

UNIVERSIDADE ESTADUAL DE CAMPINAS Faculdade de Engenharia Química

HOMERO JACINTO SENA

CONTROLADOR MPC NEURAL COM ADAPTAÇÃO EM TEMPO REAL DO MODELO APLICADO AO CONTROLE DE pH E AO CONTROLE DE NÍVEL COM TANQUES COMUNICANTES

CAMPINAS

HOMERO JACINTO SENA

CONTROLADOR MPC NEURAL COM ADAPTAÇÃO EM TEMPO REAL DO MODELO APLICADO AO CONTROLE DE pH E AO CONTROLE DE NÍVEL COM TANQUES COMUNICANTES

Tese apresentada à Faculdade de Engenharia Química da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Química

Orientadora: PROFA. DRA. ANA MARIA FRATTINI FILETI

ESTE TRABALHO CORRESPONDE À VERSÃO FINAL DA TESE DEFENDIDA PELO ALUNO HOMERO JACINTO SENA, E ORIENTADO PELA PROFA. DRA. ANA MARIA FRATTINI FILETI

CAMPINAS

Ficha catalográfica Universidade Estadual de Campinas Biblioteca da Área de Engenharia e Arquitetura Luciana Pietrosanto Milla - CRB 8/8129

Sena, Homero Jacinto, 1988-

Se55c

Controlador mpc neural com adaptação em tempo real do modelo aplicado ao controle de ph e ao controle de nível com tanques comunicantes / Homero Jacinto Sena. – Campinas, SP: [s.n.], 2019.

Orientador: Ana Maria Frattini Fileti.

Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Química.

Controle de processos químicos - Automação.
 Controle preditivo.
 Redes neurais (Computação) - Modelos matemáticos.
 Inteligência artificial - Aplicações industriais.
 Fileti, Ana Maria Frattini, 1965-.
 Universidade Estadual de Campinas.
 Faculdade de Engenharia Química.
 Título.

Informações para Biblioteca Digital

Título em outro idioma: Neural MPC controller with real time model adaptation applied to pH control and level control of communicant tanks

Palavras-chave em inglês:

Control of chemical processes - Automation

Predictive control

Neural networks (Computing) - Mathematical models

Artificial intelligence - Industrial applications Área de concentração: Engenharia Química Titulação: Doutor em Engenharia Química

Banca examinadora:

Ana Maria Frattini Fileti [Orientador] Antonio José Gonçalves da Cruz Luz Adriana Alvarez Toro Matheus Souza

Ruy de Sousa Júnior

Data de defesa: 18-10-2019

Programa de Pós-Graduação: Engenharia Química

Identificação e informações acadêmicas do(a) aluno(a) - ORCID do autor: https://orcid.org/0000-0002-2734-1097

- Currículo Lattes do autor: http://lattes.cnpq.br/0220906938149410

Folha de Aprovação da Tese de Doutorado defendida por Homero Jacinto Sena e aprovada em 18 de outubro de 2019 pela banca examinadora constituída pelos seguintes doutores:

Profa. Dra Ana Maria Frattini Filet FEQ/UNICAMP

Prof. Dr. Antonio José Gonçalves da Cruz DEQ/UFSCAR

Profa. Dra. Luz Adriana Alvarez Toro FEQ/UNICAMP

Prof. Dr. Matheus Souza
FEEC/UNICAMP

Prof. Dr. Ruy de Sousa Júnior DEQ/UFSCAR

A ATA da Defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade

AGRADECIMENTOS

Agradeço à Profa. Ana Maria Frattini Fileti pela orientação, pelo apoio em todas as fases do desenvolvimento dessa tese e no crescimento alcançado durante esse período. Ao Prof. Flavio Vasconcelos da Silva pelas trocas de experiencias, a disponibilização de equipamentos sob sua responsabilidade e aos conselhos no desenvolvimento dessa tese e para a vida profissional. Agradeço ainda aos professores Roger Zemp, Thiago Vaz da Costa, Luz Adriana Alvarez Toro, Matheus Souza, Antônio José Gonçalves da Cruz, Ruy de Sousa Júnior e Paulo Roberto Alves Pereira que muito contribuíram com sugestões e correções. Estendo os agradecimentos ao Prof. Marco Aurélio Cremasco pelos ensinamentos ao longo do estágio docente e a toda comunidade da FEQ Unicamp, seria impossível citar todos aqui, mas sem essa equipe não seria possível o desenvolvimento de qualquer pesquisa.

Agradeço aos colegas Luís Felipe Safady, Rafael Sencio e Marcos Estevom que contribuíram diretamente com a realização dos experimentos facilitando muito processo. Aos amigos Carlos Adriano, Raphael Ribeiro, Tarcísio Soares, Victor Ramos e Filipe Coelho que além de contribuírem com o trabalho estiveram presentes nos eternos cafés do LESQ. Aos amigos Helberth Lopes, Matheus Pedroso, Ítalo Pedrosa, Breno Saldanha, Raissa Costa, Karina Klock e a muitos outros que mesmo não citados foram essenciais para que a caminhada até aqui tenha sido mais divertida.

Por fim, mas não menos importante, agradeço aos meus pais, Sebastião Sena e Lúcia Jacinto que, mesmo a distância, sempre me apoiaram e que são a base do que me tornei hoje. Agradeço à minha companheira, Flavia Secco, personagem muito importante dessa história, quem muito me apoiou, muitas vezes com paciência e compreensão e sempre com muito carinho, amor e companheirismo. Cito ainda minhas irmãs Aurea e Hortência e meu sobrinho Cauã que são igualmente importantes.

Estendo ainda os agradecimentos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) que, por meio do processo de nº 141922/2017-6, concedeu apoio financeiro para realização dessa pesquisa.

Sem todos vocês esse capítulo de minha vida não seria possível, muito obrigado.

RESUMO

O desempenho do controlador preditivo baseado em modelo (MPC) está diretamente ligado à capacidade do modelo em descrever o processo. Um dos desafios dessa classe de controladores é manter o bom funcionamento do processo quando há mudanças de condições operacionais ou distúrbios não modelados. Nesse sentido, esse trabalho propõe o uso de redes neurais artificiais (RNA) como modelo de predição do MPC utilizando um algoritmo de aprendizado em tempo real, baseado no algoritmo de filtro de Kalman estendido (FKE), para ajustar o modelo neural ao processo e manter o controle quando houver mudanças de condições operacionais e distúrbios não modelados. O MPC neural de modelo adaptativo desenvolvido foi avaliado em dois processos distintos não lineares: controle de nível em uma planta de tanques comunicantes e controle de pH em um tanque de mistura. Para ambos os processos foi realizada a modelagem fenomenológica, usada para desenvolver e testar a lógica de controle, e posteriormente foram realizados experimentos nos protótipos reais. Nestes experimentos, o controlador de modelo adaptativo apresentou redução de até 93% do somatório do erro quadrático, comparado à mesma lógica de controle sem adaptação do modelo. Além disso, a adaptação do modelo foi capaz de reduzir o esforço de controle, corrigir falhas da modelagem neural, eliminar desvios de setpoint e manter o controle do processo após mudanças de condições operacionais e aplicação de distúrbios. O MPC neural de modelo adaptativo se mostrou uma estratégia promissora para o controle de processos multivariáveis não lineares.

Palavras chave: Controle de processos químicos - Automação; Controle preditivo; Redes neurais (Computação) - Modelos matemáticos; Inteligência artificial - Aplicações industriais

ABSTRACT

The performance of model predictive controller (MPC) are related to the suitable fit between model response and actual process behavior. One of the challenges in MPC application is to keep the process control performance while operating conditions change or when unmodeled disturbances happen. In this context, the present work proposes the use of artificial neural network (ANN) as a prediction model into MPC strategy. While MPC is working, this ANN model is updated by an algorithm based on extended Kalman filter (FKE) logic. The updates are crucial to keep MPC working properly in case of changes in process operating conditions or when unmodeled disturbances happen. The developed adaptive neural MPC was applied to two nonlinear processes: level control of communicant tanks and pH control of a mixing tank. The first principles modeling of both processes were implemented for computational simulation in order to help the control development and previously testing of the control logic, before actual closed loop experiments. From actual prototype experiments, the adaptive neural MPC reduced the sum of square error in 93% compared to nonadaptive neural MPC. The adaptation of the model was able to reduce the control effort, correcting neural modeling mismatches, reducing setpoint deviations and keeping the MPC working properly in different process operating conditions and while unmodeled disturbance was induced. The adaptive neural MPC proved to be a promising control strategy for multivariable nonlinear processes.

Keywords: Chemical process control - Automation; Predictive control - Case studies; Neural Networks (Computer); Artificial intelligence

LISTA DE FIGURAS

Figura 2.1. Diagrama de blocos MPC.
Figura 2.2. Diagrama básico de um neurônio
Figura 2.3. Fluxograma dos algoritmos (a) do Filtro de Kalman e (b) do Filtro de Kalman
Estendido
Figura 2.4. Diagrama de blocos do controlador MPC Neural com adaptação do modelo em
tempo real
Figura 3.1.Fotos da montagem de bancada de tanques comunicantes construída pela
DidatiControl (a) tanques comunicantes; (b) placa de aquisição de dados; (c)
Diagrama de instrumentação (P&ID) do processo
Figura 3.2. (a) Foto da planta em escala piloto de controle de pH. (b) Diagrama de
instrumentação (P&ID) do processo.
Figura 3.3. Fluxograma do algoritmo de cálculo do pH usado para simular o peagâmetro.45
Figura 4.1. Experimento de identificação da planta de tanques comunicantes - simulação
fenomenológica
Figura 4.2. Comparação entre simulação computacional e experimento real da planta de
tanques comunicantes
Figura 4.3. Resultados do Treinamento da RNA. a) e b) Comparação da rede e dados de
treinamento do tanque V-101 e V-102, respectivamente; c) e d) Validação da RNA
para o nível do tanque V-101 e V-102, respectivamente
Figura 4.4. Teste servo do controlador MPC Neural sem adaptação do modelo, processo
simulado de tanques comunicantes
Figura 4.5. Módulo do erro de predição da rede neural (sem adaptação) em relação à
resposta do modelo fenomenológico do sistema de tanques de nível comunicantes. 60
Figura 4.6. Teste servo do controlador MPC Neural com adaptação do modelo, processo
simulado de tanques comunicantes
Figura 4.7. a) Módulo do erro de predição da rede neural (com adaptação) em relação à
resposta do modelo fenomenológico do sistema de tanques de nível comunicantes. b)
Norma da matriz de pesos e vieses de cada uma das camadas da rede neural. c)
Derivada (discreta) da norma da matriz de pesos e vieses da rede em relação ao
tempo

Figura 4.8. Teste servo do controlador MPC Neural sem adaptação do modelo, processo
real de tanques comunicantes
Figura 4.9. Módulo do erro de predição da rede neural (sem adaptação) em relação à
resposta da planta de tanques de nível comunicantes (processo real)
Figura 4.10. Teste servo do controlador MPC Neural com adaptação do modelo, processo
real de tanques comunicantes. 64
Figura 4.11. Módulo do erro de predição da rede neural (com adaptação) em relação à
resposta da planta de tanques de nível comunicantes (processo real)
Figura 4.12. Teste Regulatório do controlador MPC Neural com adaptação do modelo,
processo real de tanques comunicantes
Figura 4.13. Módulo do erro de predição da rede neural (com adaptação) em relação à
resposta da planta de tanques de nível comunicantes (processo real) para o teste
regulatório67
Figura 4.14. Experimento de identificação da planta controle de pH com perturbação
degrau na vazão de ácido - simulação fenomenológica
Figura 4.15. Experimento de identificação da planta controle de pH com perturbação
degrau na vazão de base - simulação fenomenológica
Figura 4.16. Comparação entre simulação computacional e experimento real da planta de
controle de pH
Figura 4.17. Experimento simulado staircase para obtenção de dados para treinamento da
RNA do processo de controle de pH
Figura 4.18. Resultado do treinamento da RNA. a) Comparação da resposta da rede e
dados de treinamento; b) Validação da RNA
Figura 4.19. Teste servo do controlador MPC Neural sem adaptação do modelo, processo
simulado de controle de pH
Figura 4.20. Módulo do erro de predição da rede neural (sem adaptação) em relação à
resposta do modelo fenomenológico do tanque de controle de pH
Figura 4.21. Teste servo do controlador MPC Neural com adaptação do modelo, processo
simulado de controle de pH
Figura 4.22. Módulo do erro de predição da rede neural (com adaptação) em relação à
resposta do modelo fenomenológico do tanque de controle de pH
Figura 4.23. Teste servo do controlador MPC Neural sem adaptação do modelo, processo
real de controle de pH76

Figura 4.24. Módulo do erro de predição da rede neural (sem adaptação) em relação à
resposta da planta de controle de pH (processo real)
Figura 4.25. Teste servo do controlador MPC Neural com adaptação do modelo, processo
real de controle de pH
Figura 4.26. Módulo do erro de predição da rede neural (com adaptação) em relação à
resposta da planta de controle de pH (processo real)
Figura 4.27. Comparação dos resultados do teste regulatório 1 (distúrbio na vazão da
solução tampão) do controlador MPC Neural com e sem adaptação do modelo
processo real de controle de pH
Figura 4.28. Comparação das variáveis manipuladas no teste regulatório 1 (distúrbio na
vazão de tampão). a) Bomba de ácido; b) Bomba de base
Figura 4.29. Comparação dos resultados do teste regulatório 2 (distúrbio na concentração
da solução tampão) do controlador MPC Neural com e sem adaptação do modelo
processo real de controle de pH
Figura 4.30. Módulo do erro de predição da rede neural em relação à resposta da planta de
controle de pH (processo real) no teste regulatório 2: a) Modelo sem adaptação; b)
Modelo adaptativo
Figura 4.31. Comparação dos resultados do teste regulatório 3 (mudança das concentrações
do ácido e base) do controlador MPC Neural com e sem adaptação do modelo,
processo real de controle de pH
Figura 4.32. Resultados apresentados por SAFADY (2018) do controlador baseado na
técnica de Extremum-seeking aplicado ao processo real de controle de pH
Figura 4.33. Sinal enviado às bombas de ácido e base do controlador MPC neural com
adaptação do modelo no experimento regulatório 3 (mudança das concentrações do
ácido e base)85
Figura 4.34. Módulo do erro de predição da rede neural em relação à resposta da planta de
controle de pH (processo real) no teste regulatório 3: a) Modelo sem adaptação; b)
Modelo adaptativo
Figura 4.35. Resultados do teste servo regulatório de longa duração usando o controlador
MPC Neural com adaptação do modelo, processo real de controle de pH 88
Figura 4.36. Módulo do erro de predição da rede neural (com adaptação) em relação à
resposta da planta de controle de pH (processo real) em teste servo regulatório de
longa duração89

LISTA DE TABELAS

Tabela 2.1. Funções de ativação
Tabela 3.1. Tabela com dados dos instrumentos da montagem de tanques comunicantes em
escala de bancada
Tabela 3.2. Detalhes dos principais instrumentos do processo de Controle de pH em escala
de laboratório
Tabela 3.3. Parâmetros do Controlador PID para o nível do tanque de mistura 44
Tabela 3.4. Variáveis de entrada e saída da rede neural do processo de tanques
comunicantes
Tabela 3.5. Variáveis de entrada e saída da rede neural do processo de controle de pH 48
Tabela 3.6. Parâmetros do método de otimização (fmincon)
Tabela 3.7. Lista de experimentos realizados
Tabela 4.1. Constantes do modelo fenomenológico de tanques comunicantes 55
Tabela 4.2. Valores das constantes de tempo do processo de tanques comunicantes para
cada uma das perturbações de identificação
Tabela 4.3. Experimentos para definição dos pesos de penalização da função objetivo do
processo de tanques comunicantes
Tabela 4.4. Coeficientes de Desempenho do Controlador com e sem adaptação do modelo
no processo real de níveis comunicantes
Tabela 4.5. Parâmetros do Modelo
Tabela 4.6. Valores das constantes de tempo do processo de controle de pH para cada uma
das perturbações de identificação
Tabela 4.7. Experimentos para definição dos pesos de penalização da função objetivo do
processo de controle de pH
Tabela 4.8. Índices de Desempenhos dos testes servo e regulatórios no processo real 86
Tabela 4.9. Condições do teste servo-regulatório de longa duração

