{hg}$  são matrizes com os pesos de cada uma das quatro camadas para conexão com o termo de curta memória  $h_{t-1}$ . Os bias de cada conexão são representador por  $b_i, b_f, b_o,$  e  $b_g$ .  $\varphi$  representa a função de ativação.

### 3.4.3 Gated Recurrent Unit (GRU)

A célula *Gated Recurrent Unit* (GRU) - Figura 9 - foi proposta em 2014 por Kyunghyun Cho *et al.*, como sendo uma versão simplificada da célula LSTM. A sua crescente popularidade tem base na simplicidade da estrutura e na capacidade de filtrar informações relevantes com relação à série temporal (GÉRON, 2019).

Figura 9 – Célula Gated Recurrent Unit (GRU)



Fonte: Géron (2019)

Quando comparado à LSTM, há simplificações na célula GRU que devem ser levadas em consideração, sendo as principais:

- Os vetores de estado ( $h_t$  e  $c_t$ ) são fundidos em um único vetor  $h_t$ ;
- Um controlador de porta única  $z_t$  controla o ponto de esquecimento e o ponto de entrada. Se o controlador de porta gera 1, a porta de esquecimento está aberta ( $= 1$ ) e a porta de entrada é fechada ( $1 - 1 = 0$ ). Se a saída for 0, o oposto acontece. Em outras palavras, sempre que há a necessidade de se armazenar uma memória, o local de armazenamento é apagado primeiro (GÉRON, 2019).

A forma matemática da célula GRU é descrita conforme as Equações (10 a (13).

$$z_t = \varphi(W_{xz}^T x_t + W_{hz}^T h_{t-1} + b_z) \quad (10)$$

$$r_t = \varphi(W_{xr}^T x_t + W_{hr}^T h_{t-1} + b_r) \quad (11)$$

$$g_t = \tanh(W_{xg}^T x_t + W_{hg}^T (r_t \otimes h_{t-1}) + b_g) \quad (12)$$

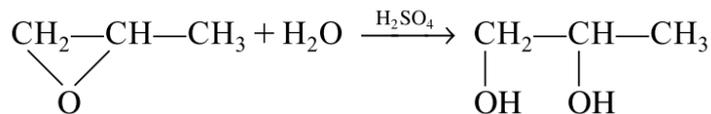
$$h_t = z_t \otimes h_{t-1} + (1 - z_t) \otimes g_t \quad (13)$$

Sendo,  $W_{xz}, W_{xr}, W_{xg}$  matrizes referentes aos pesos de cada uma das três conexões ao vetor de entrada  $x_t$ .  $W_{hz}, W_{hr}, W_{hg}$  são matrizes com os pesos de cada uma das três camadas para conexão com o termo de longa memória  $h_{t-1}$ . Os bias de cada conexão são representados por  $b_z, b_r, b_g$ .  $\varphi$  representa a função de ativação.

### 3.5 Descrição do processo: reação de produção do propileno glicol

O propileno glicol é um componente com vasta aplicação, sendo utilizado principalmente na produção de medicamentos, cosméticos, produtos alimentícios e sanitizantes (agentes antibacterianos). Ele pode ser obtido a partir da hidrólise do óxido de propileno catalisada com ácido sulfúrico – Figura 10 (FOGLER, 2011).

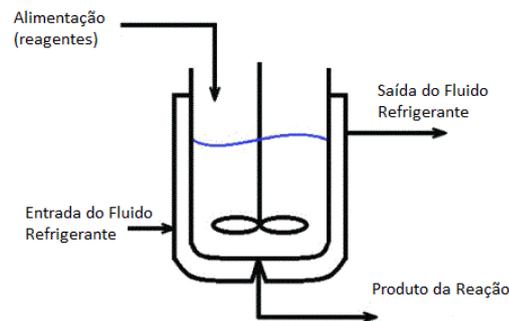
**Figura 10 – Reação de produção do propileno glicol**



Fonte: Fogler (2016)

A reação é exotérmica e pode acontecer em um reator do tipo CSTR (*continuous stirred tank reactor*) - Figura 11 - com temperatura controlada a partir da circulação de fluido refrigerante em sua camisa.

**Figura 11 – Reator contínuo de mistura perfeita (CSTR)**



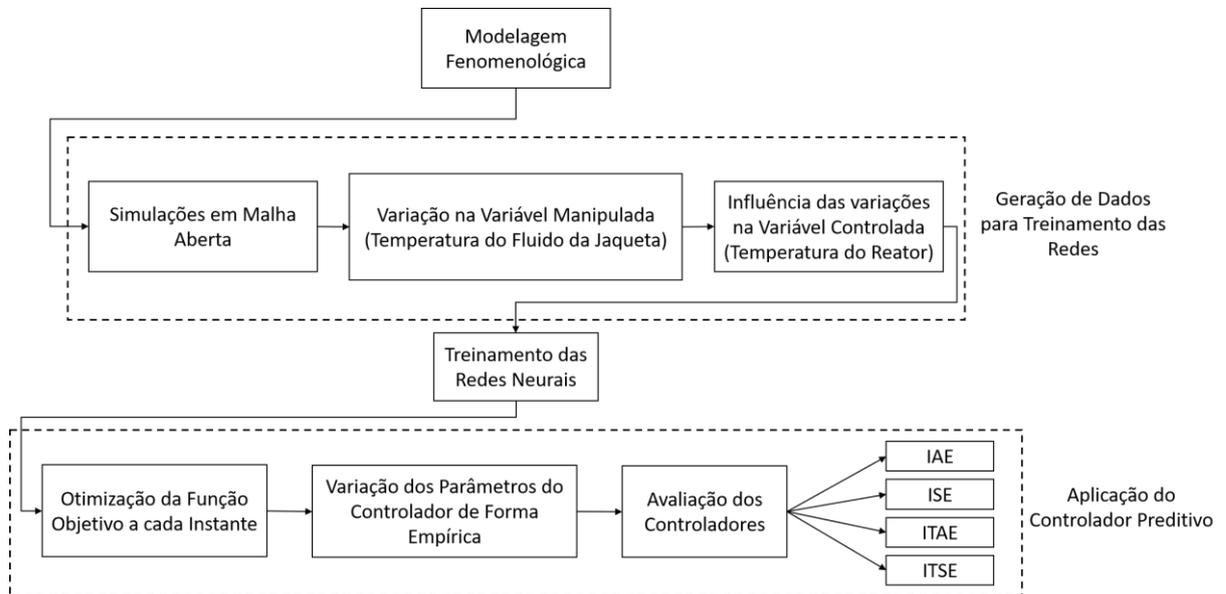
Fonte: Utkin e Gulyukina (2020)

Estratégias de controle têm sido amplamente estudadas com utilização da reação de produção do propileno glicol como *benchmarking* por se caracterizar de extrema dificuldade de obtenção de um sistema de resfriamento (circulação de fluido na camisa do reator) estável. A não linearidade do sistema faz com que se busque a utilização de estratégias mais robustas, como o controle preditivo com base em modelos de *Deep Learning*.

#### 4 METODOLOGIA

A seguir é apresentado uma compilação da metodologia utilizada no trabalho - Figura 12.

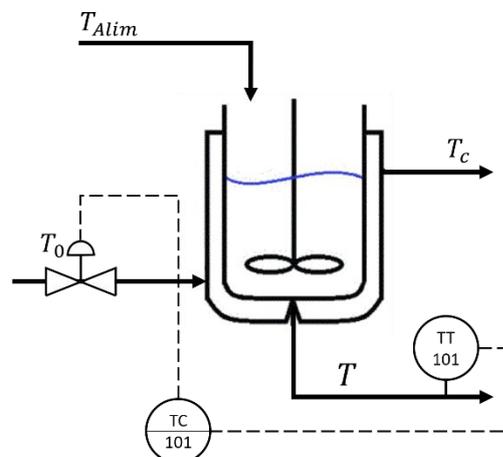
**Figura 12 – Metodologia adotada no trabalho para construção do controlador preditivo**



Fonte: O Autor (2020)

As estratégias de controle desenvolvidas no trabalho serão testadas com base na análise do comportamento dinâmico do sistema para uma malha de controle com uma entrada e uma saída – Controle SISO (*Single Input Single Output*) - Figura 13. A variável controlada será a Temperatura do conteúdo do reator, enquanto a variável manipulada, a Temperatura de alimentação do fluido da jaqueta.

**Figura 13 – Esquema do sistema de controle estudado**



Fonte: O Autor (2020)

#### 4.1 Modelo do processo

O modelo do processo foi obtido a partir de balanço material e energético, com base no estudo desenvolvido por (KMETOVA; VASICKANINOVA; DVORAN, 2013). Na modelagem fenomenológica, foi considerado que:

- Os reagentes são perfeitamente misturados no reator (mistura perfeita);
- O calor específico, o coeficiente global de transferência térmica e a massa específica são constantes no decorrer da reação;
- Existe um controle ideal do nível do reator (SARAIVA, 2017).

Com base nas considerações anteriores, a dinâmica do sistema pode ser descrita pelas equações diferenciais a seguir – Equações (14 a (16.

$$\frac{dC_a}{dt} = \frac{q}{V}(C_{Alim} - C_a) - k_0 e^{-g/T} C_a \quad (14)$$

$$\frac{dT}{dt} = \frac{q}{V}(T_{Alim} - T) - \frac{\alpha A}{V\rho C_p}(T - T_c) + \frac{1}{\rho C_p} k_0 e^{-g/T} C_a (-\Delta H_r) \quad (15)$$

$$\frac{dT_c}{dt} = \frac{q_c}{V_c}(T_0 - T_c) + \frac{\alpha A}{V_c \rho_c C_{pc}}(T - T_c) \quad (16)$$

Onde  $C_a$  representa a concentração de reagente no reator,  $q$  a vazão de alimentação,  $V$  o volume do reator,  $C_{Alim}$  a concentração do reagente na alimentação,  $k_0$  o fator pré-exponencial,  $g$  é a razão entre a energia de ativação e a constante dos gases ideais,  $T$  a temperatura do reator,  $T_{Alim}$  a temperatura do fluido de alimentação,  $\alpha$  o coeficiente de troca térmica,  $A$  área de troca térmica,  $\rho$  a massa específica do conteúdo do reator,  $C_p$  o calor específico do conteúdo do reator, a  $T_c$  temperatura do fluido da camisa do reator,  $\Delta H_r$  a entalpia de reação,  $q_c$  a vazão de alimentação de fluido da camisa,  $V_c$  o volume da camisa,  $T_0$  a temperatura de fluido de alimentação da camisa,  $\rho_c$  a massa específica do fluido da camisa e  $C_{pc}$  o calor específico do fluido da camisa.

Além da modelagem, KMETOVA; VASICKANINOVA e DVORAN (2013) apresentaram parâmetros da reação e propriedades do sistema, disponíveis nas Tabelas Tabela 1, Tabela 2, Tabela 3 e Tabela 4.

**Tabela 1 – Cinética de reação considerada no estudo**

Parâmetros cinéticos da reação		
$k_0$	$2.82E + 11$	$min^{-1}$
$\Delta H_r$	$-5.46E + 06$	$kJ/kmol$

Fonte: kmetova; vasickaninova; Dvoran (2013)

**Tabela 2 – Parâmetros físico-químicos considerados no estudo**

Parâmetros Físico-Químicos		
$\rho$	947.19	$kg/m^3$
$\rho_c$	998	$kg/m^3$
$C_p$	3.7187	$kJ/(kg * K)$
$C_{pc}$	4.182	$kJ/(kg * K)$
$\alpha$	13.8	$kJ/(min * m^2 * K)$
$g$	10183	$K$

Fonte: kmetova; vasickaninova; Dvoran (2013)

**Tabela 3 – Dimensões do reator considerado no estudo**

Dimensões do reator		
$V$	2.407	$m^3$
$V_c$	2	$m^3$
$A$	8.695	$m^2$

Fonte: kmetova; vasickaninova; Dvoran (2013)

**Tabela 4 – Variáveis de processo consideradas no estudo.**

Variáveis do processo		
$T_{alim}$	299.05	$K$
$T_0$	288.15	$K$
$C_{alim}$	0.0824	$kmol/m^3$
$q$	0.072	$m^3/min$
$q_c$	0.6307	$m^3/min$

Fonte: kmetova; vasickaninova; Dvoran (2013)

## 4.2 Geração de dados para treinamento das redes

Os dados a serem utilizados para o treinamento das estruturas de *Machine* e *Deep Learning* foram obtidos a partir de simulações em malha aberta através da variação aleatória da variável manipulada (temperatura de alimentação do fluido de resfriamento) utilizando a linguagem de programação *Python* (SILVA, 2014).