LISTA DE SÍMBOLOS

Α	Matriz do modelo de espaço de estados	u.a.
В	Matriz do modelo de espaço de estados	u.a.
b ^(j)	Vieses do neurônio j	-
B_{RNA}	Matriz de Vieses (Bias) de uma camada da Rede Neural	u.a.
С	Matriz do modelo de espaço de estados	u.a.
C ₁	Concentração de HNO ₃	mol/l
C_2	Concentração de NaOH	mol/l
C3	Concentração de NaHCO ₃	mol/l
Carg	Saldo de cargas no balanço de cargas do modelo do peagâmetro	u.a.
err	Erro de predição do modelo	_
f	Função de ativação do neurônio	_
Н	Matriz Hessiana do método de Levenberg-Marquardt	
HC	Horizonte de controle	u.a.
HP	Horizonte de predição	u.a.
I	Matriz identidade	_
J	Função custo (otimização)	u.a.
Jac	Matriz Jacobiana do modelo Neural	_
k _{ac1}	Constante de dissociação do H ₂ CO ₃	u.a.
k _{ac2}	Constante de dissociação do HCO ₃ ¹⁻	u.a.
k _C	Ganho proporcional do PID	u.a.
k_{HV}	Constante de Válvula	dm ^{5/2} /
k _P	Constante do modelo da bomba	dm³/s
L _{max V}	Altura máxima da coluna de água (Nível)	dm
L_V	Altura da coluna de água (Nível)	dm
N_{ref}	Trajetória de referência de nível	u.a.
N_V	Nível percentual do Tanque	%
Р	Matriz de Covariância de Estados Atualizada no FKE	-
P _{erro SP}	Peso do desvio de Setpoint	u.a.
рН	Valor de pH	u.a.
pH_{ref}	Trajetória de referência de pH	u.a.
P_u	Peso de magnitude da atuação (custo da planta de pH)	u.a.
$P_{\Delta u}$	Peso da variação da variável manipulada	u.a.
q	Variância das variáveis observadas (FKE)	dm³/s
Q_{HV}	Vazão volumétrica na válvula	dm³/s
Q_P	Vazão volumétrica na bomba	dm³/s

R	Variância do modelo de observação (FKE)	-
R^2	Coeficiente de determinação	_
S	Sinal recebido no computador	Volts
SP	Setpoint definido para o processo	_
SSE	Somatório do Erro Quadrático	S
S_V	Área Transversal do Tanque	dm²
SVM	Somatório da Variação da Manipulação	S
t	Tempo	S
t_{amost}	Intervalo de amostragem	S
и	Variável manipulada adimensionalizada	u.a.
$U_{0,P}$	Constante do modelo de vazão da bomba	dm³/s
U_{FK}	Matriz calculada - passo do filtro de Kalman estendido	_
V	Matriz de ruído do modelo de espaço de estados	_
W	Matriz de Pesos de uma camada da Rede Neural	_
W	Peso sináptico do neurônio	
X	Variável independente qualquer	_
\widehat{X}	Vetor de Estados do modelo de espaços estados	_
$x^{(j)}$	Somatório dos sinais (interno ao neurônio)	_
Y	Vetor de Variáveis Estimadas do modelo de espaços estados	_
$y^{(j)}$	Resposta do Neurônio	-
y rede	Resposta da Rede Neural	-
y ref	Trajetória de referência para chegar ao setpoint	-
Z	Dados aquisitados do processo	_
[CO3 ²⁻]	Concentração de íons carbonato	mol/l
[H ⁺]	Concentração de íons hidrônio	mol/l
$[H_2CO_3]$	Concentração de carbonato de cálcio	mol/l
[HCO³-]	Concentração de íons bicarbonato	mol/l
[Na⁺]	Concentração de íons sódio	mol/l
$[NO_3^-]$	Concentração de íons nitrato	mol/l
[OH ⁻]	Concentração de íons hidroxila	mol/l
α	Constante para definir a trajetória de referência	u.a.
$lpha_{ac1}$	Fator de dissociação do HCO ₃ ¹⁻	u.a.
$lpha_{ac2}$	Fator de dissociação do íon CO ₃ ²⁻	u.a.
δ	Vetor gradiente do método de Levenberg-Marquard	
λ	Variável regularizadora do método de Levenberg-Marquard	
τ	Constante tempo do processo	S
$ au_D$	Constante derivativa PID	S

τ_{\prime}	Constante tempo Integral PID	5
φ	Função genérica	-
ω	Matriz de ruído do modelo de espaço de estados	-
	SUBSCRITO	
ent	Referente à entrada	
esc	Referente à camada escondida	
i	Índice de contagem	
k	Instante presente	
k+1	Instante futuro	
k-1	Instante passado	
sai	Referente à saída	
	SOBRESCRITO	
j	Índice da camada de neurônios da RNA	
t	Transposto	
	SIGLAS	
BPTT	Retro propagação ao longo do tempo (Back-Propagation Through Time)	
CLP	Computador lógico programável	
FKE	Filtro de Kalman estendido	
MIMO	Múltiplas entradas e múltiplas saídas	
MISO	Múltiplas entradas e única saída	
MPC	Controlador preditivo baseado em modelo	
PI	Controlador proporcional integral	
PID	Controlador proporcional integral derivativo	
PI&D	Diagrama de instrumentação	
RNA	Rede Neural Artificial	

Treinamento recorrente em tempo real (Real Time Recurrent Learning)

RTRL

SUMÁRIO

1	IN'	TRODUÇÃO	17
2	FU	NDAMENTAÇÃO TEÓRICA E REVISÃO BIBLIOGRÁFICA	20
	2.1	Controlador Preditivo Baseado em Modelo	20
	2.1	.1 Modelo de predição	21
	2.1	.2 Função Objetivo	22
	2.1	.3 Método de Otimização	23
	2.2	Modelo Neural	24
	2.3	Treinamento em tempo real	28
	2.4	Processos	31
	2.4	.1 Tanques comunicantes	31
	2.4	.2 Controle de pH	32
	2.5	Conclusão	33
3	MI	ETODOLOGIA	36
	3.1	Processos	36
	3.1	.1 Planta de Tanques Comunicantes	36
	3.1	.2 Planta de controle de pH	38
	3.2	Modelagem dos Processos	40
	3.2	.1 Modelagem fenomenológica	41
	3.2	.2 Modelagem Neural	46
	3.3	Controlador Preditivo Baseado em Modelo	49
	3.4	Algoritmo de adaptação do modelo	52
	3.5	Avaliação dos controladores	53
4	RE	SULTADOS	55
	4.1	Planta de Tanques Comunicantes	55
	4.1	.1 Modelagem e Identificação	55

	4.1.2 Testes em malha fechada		58
4	4.2 Pla	anta de Controle de pH	67
	4.2.1	Modelagem e Identificação	67
	4.2.2	Testes em Malha fechada	
5	CONC	CLUSÃO	90
6	SUGE	STÕES DE TRABALHOS FUTUROS	93
BI	BLIOGI	RAFIA	94

1 INTRODUÇÃO

Controle Preditivo Baseado em Modelo (MPC) é denominado como uma classe de controladores que utilizam explicitamente um modelo para calcular as ações de controle minimizando uma função objetivo (CAMACHO E ALBA, 2013). Esse tipo de controlador têm ganhado espaço no meio industrial e acadêmico em razão da capacidade de: ser aplicado a processos multivariáveis, inserir restrições do processo à lógica de controle e realizar otimização do processo em tempo real (WANG, 2009).

Segundo CAMACHO E ALBA (2013) uma das diferenças entre os vários algoritmos MPC está no tipo de modelo (usado para realizar previsões sobre os estados do processo). Modelos do tipo Redes Neurais Artificiais (RNA), são opção para aplicação a lógica de controle MPC.

As redes neurais artificiais, são estruturas matemática inspiradas no funcionamento de células nervosas de sistemas biológicos. Na RNA, estímulos de entrada são aplicados à estrutura e em cada unidade de processamento (denominado neurônios) esses sinais, ponderados por pesos, são somados e passam por uma função de ativação, produzindo uma resposta a ser enviada ao neurônio seguinte.

Essa estrutura passa por uma etapa de ajuste de parâmetros, denominada treinamento, com a finalidade de condicionar as respostas dos neurônios aos padrões dos estímulos de entrada. Dessa forma, a RNA pode ser utilizada como um modelo empírico de um processo, relacionando variáveis de entrada do modelo com a respostas do processo. Por ser um modelo empírico, a RNA necessita de dados do processo para realizar a etapa de treinamento. Esse banco de dados deve representar bem o domínio de operação do processo já que, como modelo empírico, não pode representar o comportamento fora da faixa apresentada durante a modelagem.

Autores como YU E ZHANG (2006) e NG *et al.* (2014) apresentam estudos usando RNA como modelo de predição em controladores MPC. YU E ZHANG (2006) aplica o controlador MPC Neural (NMPC) para controlar a conversão em um modelo de um reator CSTR manipulando a vazão de alimentação dos reagentes. Os autores propuseram um algoritmo de adaptação do modelo RNA usando o método de Levenberg-Marquardt em paralelo ao funcionamento do controlador, esse algoritmo armazena uma quantidade de dados e periodicamente atualiza a rede por um novo treinamento.

1. Introdução

NG et al. (2014) aplicam o NMPC em um sistema planta de bancada, sumulando um sistema de ar condicionado automotivo. Os autores também utilizam de um algoritmo de treinamento usando o método de Levenberg-Marquardt para treinar uma rede neural em paralelo ao funcionamento do controlado e o MPC alterna entre uma RNA de parâmetros fixos e a RNA de treinamento online para realizar os cálculos da ação de controle.

Inspirado nos resultados apresentados pelos autores, levantou-se a hipótese de recorrer a algoritmos envolvendo aprendizado de máquina para aprimorar o ajuste de um único modelo neural ao sistema real durante o funcionamento do controlador. Autores como IIGUNI *et al.* (1992), PUSKORIUS E FELDKAMP (1994) e CAO (2008) apresentam algoritmos baseados no filtro de Kalman Estendido (FKE) para ajustar, em tempo real, os parâmetros da rede neural.

O algoritmo de FKE é uma variação do estimador de estados linear, o filtro de Kalman, adaptado a modelos não lineares. O algoritmo busca estimar estados observáveis a partir de variáveis medidas do processo e da linearização do modelo que relaciona essas variáveis aos estados.

O objetivo dessa pesquisa foi a implementação de um controlador do tipo MPC com um algoritmo de adaptação do modelo de predição em tempo real, mostrando a possibilidade de desenvolvimento e ajustes do controlador em uma simulação do processo para posteriormente a aplicação no processo real.

Com essa abordagem, buscou-se agilizar a aplicação do controlador no processo real e mostrar que, na presença de incertezas de modelagem, ou mudanças de condições não previstas durante o desenvolvimento do controlador, o algoritmo de adaptação do modelo reduz o erro de predição e melhora o desempenho do controlador.

O controlador NMPC foi aplicada em dois processos não lineares de características distintas: planta de tanques comunicantes; e planta de Controle de pH. O primeiro processo é um sistema de múltiplas entradas e múltiplas saídas (*MIMO*) em que o nível de dois tanques, ligados por uma válvula, são controlados manipulando a vazão de alimentação de cada um dos tanques. O segundo processo é um sistema de múltiplas entradas e uma única saída (*MISO*) em que um tanque agitado é alimentado continuamente com uma solução tampão e o controle de pH é realizado a partir da dosagem de alimentação de uma solução ácida e outra básica.

1. Introdução

O desempenho do controlador com modelo adaptativo foi avaliado e comparado com o controlador de modelo fixo por meio do erro de *setpoint* e da variação da variável manipulada em cada um dos processos.

A contribuição desse trabalho é a união das técnicas já existentes de MPC neural e aprendizado usando FKE, aplicado a processos reais usuais da indústria química como controle de nível e controle de pH, buscando apresentar os ganhos em desempenho do controlador em decorrência da redução do erro de predição da RNA.

Buscou-se ainda apresentar uma metodologia de aplicação do MPC neural com adaptação do modelo em que todo o desenvolvimento do controlador e os ajustes dos parâmetros foram realizados em uma modelagem fenomenológica, evitando assim grandes intervenções no controlador quando ligado no processo real.

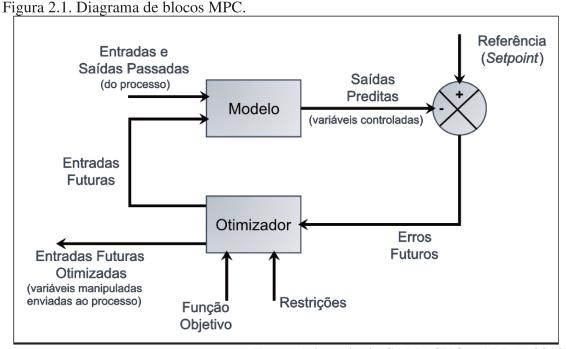
Nesta tese, é apresentado no Capítulo 2 o embasamento teórico sobre os principais temas que deram base ao desenvolvimento do controlador e ao final desse capítulo são apresentadas as conclusões sobre o estudo teórico que levaram ao desenvolvimento da pesquisa. No Capítulo 3 são apresentadas as metodologias usadas no desenvolvimento e testes do controlador, detalhando o algoritmo usado, as funções e os experimentos. O Capítulo 4 é dedicado aos resultados obtidos. Primeiramente são apresentados os resultados da simulação, tanto em malha aberta quanto em malha fechada, e posteriormente os resultados no processo real. Por último, no Capítulo 5 são apresentadas as conclusões do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO BIBLIOGRÁFICA

Como o objetivo principal desta pesquisa é a obtenção do controlador MPC neural com adaptação *online* do modelo, neste capítulo apresentamos a teoria dos principais itens que envolvem a pesquisa: controlador preditivo baseado em modelo; modelo neural; filtro de Kalman estendido e os processos de tanques comunicantes e de controle de pH.

2.1 Controlador Preditivo Baseado em Modelo

CAMACHO E ALBA (2013) classificam a estratégia de controle denominada Controle Preditivo Baseado em Modelo (MPC) como uma classe de controladores que utilizam, explicitamente, um modelo para calcular as ações de controle minimizando uma função objetivo. A ideia básica do funcionamento de um MPC é a de encontrar valores ótimos para as variáveis manipuladas (entradas do modelo) de forma a atingir (e/ou manter) o valor objetivo (*setpoint*) (WANG, 2009). Os cálculos dentro do controlador são realizados seguindo o fluxo apresentado no diagrama da Figura 2.1. Para realizar os cálculos do MPC, o algoritmo conta com 3 elementos essenciais: 1-Modelo de predição; 2- Função objetivo; 3- Método de otimização (CAMACHO E ALBA, 2013).