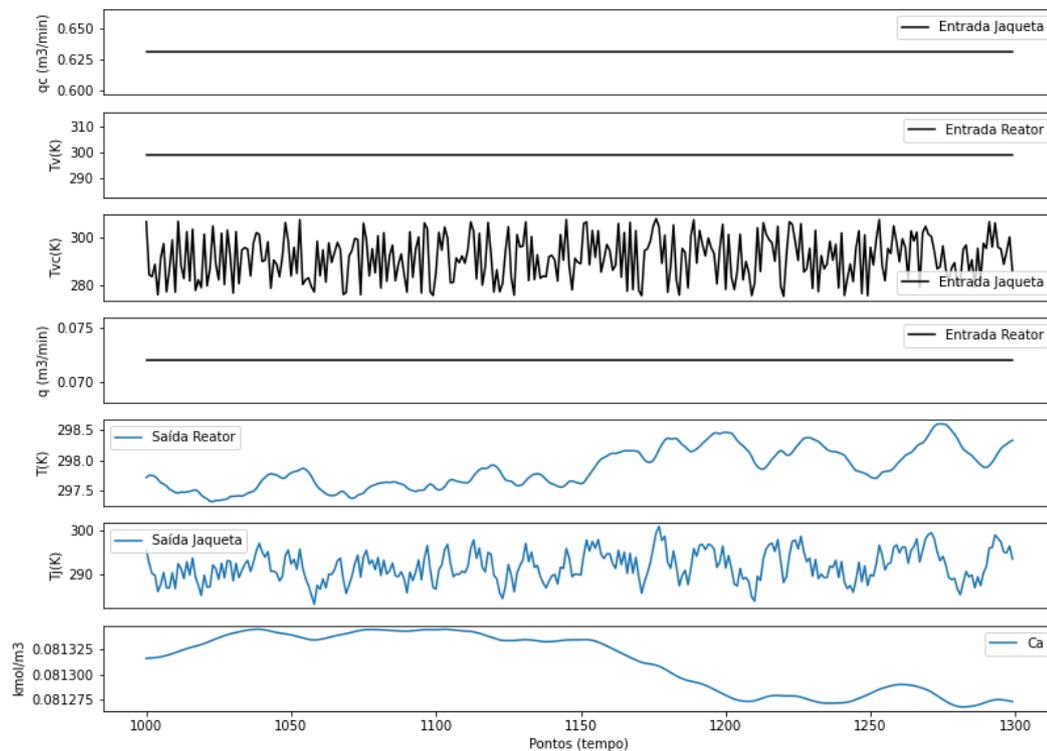
**Tabela 5 - Limite das variáveis utilizadas na geração de dados de treinamento**

Variável	Limite Inferior	Limite Superior	Unidade
$T_0$	275.15	308.25	$K$

Fonte: O Autor (2020)

A geração de dados ocorreu de acordo com os parâmetros dispostos nas Tabela 2 a Tabela 5. Foram gerados 40000 segundos de operação do sistema, com comportamento disposto na Figura 14.

**Figura 14 – Dados gerados entre os instantes 1000s e 1300s**

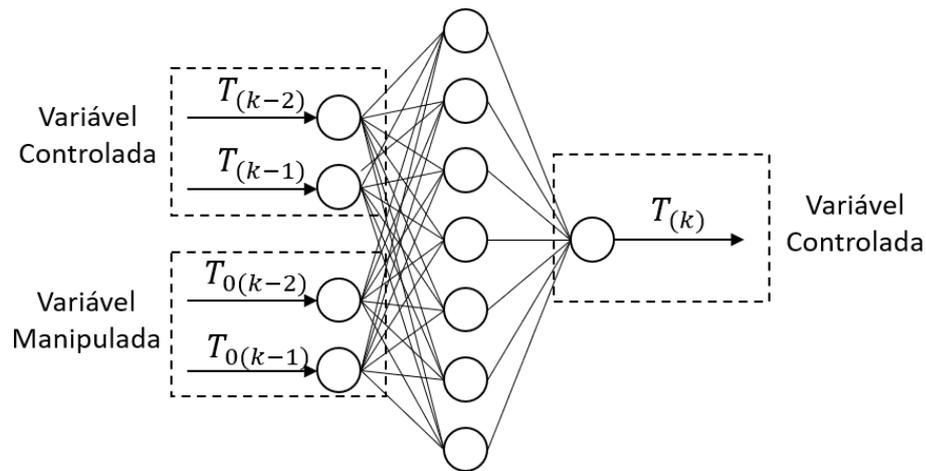


Fonte: O Autor (2020)

### 4.3 Treinamento das redes neurais

Na etapa de treinamento, foram utilizados como entrada dados de dois instantes anteriores ( $k - 2$ ) e ( $k - 1$ ) da variável manipulada (temperatura de entrada do fluido de circulação da camisa do reator) e da variável controlada (temperatura do reator). A saída da rede foi composta pela variável controlada (temperatura do reator) no instante atual ( $k$ ) - Figura 15.

**Figura 15 – Estrutura de treinamento das células de *Machine e Deep Learning***



Fonte: O Autor (2020)

No treinamento foram utilizados 80% dos dados gerados, sendo 20% utilizados para teste. O treinamento foi realizado utilizando a biblioteca *Keras* disponível na linguagem de programação *Python* com utilização das métricas de erro dispostas nas Equações (17) e (18).

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y - \hat{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (17)$$

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} |y - \hat{y}| \quad (18)$$

Sendo  $R^2$  o coeficiente de determinação,  $\hat{y}_i$  o valor predito da amostra  $i$ ,  $y_i$  o valor real correspondente num total de  $n$  dados com valor médio  $\bar{y}$ . MAE representa o erro médio absoluto.

#### 4.4 Aplicação do controle preditivo não linear

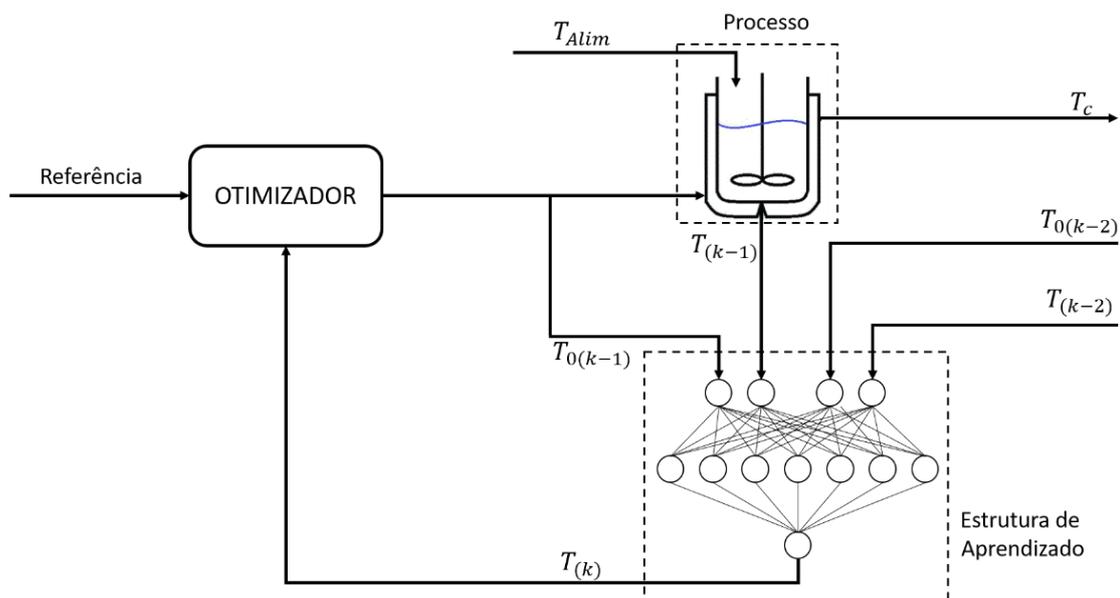
O controlador preditivo utilizado foi idealizado conforme o esquema da Figura 16 com base no estudo desenvolvido por (SILVA, 2014). A partir da otimização da função objetivo a cada instante – Equação (19), foi avaliado a estabilidade do sistema a partir de variações no *set-point*, sendo comparadas as performances do controlador a partir das estruturas de rede utilizadas (*Machine e Deep Learning*).

$$J = w_T \sum_{j=1}^{H_p} [\hat{T}(k+j) - r(k+j)]^2 + w_{T_0} \sum_{i=1}^{H_c} \Delta T_0^2(k+i-1) \quad (19)$$

Na otimização, foi utilizado o método *Particle Swarm Optimization* (PSO) – Otimização por Enxame de Partículas. Este método não será detalhado no trabalho por ter sido abordado de forma detalhada por Soares (2019), com estudo desenvolvido no Laboratório de Sistemas Inteligentes Aplicados.

De forma resumida, a PSO funciona de acordo com uma geração aleatória inicial de partículas que se movem no espaço de soluções na busca pelo ótimo global. Esta movimentação acontece a partir da atualização da velocidade e da posição das partículas com base em suas melhores posições individuais e a melhor posição global (SOARES, 2019).

**Figura 16 - Estrutura do controlador preditivo utilizado no estudo**



Fonte: O Autor (2020)

Os pesos do controlador  $w_T$  e  $w_{T_0}$  também foram variados de forma empírica a fim de verificar a melhor condição para cada estrutura (DUTRA, 2003). Os pesos que resultaram na melhor performance foram avaliados quanto à estrutura de aprendizado. Vale salientar que utilização de ponderações com maior valor não garantem que o objetivo (convergência rápida ao SP ou variações bruscas da variável manipulada) tenha prioridade na obtenção da solução do problema (DUTRA, 2003).

As estratégias de controle preditivo desenvolvidas foram avaliadas pelas métricas: Integral do Erro Absoluto (IAE - *Integral of the absolute value of error*), Integral do Erro Absoluto Ponderado pelo Tempo (ITAE - *integral of time-weighted absolute error*) e Integral do Erro Quadrático Ponderado pelo Tempo (ITSE - *integral of time-weighted squared error*) para a comparação entre os resultados.

$$IAE = \int |e(t)| dt \quad (20)$$

$$ISE = \int e^2(t) dt \quad (21)$$

$$ITAE = \int t|e(t)| dt \quad (22)$$

$$ITSE = \int t|e(t)|^2 dt \quad (23)$$

## 5 RESULTADOS E DISCUSSÃO

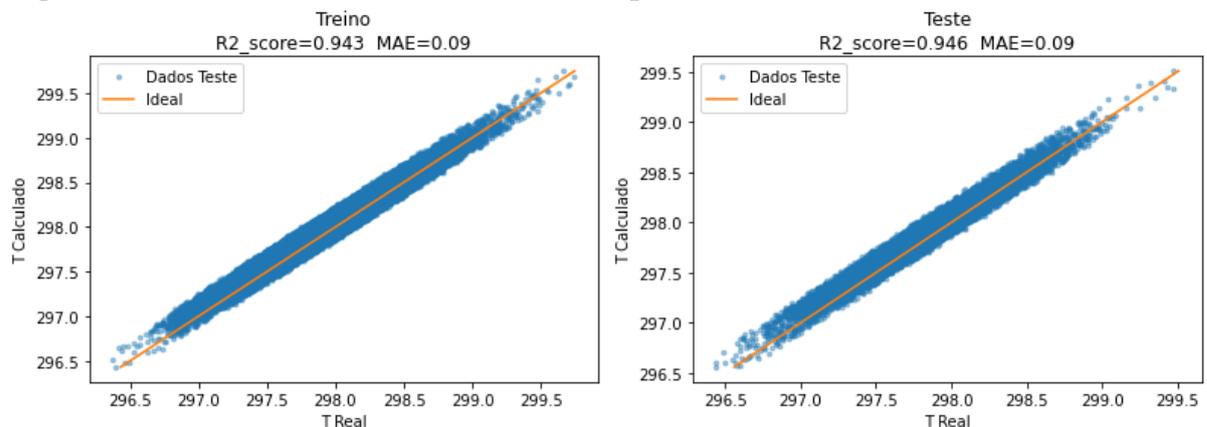
São apresentados a seguir os resultados obtidos com base na metodologia adotada no trabalho. A sequência dos resultados está coerente com as etapas de desenvolvimento e avaliação dos controladores.

### 5.1 Treinamento das redes

Na etapa de treinamento, foram utilizados 4 neurônios na camada oculta em todas as estruturas de aprendizado para evitar sobreajuste nos dados. Nas Figuras Figura 17 a Figura 20 são apresentados os dados reais versus preditos para as estruturas MLP, RNN, LSTM e GRU, tendo como variável de saída a temperatura do reator.

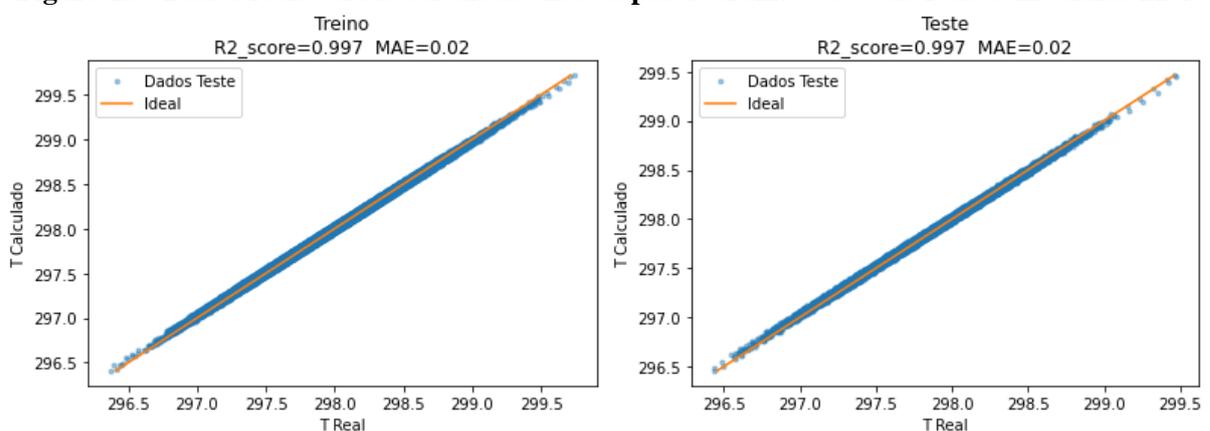
Mesmo com uma baixa quantidade de neurônios, pôde-se obter uma boa performance de predição para a rede MLP, mostrando que a estrutura tem boa captação de tendências oscilatórias do sistema - Figura 17.