Fonte: Adaptado de CAMACHO E ALBA (2013)

2.1.1 Modelo de predição

O modelo de predição é responsável em fornecer ao controlador o comportamento do processo e por isso é de grande importância nesse tipo de controle. Dentro do controlador, o modelo prevê o comportamento do processo diante dos diferentes valores das variáveis manipuladas calculadas pelo algoritmo de otimização.

Devido à simplicidade de solução e de modelagem, modelos lineares são normalmente usados no projeto do MPC. Entretanto, muitos processos da indústria química possuem comportamento não linear, nesses processos modelos linearizados são aplicados em faixas estreitas de operação. Porém, quando há necessidade em representar o processo em diversas condições a linearização resulta em erros que inviabilizam o uso dessa classe de modelo (ŁAWRYŃCZUK, 2014). Nesses casos, modelos não lineares são necessários.

Em geral os modelos podem ser divididos em três grandes classes: 1- Modelo fenomenológico; 2- Modelo Empírico; 3- Modelo semi-empírico (WANG, 2009).

A modelagem fenomenológica parte de princípios de conservação e emprega equações constitutivas para gerar o equacionamento que o descreve. Esse tipo de modelo pode resultar em equações diferenciais (ordinárias ou parciais, dependendo do nível de detalhamento necessário) lineares ou não lineares. A solução dessas equações nem sempre pode ser realizada analiticamente (principalmente nos casos não lineares) e métodos numéricos de solução devem ser empregados, aumentando o custo computacional para a solução. Além disso, por partir de princípios físicos a modelagem depende de um profundo conhecimento técnico sobre o processo (LUYBEN, 1989).

Modelos empíricos não requerem detalhes sobre o processo, basta a reunião de dados experimentais que represente o processo. Esse tipo de modelagem é realizado por meio de ajuste de parâmetros de um modelo usando os dados disponíveis. Uma das limitações da modelagem empírica é sua faixa de validade uma vez que esses modelos só permitem interpolação dentro da faixa de ajuste do modelo, não permitindo extrapolar o comportamento do processo (LUYBEN, 1989). São exemplos de modelos empíricos: modelos polinomiais; e Redes Neurais Artificiais (RNA).

Já a modelagem semi-empírica é um híbrido dos dois tipos. O equacionamento é realizado com base no conhecimento fenomenológico do processo, porém alguns parâmetros do modelo são ajustados usando dados do processo (LUYBEN, 1989).

Sempre que possível é preferível usar a modelagem fenomenológica, que, geralmente, resulta em uma modelagem mais robusta do processo. Porém, principalmente devido a dificuldade em resolver o equacionamento resultante, modelos empíricos são mais usados para o fim de modelo de predição em MPC (WANG, 2009).

2.1.2 Função Objetivo

A Função objetivo é a equação que avalia as possíveis respostas para o problema e com base no julgamento dessa equação o algoritmo retorna uma solução a ser aplicada ao processo. O problema de otimização de um MPC pode ser formulado no sentido de minimizar uma função custo ou maximizar a produtividade ou lucro no processo (SEBORG *et al.*, 2010). Equações do tipo desvio quadrático podem ser usadas como função objetivo, podendo ser modificada com pesos que ponderem o desvio em diferentes variáveis consideradas pelo controlador.

Além dos pesos, outros dois parâmetros do MPC devem ser ajustados na função objetivo, o horizonte de predição (HP) e o horizonte de controle (HC). O horizonte de predição é o número de passos no futuro, considerados a cada etapa de cálculo do controlador. O horizonte de controle é o número de vezes que as variáveis manipuladas otimizadas serão calculadas a cada etapa de cálculo do controlador. Esses dois parâmetros estão ligados à agressividade com que o controlador atuará no processo, sendo que $HP \ge HC$ (CAMACHO E ALBA, 2013).

Se o horizonte de predição é pequeno, a função objetivo será calculada com base na resposta imediata do processo às ações de controle, resultando em ações corretivas mais imediatas. No caso de horizonte de predição grande, o custo considera os efeitos de longo prazo das ações de controle, suavizando as ações corretivas. Porém, quanto maior o horizonte de predição, maior o custo computacional na solução do problema de otimização.

Da mesma forma, o horizonte de controle pequeno faz com que a ação de controle seja imediata e horizonte de controle grande distribui a ação de controle ao longo do horizonte de predição. Apesar da possibilidade de cálculo de uma sequência de ações de controle (*HC>1*), somente a primeira atuação é aplicada ao processo, isso porque as demais ações de controle estão mais suscetíveis a incertezas do modelo de predição (SEBORG *et al.*, 2010).

2.1.3 Método de Otimização

O algoritmo usado para resolver a otimização de um MPC depende, principalmente, de dois fatores: 1- A existência de restrições no universo de solução ou respostas e 2- O tipo de modelo de predição escolhido.

A adição de restrições à lógica de controle é uma característica marcante do algoritmo do MPC (SEBORG *et al.*, 2010). Os processos reais são limitados, seja por limites físicos (fluxos de alimentação positivos, rendimento máximo, porcentagens) ou por limites de equipamentos (abertura máxima de válvulas, capacidade de bombeamento, pressão limite de segurança), e esses limites não podem ser ignorados pelo controlador. A inserção de restrições ao controlador cria a necessidade da utilização de métodos de otimização que respeitem esses limites adicionados.

Dois tipos de restrições podem ser considerados na formulação do MPC: as restrições flexíveis (soft); as restrições rígidas (hard). Restrições flexíveis são consideradas pela penalização diferenciada na função objetivo fazendo com que o algoritmo evite tais condições. Já as restrições rígidas são adicionadas ao método de otimização. A principal diferença entre os dois tipos de restrições é que as restrições flexíveis podem não ser respeitadas em determinadas ocasiões, porém as restrições rígidas sempre serão respeitadas. A inserção de restrições rígidas requer tipos especiais de métodos de otimização que considerem os limites no universo de soluções (SEBORG et al., 2010).

Modelos lineares, muito usados em MPC, podem ser otimizados usando métodos algébricos de otimização. Métodos baseados em programação quadrática são amplamente discutidos na literatura e suficientemente rápidos de serem resolvidos, sendo facilmente aplicáveis em um processo funcionando em tempo real (CAMACHO E ALBA, 2013).

Quando o processo apresenta não linearidades, que não podem ser eliminadas sem comprometer a qualidade de predição do modelo, são necessários métodos especiais de otimização capazes de lidar com as não linearidades. Muitos dos algoritmos são computacionalmente custosos ou não garantem que a solução encontrada é, de fato, a melhor solução para o problema (convergência a um mínimo local). Alguns *softwares* matemáticos trazem métodos de solução de problemas não lineares pré-programados; uma das classes de métodos mais comuns é o de pontos interiores (*interior-point*).

Os métodos de pontos interiores surgiram como alternativa mais rápida (comparada aos métodos *simplex*) de solução de problemas de programação linear. Apesar de convergir em menor número de iterações que os métodos *simplex*, cada iteração dos métodos de pontos interiores envolve operações computacionalmente mais custosas, sendo preferível em problemas de muitas dimensões (FORSGREN *et al.*, 2002). Outra vantagem dos métodos de pontos interiores é a possibilidade de aplicação em problemas de programação quadrática e não lineares (FORSGREN *et al.*, 2002).

Nos métodos de pontos interiores a busca pelo ponto ótimo é realizada a partir de pontos internos (ou externos, dependendo do método específico) a região factível (descrita pelas restrições do problema), nunca pelos limites dessa região (FORSGREN *et al.*, 2002). Sendo essa a característica que dá nome a classe de métodos e a principal diferença em relação aos métodos simplex (cuja busca é realizada nos limites da região factível).

Diversos métodos encontrados na literatura podem ser classificados como de pontos interiores, dentre eles os métodos primal-dual; afim escada; preditor corretor; e o método da barreira logarítmica (disponível no pacote de otimização do *software Matlab*[®]). O método de pontos interiores de barreira logarítmica é indicado para problemas de otimização não linear com restrição (NOCEDAL E WRIGHT, 2006).

2.2 Modelo Neural

Redes Neurais Artificiais são, segundo (HAYKIN, 2001), estruturas que imitam o funcionamento de células nervosas conferindo respostas a certos estímulos. Essas estruturas são formadas por unidades menores, denominadas neurônios. Os neurônios são organizados em camadas e interagem uns com os outros imitando o funcionamento do sistema nervoso biológico.

Cada neurônio (Figura 2.2) é composto por uma matriz de pesos (fatores que ponderam os sinais de entrada), um viés ou *bia* (uma constante que é somada aos sinais de entrada) e a função de ativação (função matemática responsável em codificar o sinal processado no neurônio antes de transferir ao próximo neurônio).

Matematicamente, a operação realizada dentro de um neurônio da camada 'j' é dada pela operação vetorial da Equação 2.1.

$$y^{(j)} = f\left(W^{(j)}.y^{(j-1)} + B_{RNA}^{(j)}\right)$$
 2.1

Apesar de ser composta por operações relativamente simples, uma rede neural é capaz de descrever uma infinidade de sistemas, principalmente não lineares, cuja modelagem fenomenológica é complexa e custosa para resolver.

A modelagem neural é realizada através do ajuste dos pesos e dos vieses que compõem cada um dos neurônios de forma a obter a resposta desejada a partir dos estímulos de entrada. Além do ajuste dos parâmetros é necessário definir também: 1- A arquitetura da rede (como os neurônios são organizados e qual o fluxo de sinal dentro da rede); 2- Número de neurônios e; 3- O tipo de função de ativação.

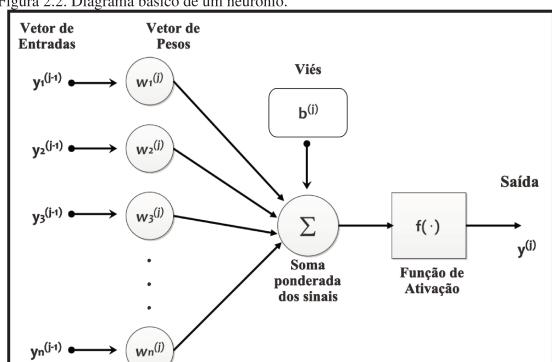


Figura 2.2. Diagrama básico de um neurônio.

Fonte: Adaptado de HAYKIN (2001)

São três os principais tipos de arquitetura de uma RNA: feedforward de camada única; feedforward de múltiplas camadas e redes recorrentes (HAYKIN, 2001).

Redes feedforward de camada única é a arquitetura mais simples, em que uma única camada de neurônios é responsável pelo processamento dos sinais de entrada e a saída já é obtida diretamente dessa camada de neurônios.

Redes feedforward de múltiplas camadas diferenciam-se do primeiro tipo unicamente pela existência de mais camadas de neurônios (camadas escondidas). A existência de múltiplas camadas adiciona complexidade à estrutura e resulta em uma maior conectividade entre as respostas dos neurônios. No caso de redes com arquitetura recursiva, sinais de saída da rede são retroalimentados (com atraso) junto com as entradas.

A quantidade de neurônios da rede usualmente é definida empiricamente levando em consideração que: redes muito pequenas têm maior dificuldade de representar sistemas complexos; e redes muito grandes tem grande número de parâmetros a serem ajustados (maior dificuldade de treinamento).

Além disso, é necessário tomar cuidado com o número de neurônios por conta do sobreajuste do modelo (*overfitting*). O grande número de parâmetros do modelo (pesos e vieses) pode resultar na interpolação dos dados de treinamento resultando em um modelo sem a capacidade de generalização (HAYKIN, 2001).

Diferentes funções de ativação podem ser usadas para transmissão de dados entre neurônios de uma rede, sendo que a Tabela 2.1 apresenta as principais. A função de ativação tem como principal função normalizar o sinal que corre dentro da rede e é ela que confere a característica não linear da estrutura. Não há uma regra geral para a escolha da melhor função de ativação para uma rede, sendo necessários testes empíricos para definir a que melhor atende. Entretanto, usualmente a função de ativação linear é usada na camada de saída da rede por permitir que as variáveis de saída da rede possam assumir quaisquer valores.

Tabela 2.1. Funções de ativação

Tabela 2.1. Funções de ativação			
Função	Equação	Gráfico	
Linear	$y^{(j)} = x^{(j)}$	0 +1 x	
Sigmoide	$y^{(j)} = \frac{1}{1 + e^{-(x^{(j)})}}$	+1 y	
Tangente hiperbólica	$y^{(j)} = \frac{e^{(x^{(j)})} - e^{-(x^{(j)})}}{e^{(x^{(j)})} + e^{-(x^{(j)})}}$	y 0 x	

Fonte: Adaptado de HAYKIN (2001)

O processo de treinamento de uma RNA pode ser do tipo supervisionado (em que existem informações sobre a resposta esperada) e não supervisionado (em que não existe resposta esperada disponível para treinamento). Em ambos os casos, os pesos e vieses da rede são ajustados de forma a excitar de forma diferente cada neurônio dependendo do conjunto de sinais de entrada. As respostas da rede são então dadas conforme o grau de excitação gerado na camada de saída.

No treinamento supervisionado, usualmente são usados algoritmos de otimização multivariável que buscam resolver o problema de mínimos quadrados do erro entre os dados de treinamento e as respostas da rede. No entanto, esta pesquisa se limitou a usar o método de *Levenberg-Marquardt*, que é o algoritmo padrão do software usado (*Matlab*®).

O método de Levenberg-Marquardt é, segundo HAYKIN (2009), uma mescla entre os métodos de Newton (que converge rapidamente quando próximo do mínimo, porém não possui garantia de convergência) e o método de máxima descida (possui convergência garantida, porém ela é lenta). No método de Levenberg-Marquardt os pesos da RNA organizados em um vetor (Ω) são atualizados a cada iteração de treinamento pelo cálculo de incrementos ($\Delta\Omega$) obtido pela Equação 2.2 (HAYKIN, 2009):

$$\Delta\Omega = (H + \lambda . I)^{-1} \cdot \delta \tag{2.2}$$

em que H é a matriz hessiana da função RNA em relação aos pesos, δ é o vetor gradiente da função RNA em relação aos pesos, I a matriz identidade (de mesma ordem da matriz hessiana) e λ é uma variável regularizadora a fim de fazer com que o termo $(H + \lambda.I)$ seja positivamente definido. Além disso, o regularizador faz com que o método seja mais próximo do método de Newton (se λ for muito pequeno) ou do método de máxima descida (se λ for elevado).

O método é iniciado com valores pequenos de λ (recomenda-se da ordem de 10^{-3}), a cada iteração o algoritmo minimiza uma função objetivo e a medida em que essa função se aproxima de um mínimo o regularizador sofre incrementos (HAYKIN, 2009). Dessa forma nas primeiras iterações o algoritmo converge rapidamente à região do mínimo e a medida em que o regularizador aumenta, a solução é refinada de forma mais lenta (HAYKIN, 2009).

Denominado um algoritmo de otimização de segunda ordem (requer o cálculo da matriz Hessiana, operação computacionalmente custosa), o método de *Levenberg*-

Marquardt, normalmente, converge a uma solução em poucas iterações (HAYKIN, 2009). Para reduzir o esforço computacional, alguns algoritmos simplificam o cálculo da matriz Hessiana nas primeiras iterações até ser detectada a aproximação do mínimo de forma a minimizar o esforço computacional (HAYKIN, 2009).

2.3 Treinamento em tempo real

HAYKIN (2001) apresenta dois modos de treinamento de uma RNA: o treinamento em batelada; e o treinamento sequencial. No treinamento em batelada, existe um banco de dados que é usado para treinar a rede, resultando em um modelo que represente (quando o treinamento é bem-sucedido) o comportamento dos dados apresentados. Já no treinamento sequencial, os dados são apresentados aos poucos e a rede é ajustada a cada conjunto de dados recebidos.