**Figura 17 – Dados reais versus calculados nas etapas de treino e teste da rede com células MLP**



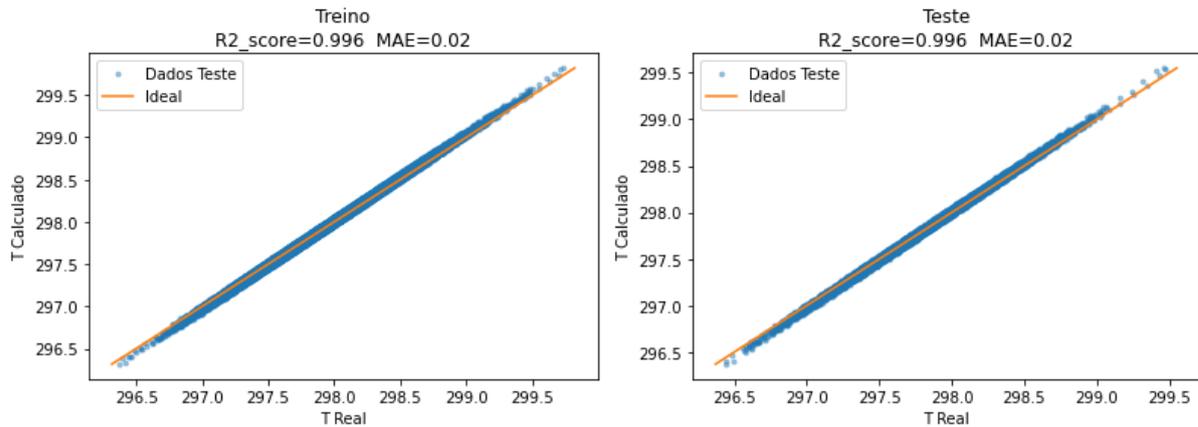
Fonte: O Autor (2020)

**Figura 18 - Dados reais versus calculados nas etapas de treino e teste da rede com células RNN**



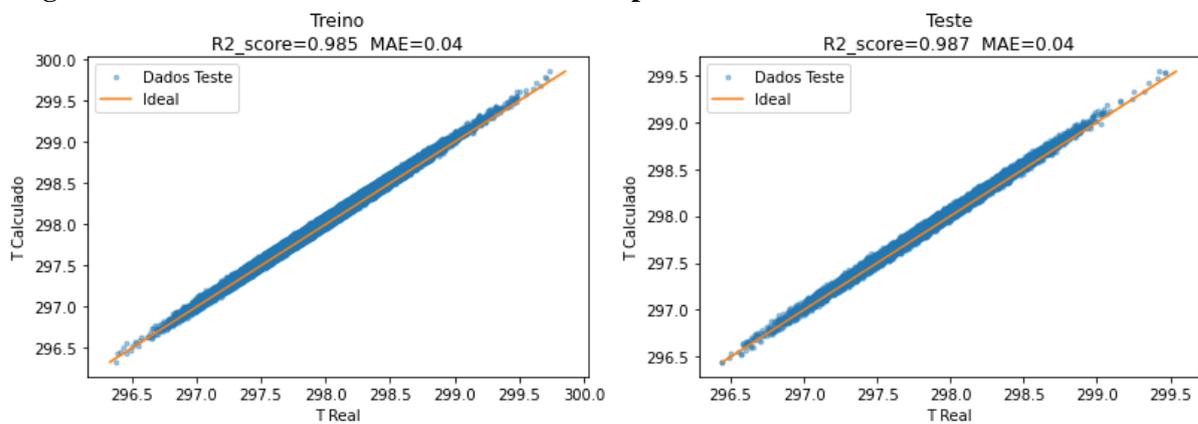
Fonte: O Autor (2020)

**Figura 19 - Dados reais versus calculados nas etapas de treino e teste da rede com células LSTM**



Fonte: O Autor (2020)

**Figura 20 - Dados reais versus calculados nas etapas de treino e teste da rede com células GRU**



Fonte: O Autor (2020)

Conforme esperado, a rede neural MLP apresentou menor performance em relação às redes recorrentes por não levar em consideração a tendência temporal dos dados na etapa de treinamento - Figura 17.

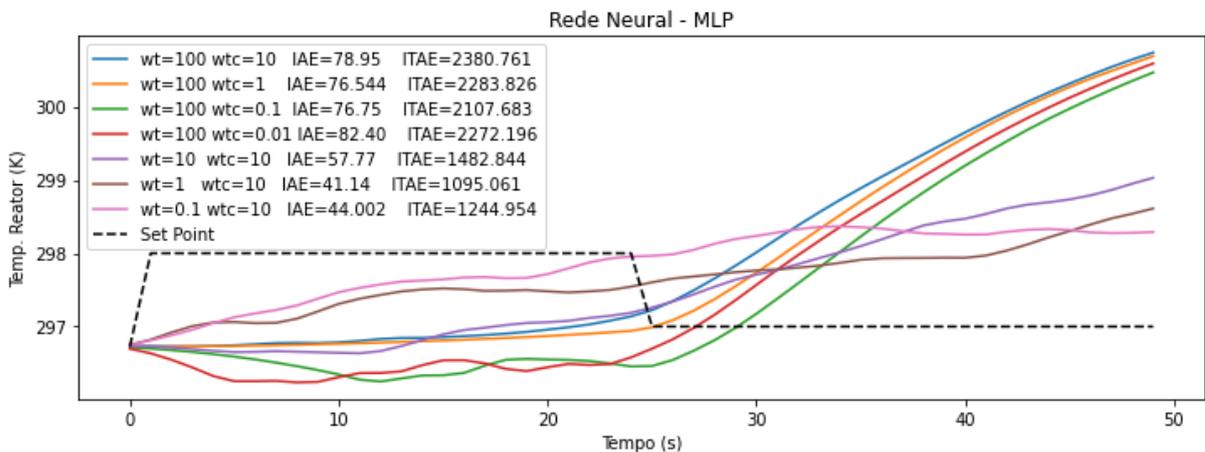
Em relação aos treinamentos das redes recorrentes – Figuras Figura 18, Figura 19 e Figura 20 –, era esperado que as redes mais robustas captassem a dinâmica do processo com maior acurácia. Porém, foi observado que a RNN teve maior assertividade na predição em relação às redes LSTM e GRU (redes de longa memória).

Apesar da ligeira diferença na capacidade preditiva das redes recorrentes (com base nas métricas de erro mencionadas anteriormente), é esperado que a rede MLP gere tendências com maior oscilatoriedade no sistema em relação às demais estruturas quando utilizada no controlador preditivo.