Algoritmos que realizam o treinamento sequencial da RNA possibilitam o treinamento em tempo real. Nesse tipo de treinamento a rede fornece sua resposta aos estímulos de entrada, ao mesmo tempo essa resposta é comparada aos dados do processo e o algoritmo de aprendizado recalcula os pesos e vieses dos neurônios de forma a aprimorar a rede para estímulos futuros.

O treinamento em tempo real é indicado, principalmente, para processos não estacionários em que um modelo fixo (resultado do treinamento em batelada) perderá eficiência a medida em que o processo muda de comportamento (HAYKIN, 2001).

HAYKIN (2001) cita como exemplos de algoritmos de treinamento em tempo real: retro propagação ao longo do tempo (BPTT); treinamento recorrente em tempo real (RTRL) e algoritmos baseados em filtro de Kalman. Dentre esses, o alvo dessa pesquisa é a utilização do algoritmo de filtro de Kalman estendido (FKE).

Segundo HAYKIN (2001), o treinamento em tempo real quando realizado com algoritmos baseados em métodos de máxima descida (como o BPTT e o RTRL) são geralmente lentos em decorrência da necessidade de estimar gradientes. Essa limitação pode ser contornada tratando o treinamento como um problema de filtragem ótima. Autores como IIGUNI *et al.* (1992), PUSKORIUS E FELDKAMP (1994) e CAO (2008) apresentam em seus estudos algoritmos de treinamento de RNA usando o filtro de Kalman e suas variações.

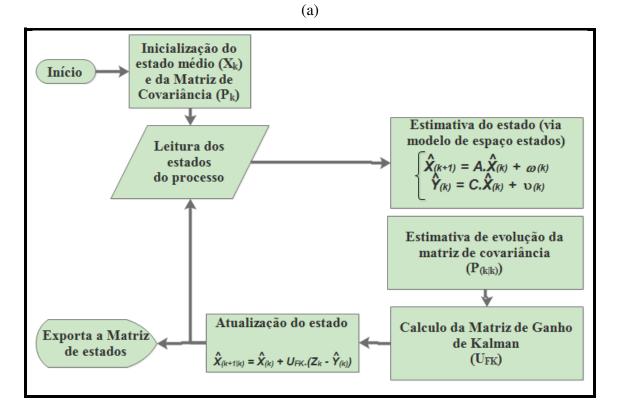
O algoritmo de filtro de Kalman foi desenvolvido na década de 1960 e trata-se de um estimador de estados ótimo, usado principalmente para fazer correções (baseadas em modelo linear) ou estimar (através de uma solução recursiva) propriedades que não podem ser medidas diretamente (HAYKIN, 2004).

O filtro de Kalman considera um modelo do tipo espaço de estados da variável estimada (Y) em relação aos estados do processo (\hat{X}) (Equações 2.3 e 2.4), considerando ainda a existência de ruídos brancos (independentes e normalmente distribuídos) na estimativa do estado $(\omega_{(k)})$ e na variável estimada $(v_{(k)})$.

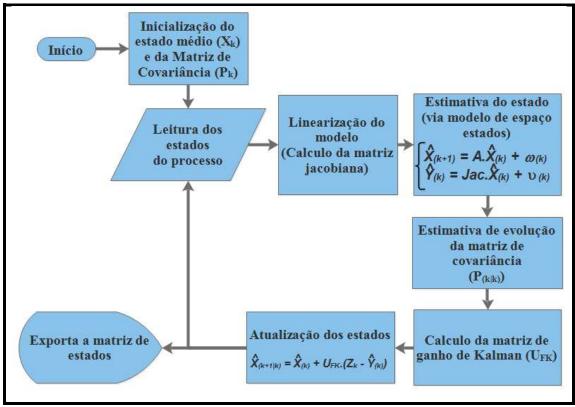
$$\begin{cases}
\hat{X}_{(k+1)} = A. \hat{X}_{(k)} + \omega_{(k)} \\
\hat{Y}_{(k)} = C. \hat{X}_{(k)} + v_{(k)}
\end{cases}$$
2.3
2.4

O filtro faz predições, a partir do modelo, dos estados médios futuros $(\hat{X}_{(k+1)})$ que podem (ou não) ser medidos diretamente no processo. Essa predição de estados é usada na próxima amostragem para estimar as variáveis mensuráveis do processo (a partir do modelo) e os resultados são comparados com os dados medidos em tempo real. A partir da comparação os parâmetros do filtro são atualizados a fim de obter uma melhor predição do estado futuro para as próximas amostragens. A Figura 2.3 apresenta o diagrama de blocos do algoritmo do filtro de Kalman e do filtro de Kalman estendido.

Figura 2.3. Fluxograma dos algoritmos (a) do filtro de Kalman e (b) do filtro de Kalman estendido.



(b)



Fonte: Adaptado de HAYKIN (2004)

Quando o modelo que relaciona os estados do processo com as variáveis mensuráveis não é linear, é realizada a linearização do modelo dando origem ao filtro de Kalman estendido (HAYKIN, 2004).

O treinamento de uma RNA usando algoritmo de filtro de Kalman estendido busca estimar os pesos e os vieses dos neurônios (conjunto de estados a serem estimados) minimizando o desvio quadrático entre a resposta da RNA e os dados recebidos do processo. A RNA, por ser um modelo não linear é então linearizada calculando a matriz de derivadas parciais, matriz Jacobiana (Jac), em relação aos pesos da rede. A representação do modelo linearizado, usado para estimar os pesos da rede, é apresentada pelas Equações 2.5 e 2.6. Observa-se que o modelo de atualização dos estados (\hat{X}) prevê que o estado futuro seja igual ao estado atual.

$$\begin{cases} \hat{X}_{(k+1)} = \hat{X}_{(k)} + \omega_{(k)} \\ Y_{(k)} = Jac \cdot \hat{X}_{(k)} + v_{(k)} \end{cases}$$
 2.5

Como há a necessidade de cálculo da matriz Jacobiana, o algoritmo de treinamento é computacionalmente custoso, principalmente quando a rede é grande (muitos parâmetros a serem ajustados). Por outro lado, a cada instante de amostragem

os parâmetros da rede serão atualizados em uma única iteração, diferente dos algoritmos de otimização que realizam várias iterações (HAYKIN, 2004).

Para reduzir o custo computacional do algoritmo de adaptação para redes grandes, PUSKORIUS E FELDKAMP (1994) sugerem considerar somente os efeitos da variação dos pesos na resposta dos neurônios a que eles afetam, reduzindo a matiz Jacobiana a uma matriz diagonal. Esse artifício recebe o nome de filtro de Kalman estendido desacoplado.

2.4 Processos

A avaliação do controlador foi realizada em dois processos distintos: planta de tanques comunicantes; e planta de controle de pH. Esses processos foram escolhidos em razão das suas características não lineares que justificam o uso de técnicas de controle avançado e por envolverem múltiplas variáveis. Além disso, destaca-se o processo de controle de pH que é um processo muito comum na indústria.

2.4.1 Tanques comunicantes

Tanques de nível são equipamentos muito comuns em indústrias químicas, seja na forma de tanques de estabilização, pontos de mistura, ou mesmo em reatores. Nesses processos o controle de nível sempre se repete com pouca variabilidade.

Em processos que não requerem o controle em um nível específico, são usados sensores que indicam nível baixo e nível alto atuando de forma discreta somente quando os extremos são atingidos. Já processos em que o controle deve ser mais rigoroso, controladores do tipo Proporcional Integral Derivativo – PID (lineares) são perfeitamente capazes de manter o *setpoint*.

Entretanto, o controle de nível pode apresentar características não lineares, dificultando a aplicação de técnicas clássicas de controle linear, principalmente quando o sistema envolve mais de um tanque a ser controlado.

O sistema de tanques comunicantes, apesar de não ser um arranjo comum da indústria química, é usado em diversos estudos de caso para teste e desenvolvimento de novas técnicas de controle. O sistema, por ser composto por mais de um tanque, se torna multivariável e, por envolver válvulas fazendo a ligação entre os tanques, tem comportamento não linear.

LIAN et al. (1998), RAMLI et al. (2009) e KHALID E KADRI (2012), fazem uso de um sistema de dois tanques que se comunicam por meio de uma válvula no fundo

dos tanques. Os tanques são alimentados por bombas (uma para cada um dos tanques) e são continuamente drenados por aberturas também reguladas por válvulas no fundo dos tanques. Em todas as pesquisas o controlador é responsável por manter o nível dos dois tanques simultaneamente em um valor desejável atuando nas bombas de alimentação (duas entradas e duas saídas).

LIAN *et al.* (1998) apresentam a sintonia de um controlador neuro-fuzzy usando algoritmo genético. RAMLI *et al.* (2009) buscam o controle usando um controlador PI-Neural de sintonia adaptativa. KHALID E KADRI (2012) fazem o controle usando controlador preditivo baseado em modelo linear.

Os três trabalhos destacam o comportamento não linear do processo, apresentam a modelagem semi-empírica, realizada pelo balanço de massa dos tanques e pela identificação dos modelos das bombas e válvulas. Em todos os trabalhos os controladores foram aplicados no processo real e conseguiram controlar o processo em testes variando o *setpoint*.

2.4.2 Controle de pH

O pH é uma medida da concentração de íons H⁺ ([H⁺]) em um sistema e é definido pela Equação 2.7.

$$pH = -log_{10}[H^+] 2.7$$

Em água pura a concentração de H⁺ é dada pelo equilíbrio apresentado na Equação 2.8. Na água pura as concentrações dos íons H⁺ e OH⁻ são iguais e o meio é classificado como neutro (pH = 7).

$$H_{(aq)}^+ + OH_{(aq)}^- \rightleftharpoons H_2O_{(l)}$$
 2.8

O desequilíbrio desses íons determina se o sistema está ácido ou básico. Esse desequilíbrio ocorre com a adição de outras espécies ao meio. Em geral, substâncias que provocam o aumento da concentração de H⁺ (diminui o pH) são classificadas como ácidos e substâncias que provocam a diminuição da concentração de H⁺ (aumenta o pH) são classificadas como bases. Porém, esse efeito de aumentar ou diminuir a concentração de H⁺ é regido por leis termodinâmicas de forma que cada substância terá um comportamento distinto quando adicionada a um sistema.

Com isso, o pH apresenta comportamento não linear, alta sensibilidade e varia bastante o comportamento dependendo da composição do meio. Por isso é considerado uma das mais complexas variáveis de processo para ser controlada (HERMANSSON E SYAFIIE, 2015).

Ainda assim, o controle de pH está presente em uma variedade de processos da indústria química. Apesar da neutralização de efluentes ser o caso mais comumente estudado, outros processos necessitam operar dentro de uma faixa ótima de pH (fermentação, coagulação, precipitação de metais, crescimento de microrganismos, etc.) (HERMANSSON E SYAFIIE, 2015). Na maioria dos casos o controle de pH é realizado em meio aquoso em que uma solução básica e/ou ácida é adicionada para promover alteração do meio.

Alguns íons (como carbonato, fosfato, etc.) provenientes de ácidos fracos, ou que possuam múltiplos hidrogênios ácidos, dissociam-se parcialmente, agindo como agentes tamponantes na solução. Esse efeito provoca um aprisionamento parcial de hidrogênios em algumas faixas de pH, impedindo que o pH varie abruptamente com adição de um ácido ou uma base.

Esses íons podem ser adicionados ao meio como forma de ajudar no controle de pH (reduzindo a sensibilidade de variação do meio). Entretanto, esses mesmos íons podem dificultar o controle do pH, caso o valor desejado esteja fora da faixa em que o equilíbrio tampão ocorre.

Diversas técnicas de controle são propostas para resolver o problema de neutralização de uma solução aquosa, como: controlador adaptativo baseado em modelo (HENSON E SEBORG, 1994), MPC de múltiplos modelos lineares (GALÁN *et al.*, 2004), MPC não linear (HERMANSSON E SYAFIIE, 2015).

2.5 Conclusão

O modelo de predição é peça importante para a formulação de controladores do tipo MPC e a literatura apresenta diversos tipos de modelos que podem ser aplicados. Ao mesmo tempo, o processo de modelagem é um dos grandes desafios para a aplicação de MPC em processos reais.

A utilização de modelos neurais facilita o processo de modelagem, no sentido de abstrair conceitos fenomenológicos do processo além de resultar em um modelo com resposta mais rápida, em geral, comparado com modelagens fenomenológicas. Porém, para garantir que o modelo represente bem o processo é necessário um grande volume de dados de treinamento, que considere o máximo de condições operacionais possíveis. A obtenção desses dados a partir do processo real pode não ser possível por limitações de segurança, altos custos de matéria prima e de intervenções no processo.

Desta forma, propõe-se a utilização de um modelo fenomenológico para a obtenção de dados para o treinamento *offline* da RNA usada como modelo de predição do MPC. O uso de um modelo fenomenológico para geração de um banco de dados permite que diversas condições sejam testadas e de forma mais rápida, comparado com experimentos no processo real.

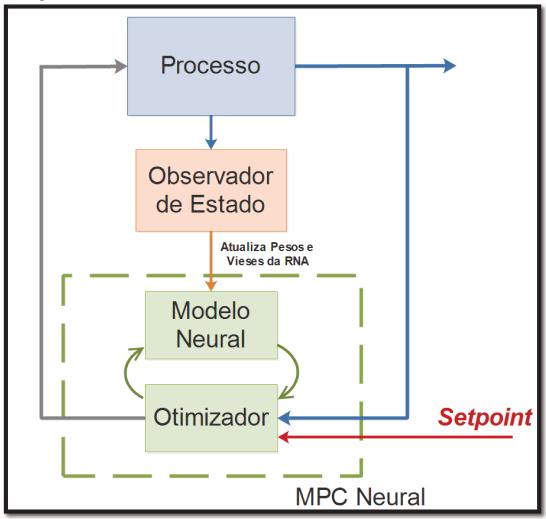
Por mais que o modelo fenomenológico seja minucioso, ele está sujeito a distúrbios e incertezas que vão resultar em desvios em relação ao processo real. Esses desvios podem interferir no desempenho do controlador MPC. Como forma de contornar essa limitação propõe-se o uso do algoritmo de aprendizado em tempo real baseado no filtro de Kalman estendido (FKE).

A literatura (IIGUNI *et al.* (1992), PUSKORIUS E FELDKAMP (1994) e CAO (2008)) apresenta bons resultados do algoritmo de FKE no treinamento de RNA que modelam processos. Apesar do algoritmo envolver a linearização da RNA (cálculo da matriz Jacobiana) e a inversão de matrizes, seus cálculos não são iterativos e, portanto, são rapidamente resolvidos possibilitando o uso como algoritmo de treinamento em tempo real.

Diante desse cenário foi levantada a hipótese de usar o algoritmo de FKE para aprimorar o modelo neural de predição do MPC em tempo real (diagrama mostrado na Figura 2.4). Espera-se que, além de aprimorar a predição da RNA em relação ao processo real, o algoritmo seja capaz de adaptar o modelo em situações em que o processo sofra distúrbios não considerados durante o treinamento da rede.

A motivação para essa estratégia é evitar experimentos e intervenções no processo real, com o intuito de, em uma possível aplicação industrial, evitar paradas no processo, custos com experimentos e a necessidade de intervenção humana para ajustes do controlador em campo.

Figura 2.4. Diagrama de blocos do controlador MPC Neural com adaptação do modelo em tempo real.



3 METODOLOGIA

Nesse capítulo são apresentados os métodos usados para o desenvolvimento da pesquisa. Inicialmente serão apresentados os processos nos quais os controladores foram aplicados. Em seguida são apresentados o modelo fenomenológico (usado para levantamento de dados e testes preliminares dos controladores) e o modelo neural (modelo de predição do MPC).

Posteriormente, é apresentada a estrutura do controlador MPC usado, seguido pelo algoritmo de adaptação da RNA em tempo real (baseado em FKE). Por fim, são apresentadas as sequências de experimentos realizados e as metodologias de avaliação de desempenho dos controladores.

3.1 Processos

O desenvolvimento do controlador com modelo adaptativo e de uma metodologia para implementação deste foram os principais objetivos da presente pesquisa. Para isso foram considerados dois sistemas distintos com características multivariáveis e nãolineares: planta de tanques comunicantes; e planta de controle de pH.

3.1.1 Planta de Tanques Comunicantes

A montagem experimental de tanques comunicantes tem escala de bancada e é de uso educacional para implementação de lógica de controle em um sistema real, como apresentado nas fotos da Figura 3.1 (a) e (b). A planta está montada no Laboratório de Redes Industriais (LIRI) da Faculdade de Engenharia Química (FEQ) da Universidade de Campinas (UNICAMP).

O diagrama de instrumentação do processo é apresentado na Figura 3.1 (c). A planta possui dois tanques (V-101 e V-102), cada um alimentado por uma bomba (P-101 e P-102), e drenados continuamente por meio de válvulas manuais (HV-101 e HV-102) instaladas no fundo de cada um dos tanques. Além disso, os tanques são conectados por uma válvula manual (HV-103) localizada na parede lateral, próximo ao fundo dos tanques. O nível de cada um dos tanques é obtido por transdutores de pressão piezoelétricos (LT-101 e LT-102). Os sinais são enviados a uma placa de aquisição (YIC101) que faz a comunicação com o computador (PC-101) via porta USB.

Figura 3.1.Fotos da montagem de bancada de tanques comunicantes construída pela DidatiControl (a) tanques comunicantes; (b) placa de aquisição de dados; (c) Diagrama de instrumentação (P&ID) do processo.





V-101 V-102 V-103 V-103 V-102 P-101 P-101

Os dados coletados no computador são então processados e usados em ambiente de programação $Matlab^{\circledR}$ para os cálculos do controlador, retornando a tensão a ser

fornecida às bombas (variáveis manipuladas). O sinal calculado retorna à placa de aquisição, que envia o sinal aos inversores de frequência (JIT-101 e JIT-102) das bombas. Além disso, a montagem conta com um tanque auxiliar (V-103) que armazena a água do sistema. O detalhamento e as dimensões dos principais componentes da montagem experimental são detalhados na Tabela 3.1.

Tabela 3.1. Tabela com dados dos instrumentos da montagem de tanques comunicantes em escala de bancada.

Equipamentos	Código	Modelo	Características	
Tangues	V-101		Tanque de acrílico de 1,34 dm³ (2,0 dm x	
Tanques	V-102	-	0,4 dm x 1,67 dm)	
Bombas	P-101	Limpador de para-brisa	Capacidade máxima de 300 l/h; pressão	
DOITIDAS	P-102	- universal	de 3,0 bar; 12 V	
Válvulas	HV-101		Válvula do abortura manual do tino	
	HV-102	-	Válvula de abertura manual do tipo Esfera	
	HV-103		LSTCTU	
Transdutor de	LT-101	Modelo Siemens	Faixa de operação (0 — 10 kPa)	
Pressão	LT-102	MPX5010®	Taixa de operação (0 – 10 kFa)	
Placa de aquisição	YIC-101	National Instruments USB-6008®	Sinais de entrada e saída de 0 - 5 V	

3.1.2 Planta de controle de pH

A planta de controle de pH em escala piloto (foto da Figura 3.2 (a)), está montada no Laboratório de Automação e Controle de Processos (LCAP) da Faculdade de Engenharia Química (FEQ) da Universidade de Campinas (UNICAMP). O diagrama de instrumentação da planta é apresentado na Figura 3.2 (b).

No processo contínuo, uma solução tampão de NaHCO₃ de concentração 0,02 M, cujo pH inicial é de 8,3, deve ser neutralizada. A solução tampão foi escolhida para ter seu pH alterado nesse estudo de caso, por reduzir a sensibilidade da variável de saída (pH) a adição da solução ácida ou da solução básica, quando comparado ao sistema sem o agente tamponante (COSTA, 2010).

O controle do pH foi realizado com soluções de ácido e base fortes. A solução ácida usada foi de HNO₃ 0,03M e a solução básica foi de NaOH à mesma concentração (0,03M). O processo foi operado com a alimentação da solução tampão constante e independente do controlador. O controlador calcula o sinal a ser convertido para tensão

enviada às bombas das soluções de ácido e base (variáveis manipuladas) de forma a levar o pH (variável controlada) ao *setpoint*.

Figura 3.2. (a) Foto da planta em escala piloto de controle de pH. (b) Diagrama de instrumentação (P&ID) do processo.



(b) V-202 Ácido (HNO₃) SP рΗ P-202 Nível V-203 AIT 201 Base (NaOH) <u>LIT</u> 201 V-201 HV 201 V-204 Tampão (NaHCO₃) **Descarte** P-201

Fonte: Adaptado de SAFADY (2018)

O detalhamento e as dimensões dos principais componentes da montagem experimental são detalhados na Tabela 3.2.

Tabela 3.2. Detalhes dos principais instrumentos do processo de Controle de pH em escala de laboratório.

Equipamentos	Código	Modelo	Descrição
Tanque de mistura	V-201	-	Tanque em aço inoxidável de 6 l (0,3 m de diâmetro)
Reservatórios	V-202 V-203 V-204	Permution [®]	Barrilete em PVC de 100 l (0,45m x 0,71 m)
Bombas	P-201 P-202 P-203 P-204	Micropump® Modelo RZR-500 Micropump® Modelo RZ-73004 Micropump® Modelo RZ-73004 Micropump® Modelo 1800-00	Capacidade máxima de 180 l/h Capacidade máxima de 110 l/h Capacidade máxima de 110 l/h Capacidade máxima de 60 l/h
Peagâmetro	AIT-201	Mettler-Toledo® Modelo InPro 3250i SG/325	De haste longa com medidor de temperatura
Transdutor de Pressão	LIT-201	Smar® Modelo LD-301	Tipo diafragma; faixa 120:1
Sensor de vazão	FT-201 FT-202 FT-203 FT-204	Signet® Modelo GF-2502 Signet® Modelo 2507-2V Signet® Modelo 2507-2V Signet® Modelo 2507-2V	Faixa de operação 0,4 - 12 l/min Faixa de operação 0,4 - 12 l/min Faixa de operação 0,4 - 12 l/min Faixa de operação 0,4 - 12 l/min
CLP	PC-201	SMAR® Modelo DFI302	-

Fonte: Adaptado de COSTA (2010)

O controle de pH foi realizado em um tanque (V-201) agitado de 6 litros de volume máximo. O nível do tanque foi controlado por uma malha de controle externa, do tipo PI programado no próprio CLP ligado ao processo, que manipula a tensão fornecida à bomba de drenagem do tanque (P-201), mantendo o nível sempre a 60 % da capacidade máxima.

Todos os dados foram coletados por meio de um CLP ligado à planta e conectado a um computador. Neste último é possível fazer a leitura dos dados do processo e efetuar os cálculos do controlador MPC.

3.2 Modelagem dos Processos

Para os dois processos propostos nesse estudo foram realizadas as modelagens fenomenológicas e a modelagem neural. O objetivo da modelagem fenomenológica foi simular computacionalmente o comportamento do processo, para obtenção dos dados

necessários para treinamento do modelo neural e para a avaliação preliminar do controlador desenvolvido. Essa estratégia foi adotada para agilizar o desenvolvimento do controlador.

A opção em usar a modelagem fenomenológica para obter os dados de treinamento para a rede neural deu maior versatilidade em diversificar a amplitude dos testes, sem danificar ou gerar alguma situação de risco durante a operação dos experimentos. Ressalta-se que é esperado que os desvios do modelo fenomenológico e processo, inseridos no modelo neural devido ao uso de dados simulados para treinamento, sejam corrigidos pelo algoritmo proposto de adaptação *online* do modelo.

3.2.1 Modelagem fenomenológica

Os modelos fenomenológicos de parâmetros concentrados foram desenvolvidos a partir das equações de balanço de massa dos processos, seguindo a teoria exposta por RAMLI *et al.* (2009) e KHALID E KADRI (2012) para o processo de tanques comunicantes e HENSON E SEBORG (1994) e OLIVEIRA *et al.* (2007) para o processo de controle de pH.

3.2.1.1 Modelo do processo de tanque comunicantes

O processo de tanques comunicantes foi composto por dois tanques, cada um deles com uma válvula de drenagem (saída) e conectados por uma válvula de abertura fixa, posicionadas no fundo dos tanques. A alimentação foi realizada por duas bombas independentes (a tensão fornecida a cada uma das bombas são as variáveis manipuláveis do processo).

Para cada um dos tanques, foi equacionado o balanço de massa apresentado nas Equações 3.1 e 3.2.

$$S_{V101} \cdot \frac{dL_{V101}}{dt} = Q_{P101} - Q_{HV101} \pm Q_{HV103}$$
3.1

$$S_{V102} \cdot \frac{dL_{V102}}{dt} = Q_{P102} - Q_{HV102} \mp Q_{HV103}$$
 3.2

em que S_{V101} e S_{V102} são as áreas dos tanques V-101 e V-102 (em dm^2); L_{V101} e L_{V102} são as alturas da coluna de água dos respectivos tanques (em dm); Q_{P101} e Q_{P102} são as vazões volumétricas de alimentação dos tanques V-101 e V-102 (em l/s); Q_{HV101} e Q_{HV102} são as vazões volumétricas das válvulas de drenagem dos tanques e Q_{HV103} é a vazão volumétrica na válvula de comunicação entre os dois tanques, todas em l/s. Nesse balanço, a densidade das correntes foi considerada constante.

As vazões foram modeladas seguindo as Equações 3.3 a 3.5.

$$Q_{Pi} = k_{Pi} * \log(u_{Pi}) + u_{0_{Pi}}$$
 $u_{Pi} \in [0, 1] \ e \ i = 101 \ ou \ 102$ 3.3

$$Q_{HVi} = k_{HVi} \cdot \sqrt{L_{Vi}}$$
 $i = 101 \text{ ou } 102$ 3.4

$$Q_{HV103} = k_{HV103} \cdot \sqrt{|L_{V102} - L_{V101}|}$$
 3.5

As constantes de válvulas (k_{HV101} , k_{HV102} e k_{HV103}) e o modelo das bombas que convertem os sinais enviados às bombas (u_{P101} e u_{P102}) em vazão volumétrica foram obtidos experimentalmente no processo.

Inicialmente, os transdutores de pressão foram calibrados a partir da leitura do sinal recebido para o nível observado nos tanques. Os gráficos de calibração são expostos no Apêndice A. O modelo das bombas foi identificado medindo o volume de líquido bombeado em um intervalo de tempo. Para isso, com as válvulas fechadas, as bombas foram acionadas por um intervalo de tempo, o nível ao final deste intervalo foi lido pelos transdutores de pressão e o volume calculado. Esse procedimento foi repetido para diferentes sinais de tensão e os dados são apresentados no Apêndice B.

A identificação das válvulas, HV-101 e HV-102 foi realizada com a verificação do estado estacionário dos tanques sem a comunicação (HV-103 fechada). A identificação foi realizada, abrindo a válvula (HV-101 ou HV-102) parcialmente e com uma tensão fixa, fornecida à bomba, aguardou-se o estado estacionário. Com a leitura do nível no estado estacionário a constante da válvula foi calculada pela Equação 3.6.

$$k_{HVi} = \frac{Q_{Pi}}{\sqrt{L_{Vi}}}$$
 $i = 101 \text{ ou } 102$ 3.6

Os resultados da identificação das constantes das válvulas são apresentados no Apêndice C.

A válvula de comunicação (HV-103) só foi então identificada após a determinação dos outros modelos usando um procedimento semelhante ao de obtenção das constantes das válvulas de drenagem.

A válvula HV-103 foi parcialmente aberta e uma das bombas (P-101 ou P-102) foi ligada com a tensão constante. O nível atingido após o regime permanente dos dois tanques foi lido e usado para calcular a constante da válvula usando a Equação 3.7.

$$k_{HV103} = \frac{Q_{Pi} - k_{HVi} \cdot \sqrt{L_{Vi}}}{\sqrt{|L_{V102} - L_{V101}|}} \qquad i = 101 \text{ ou } 102$$
3.7

O procedimento foi realizado duas vezes (cada vez com uma das bombas) e os resultados são apresentados no Apêndice C.

Com o uso do software *Matlab Simulink*® 2013a, o sistema de equações diferenciais (Equações 3.1 a 3.7) foi implementado e resolvido usando o método Implícito de *Runge-Kutta* (função "ode23tb" do pacote de solução de equações diferenciais do *Matlab*®). O método foi escolhido de forma a garantir a solução da simulação, uma vez que o método é indicado para solução de equações diferenciais em que as derivadas variam rapidamente (problemas *stiff*). Embora não tenha sido avaliada a necessidade de uso desse método.

O modelo obtido tem como entradas o sinal (percentual normalizado) de tensão fornecida às bombas *up101* e *up102*, variando no intervalo [0, 1] e as saídas são os níveis dos tanques normalizados *Nv101* e *Nv102* (Equação 3.8), variando no intervalo [0, 1].

$$N_{Vi} = \frac{L_{Vi}}{L_{max_{Vi}}}$$
 i = 101 ou 102

3.2.1.2 Modelo do tanque de controle de pH

O sistema de controle de pH foi simulado a partir dos balanços molares em um tanque de mistura perfeita. Foi considerado que o tanque é alimentado por três soluções: 1- Ácido (HNO₃); 2- Base (NaOH) e; 3- Tampão (NaHCO₃). As equações de balanço por espécie e a equação diferencial que descreve a dinâmica do nível no tanque são apresentadas a seguir (Equações 3.9 a 3.12):

HNO₃:
$$S_{V201}.L_{V201}.\frac{dC_{1,V201}}{dt} = C_{1,V202}.Q_{P202} - C_{1,V201}.Q_{P201}$$
 3.9

NaOH:
$$S_{V201}.L_{V201}.\frac{dC_{2,V201}}{dt} = C_{2,V203}.Q_{P203} - C_{2,V201}.Q_{P201}$$
 3.10

NaHCO₃:
$$S_{V201}.L_{V201}.\frac{dC_{3,V201}}{dt} = C_{3,V204}.Q_{P204} - C_{3V201}.Q_{P201}$$
 3.11

Nível:
$$S_{V201} \cdot \frac{dL_{V201}}{dt} = Q_{P202} + Q_{P203} + Q_{P204} - Q_{P201}$$
 3.12

A modelagem das bombas P-202, P-203 e P-204 foi realizada ajustando os dados obtidos do processo ao modelo da Equação 3.13. Os dados foram obtidos em testes de malha aberta em que o sinal de tensão fornecida à bomba era selecionado e a vazão resultante foi lida dos medidores de vazão. As curvas de calibração, com as quais foram obtidos os parâmetros do modelo (k_{Pi} e u_{0Pi}) são expostos no Apêndice D.

$$Q_{Pi} = k_{Pi} \cdot u_i + u_{0_{Pi}}$$
 $u_i \in [0, 1] \text{ e } i = 202, 203 \text{ e } 204$ 3.13

A bomba P-201 foi modelada usando os dados do fabricante de vazão máxima, considerando que a tensão fornecida resultava na vazão proporcional Equação 3.14.

$$Q_{P201} = \frac{180}{3600} \cdot u_i [l/s] \qquad u_i \in [0, 1]$$
 3.14

Essa simplificação foi realizada, uma vez que a malha de controle de nível da planta é independente do controle de pH e o valor exato dessa vazão não é relevante para o desenvolvimento do trabalho, somente que o processo seja mantido a nível constante.