## 5.2 Implementação do controlador

A partir da variação dos pesos referentes às variáveis manipulada e controlada, foram obtidos os comportamentos dos controladores baseados em cada estrutura de aprendizado - Figuras Figura 21 a Figura 24. A partir das figuras, pode-se verificar como a performance dos controladores foi altamente influenciada pelas ponderações referentes à função objetivo. Porém, no caso do controlador com modelo de baixa acurácia (MLP - Figura 21), o aumento nos valores do peso para a variável controlada teve pouca influência na melhoria da performance do controlador, demonstrando que não há garantia de convergência rápida para o *set-point* quando utilizados maiores valores de ponderação (DUTRA, 2003).

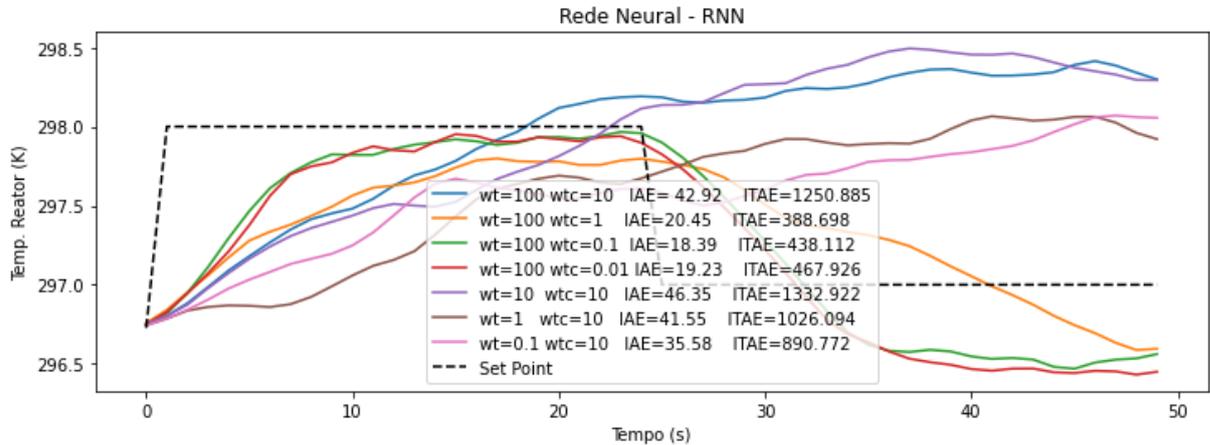
**Figura 21 – Performance do controlador baseado em MLP para diferentes pesos**



Fonte: O Autor (2020)

Observando o comportamento do controlador baseado em MLP - Figura 21, não foi possível a obtenção de um comportamento estável no sistema, apresentando tendências de aumento incontrolado da temperatura do reator. Tal comportamento pode ter sido fruto da baixa performance do controlador em relação aos baseados em redes recorrentes, considerando que a dependência temporal não é bem captada pela estrutura MLP.

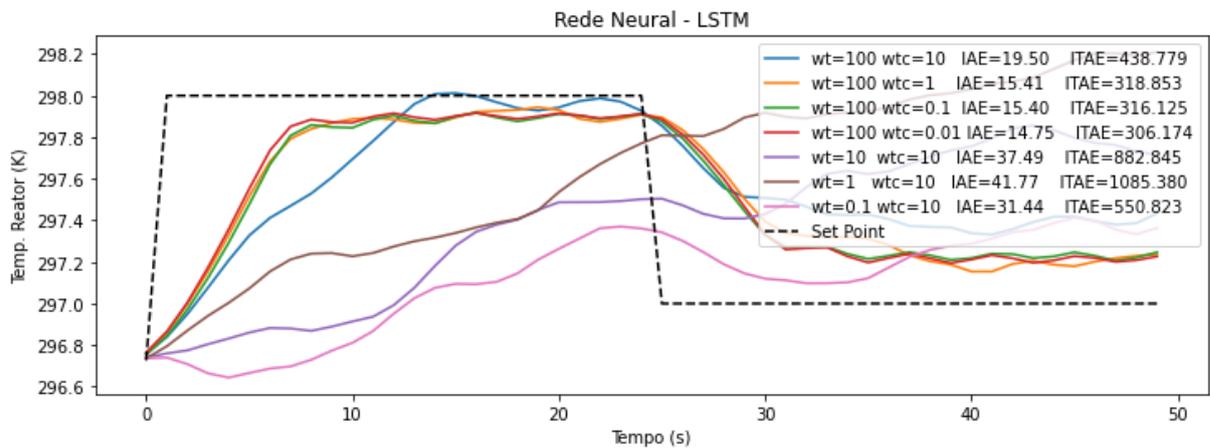
**Figura 22 - Performance do controlador baseado em RNN para diferentes pesos**



Fonte: O Autor (2020)

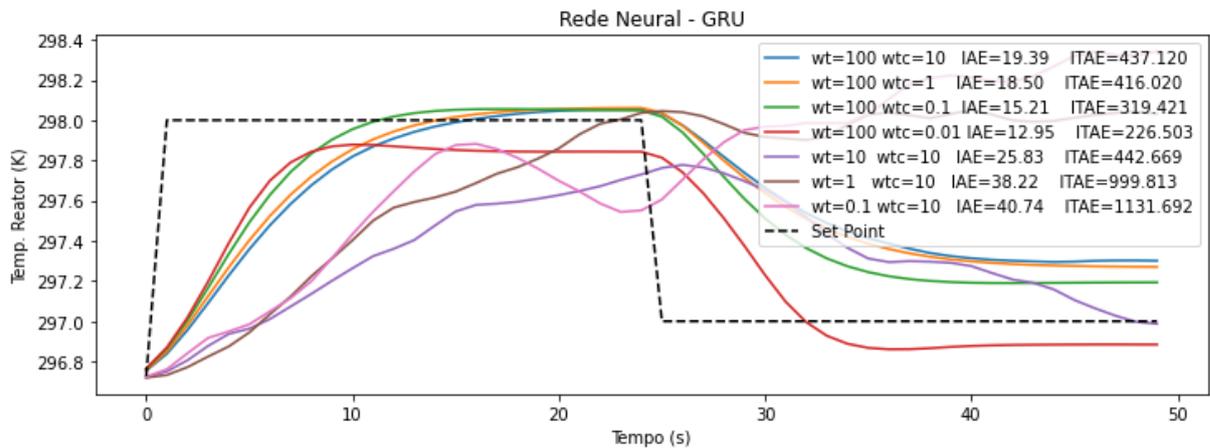
Para o controlador utilizando redes recorrentes RNN - Figura 22, pôde-se obter valores próximos aos de referência para alguns conjuntos de pesos. Quando o peso referente à variável manipulada  $w_{tc} = 10$ , comportamentos distantes do *set-point* foram obtidos (principalmente para baixos valores do peso  $w_t$ ).