Para garantir o nível constante, uma malha PID foi fechada na simulação. Os parâmetros do controlador PID foram ajustados usando o método proposto por RICE E COOPER (2008). Os parâmetros obtidos são apresentados na Tabela 3.3.

Tabela 3.3. Parâmetros do Controlador PID para o nível do tanque de mistura.

Parâmetro	Valor
k _C	-22,86 [u.a.]
$ au_I$	21 [s]
$ au_D$	1,4 [s]

Vale ressaltar que os parâmetros do Controlador PID calculados para a simulação não foram usados no processo real. No processo real considerou-se o controlador PI já implementado em trabalhos anteriores (COSTA (2010); SENCIO (2017); e SAFADY (2018)).

Para simular o peagâmetro, foi construído um algoritmo que recebe a informação de concentração das espécies no volume do reator (considerando a hipótese de mistura perfeita). O pH é estimado buscando a concentração de H⁺ e dos outros íons presentes na solução (Equações 3.15 a 3.21) que zere o balanço de cargas (Equação 3.22). O sistema de equações foi resolvido usando o método de pontos interiores, disponível no toolbox de otimização não linear do *Matlab** (função "*fmincon*"). O fluxograma do algoritmo para calcular o pH do sistema é apresentado na (Figura 3.3).

$$[H^{+}] = 10^{-pH}$$

$$[OH^{-}] = 10^{-(14-pH)}$$

$$[H_{2}CO_{3}] = C_{3,V201} \cdot (1 - \alpha_{ac1} - \alpha_{ac2})$$

$$[HCO_{3}^{-}] = C_{3,V201} \cdot \alpha_{ac1}$$

$$[CO_{3}^{2-}] = C_{3,V201} \cdot \alpha_{ac2}$$

$$[NO_{3}^{-}] = C_{1,V201}$$
3.15
3.16
3.17
3.18
3.19

$$[Na^+] = C_{3,V201} + C_{2,V201}$$
3.21

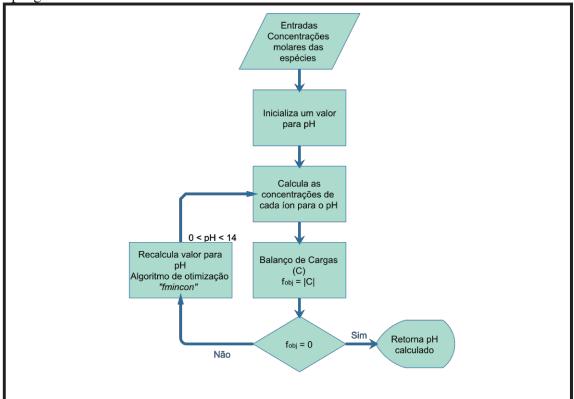
$$Carg = -[HCO_3^-] - 2 \cdot [CO_3^{2-}] - [NO_3^-] + [Na^+] + [H^+] - [OH^-]$$
 3.22

em que α_{ac1} e α_{ac2} são fatores de dissociação dos íons HCO_3^- e CO_3^{2-} , respectivamente, e são calculados pelas Equações 3.23 e 3.24 (OLIVEIRA *et al.*, 2007):

$$\alpha_{ac1} = \left\{ 1 + \frac{k_{ac1}}{[H^+]} + \frac{k_{ac1} \cdot k_{ac2}}{[H^+]^2} \right\}^{-1} \cdot \frac{k_{ac1}}{[H^+]}$$
 3.23

$$\alpha_{ac2} = \left\{ 1 + \frac{k_{ac1}}{[H^+]} + \frac{k_{ac1} \cdot k_{ac2}}{[H^+]^2} \right\}^{-1} \cdot \frac{k_{ac1} \cdot k_{ac2}}{[H^+]^2}$$
3.24

Figura 3.3. Fluxograma do algoritmo de cálculo do pH usado para simular o peagâmetro.



Fonte: Adaptado de OLIVEIRA et al. (2007).

As constantes de dissociação do ácido H_2CO_3 , k_{ac1} e k_{ac2} são $10^{-6,37}$ e $10^{-10,33}$ e foram obtidos de ATKINS E DE PAULA (2011), respectivamente. No modelo do tanque de controle de pH, as variáveis de entrada são: o sinal de tensão enviado às bombas de alimentação das soluções (ácido, base e tampão) e o *setpoint* de nível do tanque. A saída do modelo é o pH da solução no volume do reator.

3.2.2 Modelagem Neural

Para cada um dos processos (Tanque de nível e Controle de pH), uma rede neural foi treinada usando os dados da simulação fenomenológica. Esse modelo neural (representado pelas Equações 3.25 a 3.28) foi usado como modelo inicial no controlador MPC de modelo adaptativo e no MPC sem adaptação que servirá de referência para avaliar a efetividade do algoritmo de adaptação.

Trata-se de uma rede do tipo *feedfoward* com uma camada de entrada (Equação 3.25), uma camada escondida de 3 neurônios (Equação 3.26), cuja função de ativação é do tipo Tangente Hiperbólica (Equação 3.27). Na camada de saída de rede os neurônios têm função de ativação linear (Equação 3.28).

$$x^{(j)} = W^{(j)}.E - b^{(j)}$$
 em que $j = ent$ 3.25

$$x^{(j)} = W^{(j)}. y^{(j-1)} - b^{(j)}$$
 em que $j = esc \ e \ sai$ 3.26

$$y^{(j)} = f_j(x^{(j)}) = \frac{\exp(x^{(j)}) - \exp(-x^{(j)})}{\exp(x^{(j)}) + \exp(-x^{(j)})} \quad em \ que \ j = ent \ e \ esc$$
 3.27

$$y_{rede} = f_i(x^{(j)}) = x^{(j)}$$
 em que $j = sai$ 3.28

em que $W^{(j)}$ é a matriz de pesos e $b^{(j)}$ os vieses dos neurônios da camada 'j'. Na camada de entrada (j = ent) o somatório (ponderado) é realizado com os sinais de entrada (E) e nas demais camadas, escondida (esc) e de saída (sai), são usadas as respostas dos neurônios da camada anterior $(y^{(j-1)})$.

Os dados, antes de serem alimentados ao modelo neural são normalizados entre 0 e 1, dividindo-os pelo seu limite superior (definidos de acordo com as dimensões do processo real). Dessa forma, foram adicionadas restrições ao modelo de somente retornar valores dentro dos limites possíveis ao processo real.

As entradas e saídas do modelo neural do processo de Tanques Comunicantes são apresentadas nas Tabela 3.4.

Tabela 3.4. Variáveis de entrada e saída da rede neural do processo de tanques comunicantes.

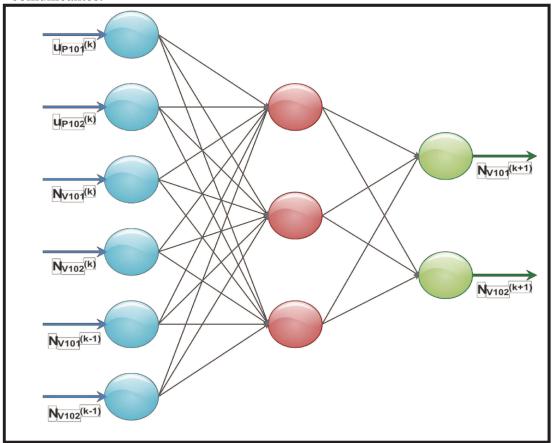
Variáveis	Descrição	Faixa absoluta
	Entradas	
$u_{P101(k)}$	Tensão fornecida à bomba P-101 no instante de amostragem	0-5 Volts
<i>UP102 (k)</i>	Tensão fornecida à bomba P-102 no instante de amostragem	0-5 Volts

Tabela 3.4. Variáveis de entrada e saída da rede neural do processo de tanques comunicantes. (Continuação)

Variáveis	Descrição	Faixa absoluta
$N_{V101\ (k)}$	Nível do tanque V-101 no instante de amostragem	0 – 1,67 dm
$N_{V102(k)}$	Nível do tanque V-102 no instante de amostragem	0 - 1,67 dm
$N_{V101\ (k-1)}$	Nível do tanque V-101 no instante passado (atraso)	0 - 1,67 dm
$N_{V102\;(k-1)}$	Nível do tanque V-102 no instante passado (atraso)	0 - 1,67 dm
	Saídas	
$N_{V101(k+1)}$	Nível do tanque V-101 no instante futuro (predição)	0 - 1,67 dm
$N_{V102\;(k+1)}$	Nível do tanque V-102 no instante futuro (predição)	0 - 1,67 dm

A estrutura da rede neural usada para modelar o processo de tanques comunicantes é apresentada na Figura 3.4.

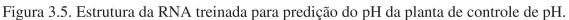
Figura 3.4. Estrutura da RNA treinada para predição do nível da planta de tanques comunicantes.

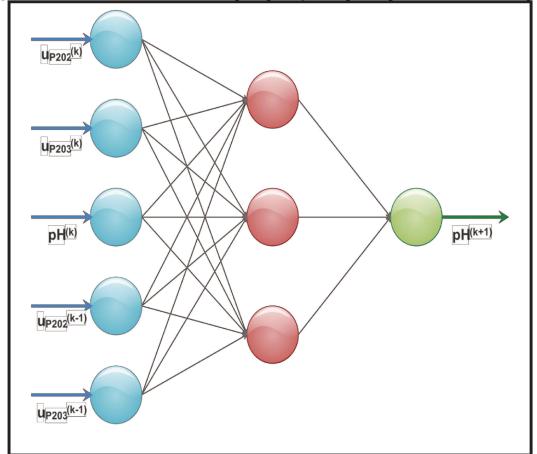


E as entradas e saídas do modelo neural do processo de controle de pH são apresentadas na Tabela 3.5.

	. Variáveis de entrada e saída da rede neural do processo de	
Variáveis	Descrição	Faixa absoluta
	Entradas	
U P202 (k)	Sinal enviado à bomba P-202 no instante de amostragem (ácido)	0 - 5 V
U P203 (k)	Sinal enviado à bomba P-203 no instante de amostragem (base)	0 - 5 V
$pH_{(k)}$	pH normalizado no instante de amostragem	0 - 14
U P202 (k-1)	Sinal enviado à bomba P-202 no instante passado (atraso)	0 - 5 V
U P203 (k-1)	Sinal enviado à bomba P-203 no instante passado (atraso)	0 - 5 V
	Saída	
$pH_{(k+1)}$	pH normalizado no instante futuro (predição)	0 - 14

A estrutura da rede neural usada para modelar o processo de controle de pH é apresentada na Figura 3.5.





Os dados para treinamento das redes foram obtidos em ensaios simulados do tipo sequência de degrau (*staircase*) em que a duração de cada degrau foi constante. O vetor de condições de entrada para o treinamento é composto pelas variáveis manipuladas de cada um dos processos e pelos valores escolhidos para os degraus, que foram determinados empiricamente de forma a obter o máximo de dados possível dentro dos limites das variáveis de saída das redes (níveis e pH). A ordem dos degraus foi determinada aleatoriamente repetindo cada combinação 3 vezes.

O intervalo entre cada amostragem de dados para treinamento é o mesmo intervalo de amostragem do controlador (aproximadamente 10 % da constante de tempo do processo em malha aberta). Por se tratar de processos não lineares, foram calculadas as constantes de tempo médias de cada processo. Ou seja, a média aritmética dos valores identificados em testes do tipo degrau em malha aberta. Vale ressaltar que os testes foram realizados usando o modelo fenomenológico das plantas em malha aberta.

O banco de dados obtido foi organizado de forma que a cada amostragem, os valores instantâneos das variáveis e seus valores atrasados (serie temporal) fossem armazenados. Dessa forma, foi possível randomizar e particionar o banco de dados para treinamento (75 % dos dados) e validação (25 % dos dados).

A rede neural foi treinada usando o método de *Levenberg Marquardt* com Regularização Bayesiana usando o *toolbox* de redes neurais artificiais do *Matlab*[®].

3.3 Controlador Preditivo Baseado em Modelo

O controlador MPC Neural sem adaptação do modelo tem uma RNA de pesos e vieses fixos (treinada com dados da simulação fenomenológica como explicado na Seção 3.2.2). Esse modelo é usado para prever o comportamento das variáveis controladas do processo com a alteração dos valores das variáveis manipuladas da malha de controle.

No processo de níveis comunicantes, as variáveis manipuladas são os potenciais aos quais as bombas P-101 e P-102 foram submetidas, e as variáveis controladas são os níveis dos tanques V-101 e V-102. No processo de controle de pH, as variáveis manipuladas são os sinais de tensão enviados aos inversores de frequência das bombas P-202 e P-203 e a variável controlada é o pH no tanque de mistura (V-201).

A determinação da manipulação, calculada pelo controlador, é realizada por um algoritmo de otimização, buscando minimizar uma função custo a partir da variação das variáveis manipuladas, sujeita as restrições de acordo com a Equação 3.29.

$$\min J^{(k)} = \varphi(u_1, u_2)$$
 sujeito a:
$$\begin{cases} 0 \le u_1 \le 1 \\ 0 \le u_2 \le 1 \end{cases}$$
 3.29

em que $J^{(k)}$ é a função objetivo do método de otimização, u_1 e u_2 são vetores das variáveis manipuladas para cada um dos processos. Esses vetores, estão sujeitos à restrição de somente assumir valores entre 0 e 1 durante a otimização.

A otimização foi realizada com um algoritmo de otimização não linear com restrição usando o método de Pontos Interiores, disponível no *toolbox* de otimização do *Matlab*[®] (*fmincon*). Os parâmetros usados na otimização são apresentados na Tabela 3.6.

Tabela 3.6. Parâmetros do método de otimização (fmincon).

Opção	Valor
Método	Pontos Interiores (interior-point)
Número máximo de iterações	1000
Tolerância da função objetivo	1.10^{-6}
Restrição	Variáveis manipuladas ϵ [0,1]

O algoritmo de Pontos Interiores foi escolhido para a solução do problema com base no trabalho de MORARI E LEE (1999), que descrevem essa classe de método como a mais ágil na solução do problema. Testes com outros métodos foram realizados não sendo observadas mudanças no desempenho do controlador em experimentos simulados.

Por terem características diferentes, a função custo do controlador da planta de nível comunicantes é diferente da função custo da planta de pH. O processo de tanques comunicantes possui duas variáveis manipuladas para duas variáveis controladas, é um processo de Múltiplas Entradas e Múltiplas Saídas (do inglês *MIMO*). A função-objetivo do MPC de controle de nível é dada pela Equação 3.30. MACIEJOWSKI (2002) sugere a aplicação de uma trajetória de referência para as variáveis controladas (Equações 3.31 e 3.32) de forma a suavizar a aproximação ao *setpoint*. O parâmetro de suavização da trajetória de referência (α) assume valores entre 0 e 1, sendo que valores próximos de 0 fazem com que a trajetória seja do tipo degrau. Esse parâmetro foi definido em testes do controlador no processo simulado.

$$J^{(k)} = P_{erro\ SP} \cdot \sum_{i=k}^{k+HP} \left[\left(N_{V101} - N_{V101_{ref}} \right)_{i}^{2} + \left(N_{V102} - N_{V102_{ref}} \right)_{i}^{2} \right]$$

$$+ P_{\Delta u} \cdot \sum_{i=k}^{k+HC} \left[\left(u_{P101(i)} - u_{P101(i-1)} \right)^{2} + \left(u_{P102(i)} - u_{P102(i-1)} \right)^{2} \right]$$

$$3.30$$

$$N_{ref_{V_{101}}} = (\alpha^i). N_{V_{101,(k)}} + (1 - \alpha^i). N_{V_{101,SP}}$$

$$i = 0.1.2.3 \dots HP$$
3.31

$$N_{ref_{V_{102}}} = (\alpha^i).N_{V_{102,(k)}} + (1 - \alpha^i).N_{V_{102,SP}}$$

$$i = 0,1,2,3 \dots HP$$
3.32

No processo de controle de pH o funcionamento simultâneo das duas bombas faz com que uma anule o efeito da outra, resultando em consumo maior de reagentes, diluição do meio reacional e resultando em um pH que poderia ser obtido apenas com a adição de uma das soluções. Portanto, foi necessário adaptar a função objetivo de forma a penalizar a adição simultânea do ácido e da base (restrição flexível). A função objetivo para o problema é dada pela Equação 3.33, sendo a Equação 3.34 a trajetória de referência desejada que o controlador deverá seguir para atingir o *setpoint*.

$$J^{(k)} = P_{erroSP}. \sum_{i=k}^{k+HP} \left[\left(pH - pH_{ref} \right)_{i}^{2} \right]$$

$$+ P_{\Delta u}. \sum_{i=k}^{k+HC} \left[\left(u_{P202(i)} - u_{P202(i-1)} \right)^{2} + \left(u_{P203(i)} - u_{P203(i-1)} \right)^{2} \right]$$

$$+ P_{u}. \left(u_{P202(k)} \cdot u_{P203(k)} \right)$$

$$pH_{ref} = (\alpha^{i}). pH_{(k)} + (1 - \alpha^{i}). pH_{SP}$$

$$i = 0,1,2,3 \dots HP$$

$$3.34$$

Em ambos os problemas, os pesos da função objetivo (P_{erroSP} , $P_{\Delta u}$ e P_u) foram ajustados com testes no processo simulado usando o controlador MPC sem adaptação do modelo. O horizonte de predição (HP) foi, também, definido empiricamente de forma a manter próximo ao recomendado de 60 % da constante de tempo do processo em malha aberta (MACIEJOWSKI, 2002). Esses parâmetros foram mantidos nos testes com o controlador de modelo adaptativo e nos testes no processo real.

O controlador MPC Neural de Modelo Adaptativo, antes de realizar a etapa de otimização, faz ajustes em seu modelo de predição usando um algoritmo de aprendizado em tempo real baseado no método de filtro de Kalman estendido.

Após atualização dos pesos e vieses da rede neural (modelo de predição), a nova rede é usada no mesmo algoritmo do MPC Neural sem adaptação do modelo (descrito anteriormente).

Essa sequência de cálculos é realizada a cada amostragem do controlador. O intervalo de amostragem foi definido como aproximadamente 10 % da constante de tempo em malha aberta do processo, como recomendado por STEPHANOPOULOS (1984).

3.4 Algoritmo de adaptação do modelo

O algoritmo de adaptação do modelo usado é baseado no algoritmo do filtro de Kalman estendido (FKE). O algoritmo foi desenvolvido com base no algoritmo disponibilizado por CAO (2008) para treinamento de redes neurais usando o filtro de Kalman estendido. O fluxograma com as principais etapas do algoritmo é mostrado na Figura 2.3 (b).

Durante cada etapa de atualização, realizada pelo algoritmo de FKE, a RNA deve ser linearizada, e para isso o algoritmo realiza o cálculo da matriz jacobiana em relação aos pesos da rede. A matriz jacobiana é obtida realizando a derivada parcial do RNA em relação a cada um dos pesos (parâmetros estimados pelo FKE) usando o método de diferenciação complexa (CAO, 2008).

O método da diferenciação complexa consiste na aplicação de um distúrbio no domínio dos números complexos a uma função e a derivada é observada na parte real da resposta da função. ABREU *et al.* (2013) apresenta um estudo sobre o método em que demostra que a derivada de uma função pode ser aproximada por meio da Equação 3.35

$$\frac{\delta\varphi}{\delta x_j} \approx \operatorname{Im}\left(\frac{\varphi(x_j + i.\,\Delta x_j)}{\Delta x_j}\right)$$
 3.35

Com a matriz jacobiana, a RNA é organizada na forma de espaços de estados (Equações 2.3 e 2.4) e o modelo linearizado é usado para atualizar os pesos e vieses da rede usando o método do filtro de Kalman, representado pelas Equações 3.36 a 3.40.

Predição:

$$\hat{X}_{(k|k)} = \hat{X}_{(k|k-1)} \tag{3.36}$$

$$P_{(k|k)} = P_{(k|k-1)} + q 3.37$$

Atualização

$$U_{FK} = P_{(k|k)} \cdot Jac^{t} \cdot (Jac \cdot P_{(k|k)} \cdot Jac^{-1} + R)^{-1}$$
3.38

$$\hat{X}_{(k+1|k)} = \hat{X}_{(k|k)} + U_{FK} \cdot (Z_k - \hat{Y}_k)$$
3.39

$$P_{(k+1|k)} = (I - U_{FK} \cdot Jac) \cdot P_{(k|k)}$$
3.40

em que \hat{X} é o vetor de estados a serem atualizados (pesos e vieses da RNA), P é a matriz de covariância do FKE, Jac é a matriz jacobiana da RNA em relação aos pesos e vieses, Z_k é o vetor de dados requisitados do processo, \hat{Y}_k é o vetor resposta da RNA, U_{FK} é a matriz de ganhos do filtro e I é a matriz identidade.

Para ambos os processos (planta de tanques comunicantes e planta de controle de pH) q (variância da variável obtida do processo) e R (variância da resposta do modelo) foram fixados em 0,01 e 0,1. O ajuste fino destes parâmetros influencia a taxa e a eficiência do aprendizado no método, porém optou-se por valores genéricos para ambos os experimentos a fim de não aumentar a complexidade da aplicação do controlador em diferentes processos.

A melhoria da predição do modelo neural foi avaliada calculando o erro absoluto entre a resposta da rede e a leitura realizada no processo (leitura dos sinais dos sensores no processo real ou a resposta do modelo fenomenológico nos ensaios simulados). O erro de predição foi calculado pela Equação 3.41.

$$err = Z - y_{rede}$$
 3.41

3.5 Avaliação dos controladores

Os controladores desenvolvidos foram avaliados nos testes servo (mudança de *setpoint*) e testes regulatórios (distúrbios de processo). Foram avaliados primeiramente na simulação fenomenológica e, posteriormente, no processo real buscando observar diferenças no desempenho do controlador MPC Neural, com e sem adaptação do modelo. As listas de experimentos realizados para avaliação dos controladores no processo de tanques comunicantes e controle de pH são apresentadas, respectivamente, nas Tabelas 3.7 e 3.8.Tabela 3.7. Lista de experimentos realizados no processo de tanques comunicantes.

Tabela 3.7. Lista de experimentos realizados no processo de tanques comunicantes.

Ехр.	Tipo	Modelo do Controlador	Processo
1	Servo	Sem adaptação	Simulado
2	Servo	Adaptativo	Simulado
3	Servo	Sem adaptação	Real
4	Servo	Adaptativo	Real
5	Regulatório (Mudança na válvula HV-101)	Sem adaptação	Real
6	Regulatório (Mudança na válvula HV-101)	Adaptativo	Real

TD 1 1 2 0 T ' 4 1	•	1' 1	1 , 1 1 TT
Tabela 3.8 Lista de ex	nerimentos rea	lizados no process	a de controle de nH
Tabela 3.0 Lista de ex	oci illiciitos ica	iizados iio process	o de controle de pri.

Exp.	Tipo	Modelo do Controlador	Processo
1	Servo	Sem adaptação	Simulado
2	Servo	Adaptativo	Simulado
3	Servo	Sem adaptação	Real
4	Servo	Adaptativo	Real
5	Regulatório 1 (Mudança na vazão de tampão)	Sem adaptação	Real
6	Regulatório 1 (Mudança na vazão de tampão)	Adaptativo	Real
7	Regulatório 2 (Mudança na concentração de tampão)	Sem adaptação	Real
8	Regulatório 2 (Mudança na concentração de tampão)	Adaptativo	Real
9	Regulatório 3 (Mudança na concentração do ácido e base)	Sem adaptação	Real
10	Regulatório 3 (Mudança na concentração do ácido e base)	Adaptativo	Real
11	Servo regulatório de longa duração	Adaptativo	Real

Além de resultados gráficos, os controladores foram avaliados numericamente usando os coeficientes de desempenho: Somatório do Erro Quadrático (SSE); Somatório da Variação da Manipulação (SVM). Esses coeficientes são definidos pelas Equações 3.42 e 3.43 (STEPHANOPOULOS, 1984) e foram calculados usando o método de integração numérica de regra do trapézio.

$$SSE = \sum_{t=0}^{t_{final}} \left(z_{(k)} - SP_{(k)} \right)^2 \cdot \Delta t$$
3.42

$$SVM = \sum_{k=0}^{t_{final}} (u_{(k)} - u_{(k-1)})^2 \cdot \Delta t$$
 3.43

O desempenho dos controladores foi calculado com os dados normalizados (variando de 0 a 1) com amostragem de 1 s, intervalo de leitura dos dados do processo.

4 RESULTADOS

Conforme descrito anteriormente, o controlador desenvolvido foi avaliado por meio de sua aplicação em dois processos não lineares: planta de tanques comunicantes e planta de controle de pH. Os resultados serão apresentados separadamente.

4.1 Planta de Tanques Comunicantes

A apresentação dos resultados dos experimentos na planta de tanques comunicantes será dividida nas etapas de: modelagem e identificação; testes em malha fechada.

4.1.1 Modelagem e Identificação

A modelagem fenomenológica da planta de tanques comunicantes foi realizada a partir dos balanços mássicos dos tanques (Equações 3.1 e 3.2). Porém, os modelos das bombas e válvulas foram identificados no processo (como explicado na Seção 3.2.1.1 e os gráficos expostos no Apêndice C). Os parâmetros do modelo são apresentados na Tabela 4.1.

Tabela 4.1. Constantes do modelo fenomenológico de tanques comunicantes.

Constante	Valor calculado	Unidade
k _{HV101}	2,3 • 10 ⁻²	dm ^{5/2} /s
k _{HV102}	2,3 • 10-2	$dm^{5/2}/s$
k_{HV103}	2,2 • 10-2	$dm^{5/2}/s$
$S_{V101} = S_{V102}$	0,8	dm^2
L_{max}	1,67	dm
$ au_{mcute{d}io}$	76	S

A constante de tempo do processo ($\tau_{m\'edio}$) foi obtida na simulação a partir de perturbações do tipo degrau nas variáveis manipuladas (tensão das bombas). O gráfico de identificação é apresentado na Figura 4.1 e os valores de τ_i para cada perturbação realizada são sintetizados na Tabela 4.2.

A simulação resultou no $\tau_{medio} = 76,6 \, s$. A partir deste valor foi determinado que o intervalo entre cada amostragem do controlador seria 5s, partindo da recomendação de STEPHANOPOULOS (1984) de que seja aproximadamente 10 % da constante de tempo do processo. Porém, optou-se por arredondar o valor para facilitar a condução dos experimentos.

Figura 4.1. Experimento de identificação da planta de tanques comunicantes - simulação fenomenológica.

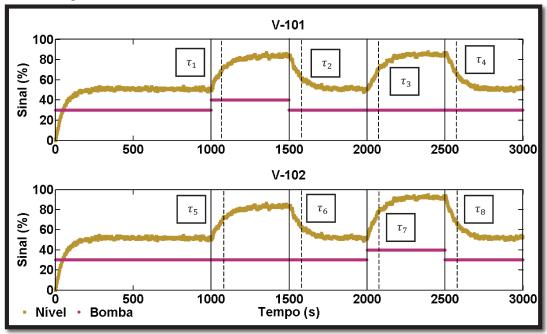


Tabela 4.2. Valores das constantes de tempo do processo de tanques comunicantes para cada uma das perturbações de identificação.

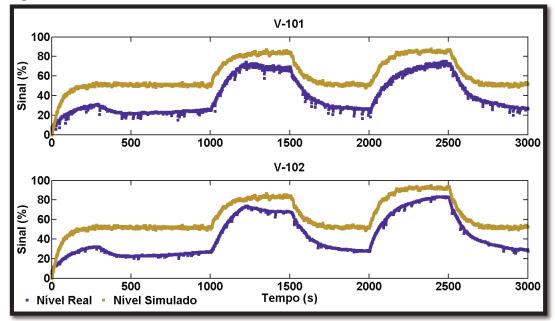
Tanque	1000 s	1500 s	2000 s	2500 s	
V-101	$\tau_1 = 71 s$	$\tau_2 = 82 s$	$\tau_3 = 74 s$	$\tau_4 = 72 s$	
V-102	$\tau_5 = 76 s$	$\tau_6 = 84 s$	$\tau_7 = 74 s$	$\tau_8 = 80 \ s$	

Além disso, a validação do modelo fenomenológico foi realizada comparando os dados experimentais com os dados simulados (acrescidos de ruídos). Os resultados do experimento de validação são apresentados na Figura 4.2.

O comportamento do processo real é bastante similar aos resultados obtidos na simulação, entretanto, há grande diferença entre o valor em regime estacionário, principalmente ao sinal de 30 % de tensão em ambas as bombas (apresentado na Figura 4.1), em que a diferença no nível chega a quase 20 pontos percentuais.

O desvio entre modelo e o processo real ocorre devido às incertezas dos dados usados para a identificação do processo. Uma vez que não havia medidores de vazão na alimentação dos tanques, as bombas foram modeladas medindo o volume de líquido bombeado durante determinado intervalo de tempo. Esse volume foi estimado com os dados de nível requisitados pelos transdutores de pressão e as dimensões dos tanques.

Figura 4.2. Comparação entre simulação computacional e experimento real da planta de tanques comunicantes.



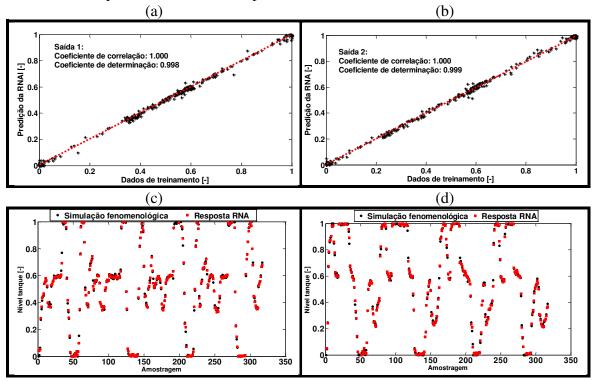
Outra explicação possível para a discrepância entre modelo e processo é que as válvulas instaladas são de abertura manual, e do tipo esfera (abertura rápida). Esse tipo de válvula é muito sensível a mudanças de abertura e por serem válvulas manuais, após qualquer manipulação é difícil garantir o retorno à mesma posição, sendo necessário a re-identificação do processo. Além disso, os modelos das válvulas foram obtidos com base na modelagem das bombas, propagando imprecisões.