**Figura 23 - Performance do controlador baseado em LSTM para diferentes pesos**



Fonte: O Autor (2020)

**Figura 24 - Performance do controlador baseado em GRU para diferentes pesos**



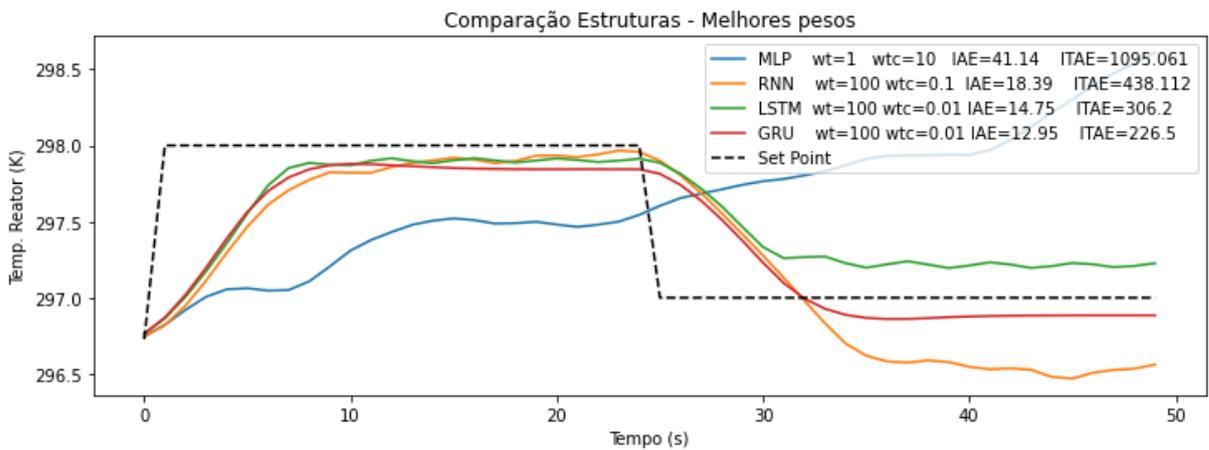
Fonte: O Autor (2020)

Os comportamentos dos sistemas com controladores baseados em redes recorrentes de longa memória (LSTM e GRU) – Figuras Figura 23 e Figura 24 – se mostraram com maior proximidade dos valores de referência (quando comparados às demais estruturas), principalmente quando utilizados maiores pesos para a variável controlada ( $w_t$ ). Tal comportamento demonstra a importância de se utilizar modelos de predição que levem em consideração a sequência dos dados, além da importância da sintonia (quanto aos valores dos pesos) para uma melhor performance do controlador.

### 5.3 Avaliações gerais dos controladores

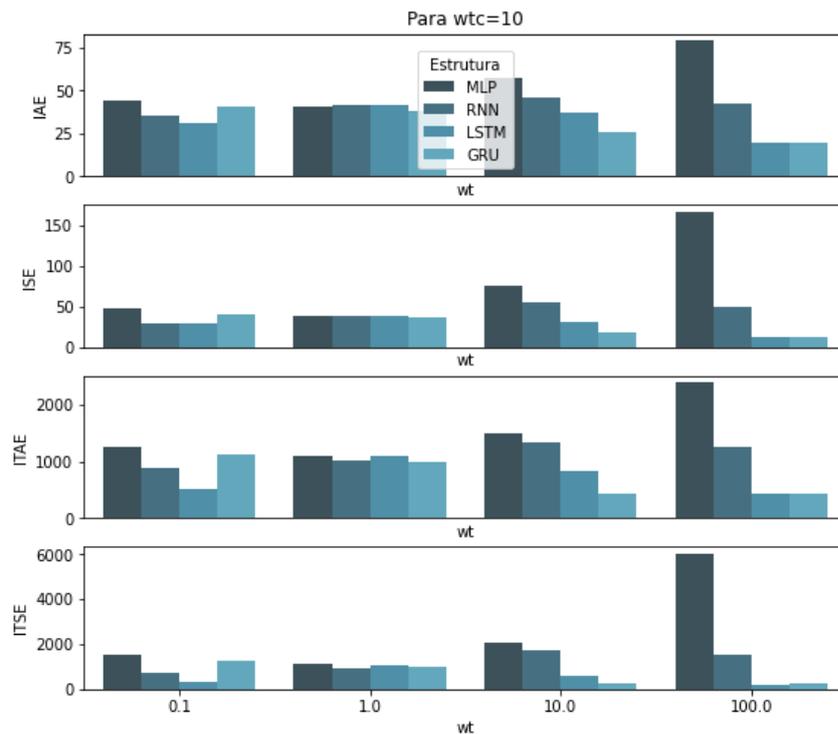
Através da seleção dos pesos que geraram sistemas de controle com maior performance (em relação a cada estrutura de rede), foi possível obter Figura 25. Nela é possível observar como as estruturas baseadas em *Deep Learning* (redes recorrentes) tendem a acompanhar os valores de referência com maior facilidade em relação ao controlador baseado em MLP (estrutura de *Machine Learning*).

**Figura 25 – Comparação das performances dos controladores utilizando os melhores pesos**



Fonte: O Autor (2020)

**Figura 26 – Visão geral da performance dos controladores com peso da variável manipulada constante**

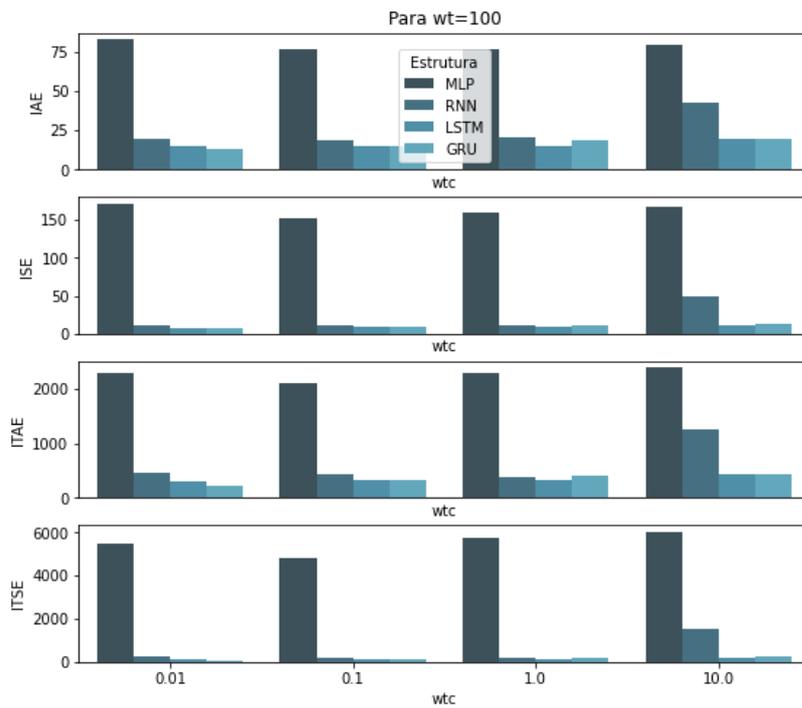


Fonte: O Autor (2020)

Através das Figuras Figura 26 e Figura 27 pode-se verificar de forma geral (através das métricas de erro) que independente dos parâmetros referentes às variáveis manipulada e controlada, os controladores baseados em estruturas *Deep Learning* performam de maneira melhor que a de *Machine Learning* (MLP).

A tendência temporal captada pelas estruturas e a capacidade de armazenamento de informações importantes fazem a diferença nas tomadas decisões realizadas pelo controlador baseado em estruturas de *Deep Learning*.

**Figura 27 - Visão geral da performance dos controladores com peso da variável controlada constante**



Fonte: O Autor (2020)

## 6 CONCLUSÕES

Foi apresentado de forma sucinta as etapas de desenvolvimento e implementação de estratégias de controladores MPC, avaliando sua performance a partir da utilização de diferentes arquiteturas de rede aplicadas a um sistema reativo não linear.

Através do estudo, foi possível a capacitação através da utilização da linguagem de programação *Python* e conteúdos ligados à área de controle de processos, como as Redes Neurais Artificiais (RNA), Redes Recorrentes (RNN), Redes Recorrentes de Longa Memória (LSTM e GRU) e de Otimização por Enxame de Partículas (PSO).

Nas delimitações do trabalho, a linguagem de programação *Python* se mostrou uma alternativa em relação a *softwares* comerciais utilizados para o controle de processos, sendo uma ferramenta gratuita e de alto nível para a solução de problemas de engenharia. A linguagem de programação se mostrou de fácil utilização, com uma gama de funções de otimização, algoritmos de *Machine Learning* e *Deep Learning* disponíveis em suas bibliotecas.

A partir das avaliações dos controladores, sistemas com maior estabilidade foram obtidos a partir das estruturas de *Deep Learning* (redes recorrentes). Em especial, as redes de longa memória apresentaram melhores performances por captar as singularidades em toda a cadeia de dados temporais. Após uma avaliação geral do sistema, condições de maior estabilidade foram obtidas para maiores valores de ponderação relacionados à variável controlada. Os controladores baseados em estruturas de *Machine Learning* (MLP) não apresentaram tal comportamento, com decaimento da performance causado pelo aumento da ponderação da variável manipulada, condizendo com a literatura no quesito da não garantia de melhores convergências para maiores ponderações.

Com o presente trabalho, foi possível contribuir com o crescimento do grupo de pesquisa do Laboratório de Sistemas Inteligentes Aplicados através do desenvolvimento de estratégias de controle que proporcionem sistemas com maior estabilidade em sistemas de alta complexidade. O estudo se mostrou de grande relevância para o desenvolvimento industrial voltado à Indústria 4.0, contribuindo assim para o desenvolvimento de processos interligados, com sistemas de controle mais estáveis e eficientes. Além disso, com contribuições para uma maior segurança de operações e o aumento na rentabilidade de processos.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

BENENATI, R. F. Process modeling, simulation and control for chemical engineers, William L. Luyben, McGraw-Hill, New York, 1973. 558 pp. **Journal of Polymer Science: Polymer Letters Edition**, v. 11, n. 4, p. 289–290, 1973.

CAI, L. et al. A noise-immune LSTM network for short-term traffic flow forecasting. **Chaos**, v. 30, n. 2, 2020.

CAMACHO, E. F.; BORDONS, C. **Model Predictive Control**, 1999.

CHEN, S.; WU, Z.; CHRISTOFIDES, P. D. Decentralized machine-learning-based predictive control of nonlinear processes. **Chemical Engineering Research and Design**, v. 162, p. 45–60, 2020.

DUTRA, C. B. S. **Controle Preditivo Multiobjetivo para Processos com Atraso**. p. 156, 2003.

E.LEBLANC STEVEN, D. R. C. Process systems analysis and control third edition. p. pp 391-401, 2009.

FOGLER, H. S. **Elements of Chemical Reaction Engineering**. Fifth Edit ed. [s.l: s.n.]. v. 44

GÉRON, A. **Hands-on Machine Learning with Scikit-Learn , Keras & TensorFlow**. [s.l: s.n.].

HERMIYANTY, WANDIRA AYU BERTIN, D. S. **Deep Learning Ian Goodfellow**. [s.l: s.n.]. v. 8

KITTISUPAKORN, P. et al. Neural network based model predictive control for a steel pickling process. **Journal of Process Control**, v. 19, n. 4, p. 579–590, 2009.

KMETOVA, J.; VASICKANINOVA, A.; DVORAN, J. Neuro-fuzzy control of exothermic chemical reactor. **Proceedings of the 2013 International Conference on Process Control**, PC 2013, p. 168–172, 2013.

OGUNNAIKE, BABATUNDE A.; RAY, W. H. **Process Dynamics, Modeling and Control**. [s.l: s.n.].

PON KUMAR, S. S. et al. A Deep Learning Architecture for Predictive Control. **IFAC-PapersOnLine**, v. 51, n. 18, p. 512–517, 2018.

RIBEIRO, C. **Controle Preditivo Multivariável Com Requisitos De Qualidade Em Plataformas De Produção De Petróleo**. 2012.

SARAIVA, S. V. **Estudo de um controlador preditivo baseado em sistemas inteligentes**. [s.l: s.n].

SCHWEDERSKY, B. B.; FLESCHE, R. C. C.; DANGUI, H. A. S. Practical nonlinear model predictive control algorithm for long short-term memory networks. **IFAC-PapersOnLine**, v. 52, n. 1, p. 468–473, 2019.

SEBORG, D. E. ET AL. **Process Dynamics and Control**. [s.l: s.n.]. v. 91

SILVA, G. D. E. C. **Estudo de um controlador preditivo não linear multivariável baseado em redes neurais**. [s.l: s.n.].

SILVA, G. D. E. C. **Estudo de um controlador preditivo não linear multivariável baseado em redes neurais**. [s.l: s.n.].

SILVA, G. D. **Machine learning no controle preditivo de processos**. [s.l: s.n.].

SILVA, R. **Contribuição à Otimização e Controle de Processos Fermentativos em Batelada Alimentada Fermentativos em Batelada Alimentada**. 2018.

SILVA, R. **Contribuição à Otimização e Controle de Processos Fermentativos em Batelada Alimentada Fermentativos em Batelada Alimentada**. 2018.

SOARES, A. P. DE M. R. **Métodos meta-heurísticos para otimização de processos químicos complexos em linguagem python**, 2019.

STEPHANOPOULOS, G. **Chemical process control: An introduction to theory and practice**. George Stephanopoulos, 1985.

TERZI, E.; FARINA, M.; SCATTOLINI, R. Model predictive control design for dynamical systems learned by Long Short-Term Memory Networks. 2019.

TOBERGTE, D. R.; CURTIS, S. **Python for Data Analysis**. [s.l: s.n.]. v. 53

UTKIN, V. A.; GULYUKINA, S. I. **Block approach for CSTR Control**. Proceedings - 2020 International Russian Automation Conference, RusAutoCon 2020, p. 261–266, 2020.