Também chama a atenção que o nível, obtido pelos transdutores de pressão, antes de estabilizar no regime permanente, passa por um máximo. Esse comportamento não é representado na simulação. Dados de simulação apresentados por DULĂU E DULĂU (2016) também não descrevem esse comportamento. Duas hipóteses que podem explicar o máximo observado são: 1-Variações na vazão de bombeamento; 2- Efeitos da filtragem do sinal vindo do processo.

Mesmo com os problemas indicados, a modelagem fenomenológica foi usada para gerar dados para o treinamento da rede neural responsável por fazer as predições do controlador MPC. Os gráficos de comparação dos dados de treinamento e de regressão dos dados de validação são apresentados na Figura 4.3.

O treinamento foi realizado assumindo como dados de entrada a tensão fornecido às bombas P-101 e P-102, o nível dos tanques (V-101 e V-102) no instante da amostragem e em uma amostragem atrasada.

Figura 4.3. Resultados do Treinamento da RNA. a) e b) Comparação da rede e dados de treinamento do tanque V-101 e V-102, respectivamente; c) e d) Validação da RNA para o nível do tanque V-101 e V-102, respectivamente.



Os dados de saída foram os níveis dos tanques (V-101 e V-102) uma amostragem a frente, conforme descrito na seção 3.2.2 da Metodologia.

Como pode ser observado nas figuras o treinamento da rede resultou em um modelo que descreve bem os resultados simulados do processo e, por essa razão, esse foi o modelo usado no controlador neural e como modelo de partida no controlador neural adaptativo.

4.1.2 Testes em malha fechada

Com base nos resultados de constante de tempo do processo apresentados na Figura 4.1 e resumidos na Tabela 4.2 e na metodologia descrita na Seção 3.3, o horizonte de controle (HC) foi fixado em 3 amostragens (15 s) e horizonte de predição (HP) fixado em 10 amostragens (50 s).

Para a determinação dos pesos de penalização da função objetivo (Equação 3.30) foram comparados os SSE e os SVM do teste servo, usando o MPC sem adaptação do modelo. Os resultados são mostrados na Tabela 4.3.

Tabela 4.3. Experimentos para definição dos pesos de penalização da função objetivo do	
processo de tanques comunicantes.	

Evnorimonto	D	D	SSE	SSE	SVM	SVM	Soma
Experimento	$P_{erro,SP}$	$P_{\Delta u}$	V-101	V-102	P-101	P-102	Normalizada
1	1	1	80,27	26,21	0,77	0,59	1,41
2	1	10	91,08	32,86	0,09	0,12	1,10
3	1	50	121,62	49,40	0,01	0,02	1,42
4	10	1	79,77	25,78	2,23	1,02	2,14
5	10	50	85,07	29,28	0,20	0,21	1,10
6	50	1	79,73	25,75	2,55	1,09	2,29
7	50	10	79,75	25,97	1,90	0,96	1,99
Máximo			121	.,62	2,	55	

A escolha foi realizada com base na normalização dos SSE e SVM. Foi escolhido o par de parâmetros que resultou em menor soma dos coeficientes de desempenho. Como os experimentos 2 e 5 apresentaram o mesmo valor de soma normalizada, optouse pelo que apresentou menor SSE dentre os dois (experimento 5: $P_{\Delta u} = 50$ e $P_{erro,SP} = 10$).

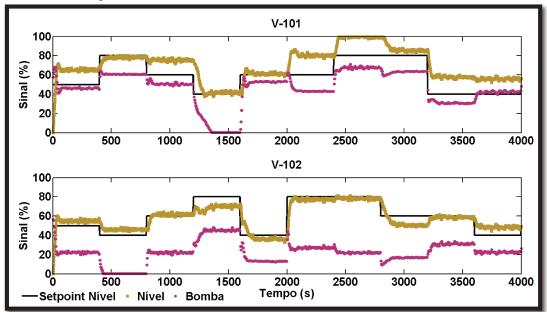
Esses testes foram realizados mantendo o fator de suavização da trajetória de referência (α) constante igual a 0,9. Esse valor foi obtido usando a Equação 4.1 (MACIEJOWSKI, 2002) como referência.

$$\alpha = \exp\left(-\frac{t_{amost}}{\tau_{m\acute{e}dio}}\right) \tag{4.1}$$

Os resultados, obtidos por simulação computacional, do teste servo do controlador MPC sem adaptação do modelo configurado são apresentados na Figura 4.4.

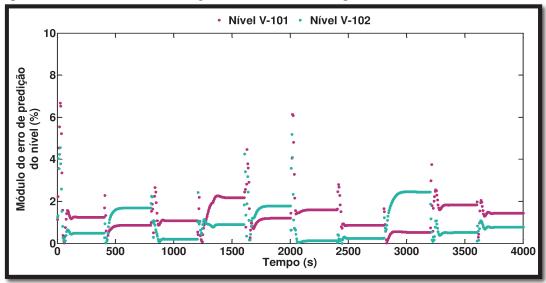
Nos resultados observa-se o controle satisfatório para alguns dos valores de *setpoint*, porém, para outros valores o controlador não consegue atingir o valor objetivo definido. Esse desvio permanente é atribuído à predição da rede neural. Apesar de apresentar bom ajuste aos dados de treinamento e de validação, a rede ainda tem dificuldades de descrever o modelo fenomenológico para todas as faixas de operação. Uma das explicações para essa dificuldade é a abrangência dos dados usados para treinamento e validação da rede. Observando os gráficos da Figura 4.3 observa-se que para algumas faixas de operação (principalmente par níveis altos ou muito baixos) poucos dados foram apresentados durante o treinamento.

Figura 4.4. Teste servo do controlador MPC Neural sem adaptação do modelo, processo simulado de tanques comunicantes.



O erro de predição do modelo neural (em relação ao modelo fenomenológico) é visualizado na Figura 4.5. Observa-se que sem a adaptação da rede neural, o erro de predição chega próximo de 5 % do nível total do tanque. O erro é mais pronunciado nos instantes de mudança de *setpoint*, porém mesmo depois que o controlador mantém o estado estacionário no processo são observados desvios do modelo resultando no *offset* observado na Figura 4.4.

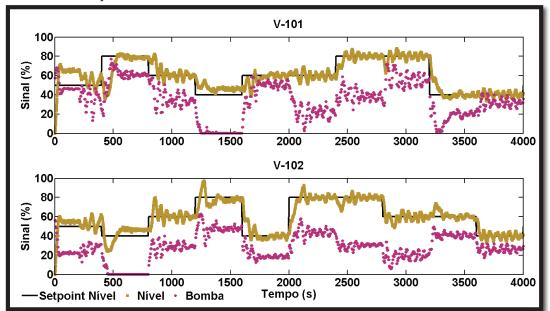
Figura 4.5. Módulo do erro de predição da rede neural (sem adaptação) em relação à resposta do modelo fenomenológico do sistema de tanques de nível comunicantes.



Aplicando o algoritmo de adaptação da rede durante o processo, a eficiência do controlador em rastrear o *setpoint* é melhorada, como apresentado na Figura 4.6.

No teste com adaptação do modelo, apesar de conseguir aproximar e manter o nível mais próximo ao *setpoint*, o controlador provoca muitas oscilações das variáveis controladas por consequência de oscilações na variável manipulada.

Figura 4.6. Teste servo do controlador MPC Neural com adaptação do modelo, processo simulado de tanques comunicantes.



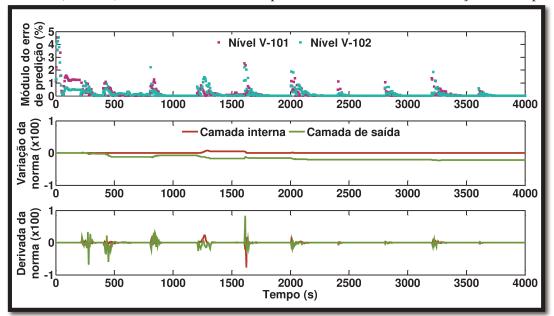
A dificuldade em realizar o controle é atribuída ao ruído adicionado às saídas do modelo, simulando o ruído do processo real. Esse ruído (número aleatório gerado de uma distribuição normal de média 0 e variância 0,001 [nível normalizado]) faz com que a leitura varie rapidamente independente da atuação.

Apesar do filtro de Kalman estendido ser uma estratégia capaz de lidar com ruídos nas variáveis medidas (HAYKIN, 2004) a convergência dos pesos da rede é dificultada com a existência do sinal aleatório.

Com o algoritmo de adaptação do modelo neural, no momento em que são realizados distúrbios no *setpoint*, são observados erros de predição, porém, após algumas amostragens, o algoritmo atualiza os parâmetros da RNA eliminando o erro de predição (como apresentado na Figura 4.7a). Na Figura 4.7 b observa-se a variação da norma da matriz de pesos e vieses de cada uma das camadas da rede. Observa-se que, a cada mudança de *setpoint*, o modelo é ajustado até que o erro de predição seja

eliminado, então a rede não sofre alteração até que seja percebido um novo desvio do modelo.

Figura 4.7. a) Módulo do erro de predição da rede neural (com adaptação) em relação à resposta do modelo fenomenológico do sistema de tanques de nível comunicantes. b) Norma da matriz de pesos e vieses de cada uma das camadas da rede neural. c) Derivada (discreta) da norma da matriz de pesos e vieses da rede em relação ao tempo.



A variação dos parâmetros da rede neural pode ser melhor observada na Figura 4.7 c, em que é apresentada a derivada (discreta) da norma das matrizes de pesos e vieses da rede neural em função do tempo. A cada perturbação de *setpoint* são observados valores diferentes de zero, significando que o modelo está sendo modificado e assim que o modelo é ajustado a derivada calculada é zero.

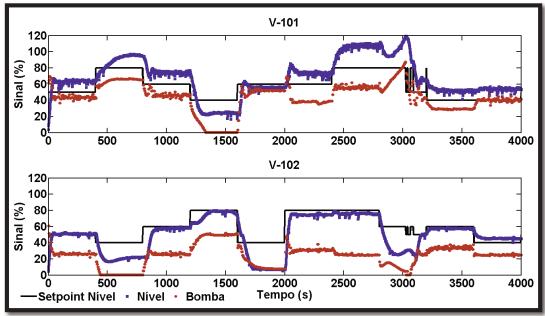
Nesse gráfico é possível observar que ao longo do experimento a alteração dos parâmetros da rede vai diminuindo a cada perturbação realizada, indicando que a rede vai evoluindo ao longo do processo e que há tendência de obter um modelo ótimo para o processo.

O controlador desenvolvido usando o modelo fenomenológico foi então levado à planta de bancada de tanques comunicantes. Os resultados do controlador MPC neural, sem adaptação, no teste servo são apresentados na Figura 4.8.

Assim como na simulação, o controlador sem adaptação não foi capaz de rastrear fielmente o *setpoint* no processo, com o agravante de manter o nível muito acima do

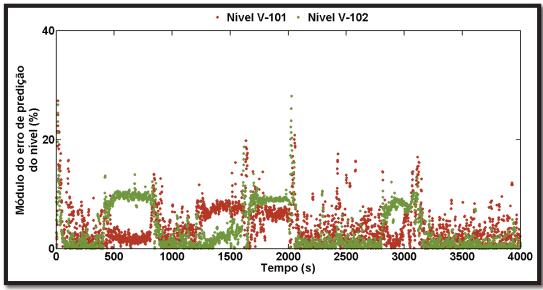
setpoint no tanque V-101 com a necessidade de intervir para evitar o transbordo no processo (entre 3000 e 3200 segundos).

Figura 4.8. Teste servo do controlador MPC Neural sem adaptação do modelo, processo real de tanques comunicantes.



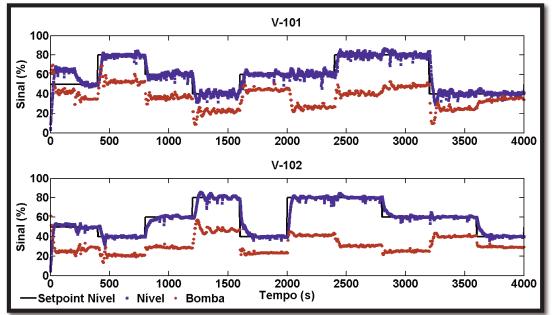
Essa dificuldade em realizar o controle é relacionada principalmente ao erro de predição da rede, apresentado na Figura 4.9.

Figura 4.9. Módulo do erro de predição da rede neural (sem adaptação) em relação à resposta da planta de tanques de nível comunicantes (processo real).



Ao repetir o experimento com o algoritmo de adaptação do modelo, a eficiência do controlador foi melhorada significativamente, como apresentado na Figura 4.10.

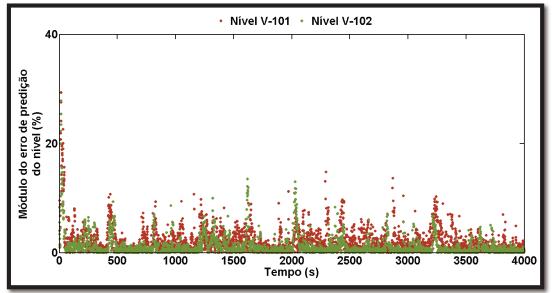
Figura 4.10. Teste servo do controlador MPC Neural com adaptação do modelo, processo real de tanques comunicantes.



Assim como na simulação, o algoritmo de adaptação da rede adicionou oscilações às variáveis controladas (nível dos tanques), porém com essa estratégia foi possível eliminar o desvio do *setpoint* em todos os casos testados, em ambos os tanques.

A Figura 4.11 apresenta o erro de predição do modelo neural durante o teste realizado no processo real com o algoritmo de adaptação do modelo. É possível observar nesse teste que o algoritmo não foi capaz de eliminar totalmente o erro de predição, diferente do que acontece no ensaio simulado. Isso se deve à presença de ruídos aleatórios no processo real e aos erros de medição dos sensores de pressão, que eventualmente retornavam valores muito abaixo das amostragens anteriores.

Figura 4.11. Módulo do erro de predição da rede neural (com adaptação) em relação à resposta da planta de tanques de nível comunicantes (processo real).



Os resultados foram quantificados a partir do Somatório do Erro Quadrático (SSE) e do Somatório da Variação das Variáveis Manipuladas (SVM) calculadas com os dados normalizados, apresentados na Tabela 4.4.

Tabela 4.4. Coeficientes de Desempenho do Controlador com e sem adaptação do modelo no processo real de níveis comunicantes.

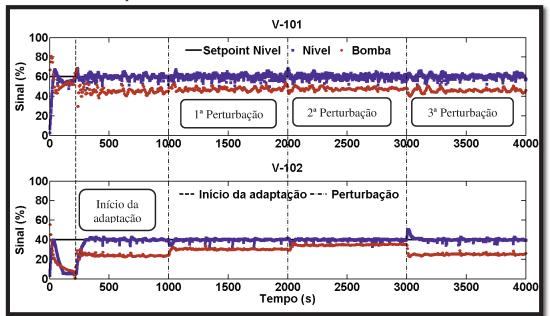
Cantualadau	SSE (V-101)	SSE (V-102)	SVM (P-101)	SVM (P-102)
Controlador	[s]	[s]	[s]	[s]
Sem Adaptação	112,3	97,0	135,6	0,45
Com Adaptação	7,8	8,6	110,9	0,26

Os resultados confirmam a observação dos gráficos em que o *SSE* foi reduzido significativamente para ambos os tanques (93% e 91%). Com relação ao esforço de controle nota-se que a adaptação reduziu a variação da variável manipulada, apesar das oscilações. Esse efeito pode ser explicado com o melhor ajuste da rede ao processo.

O controlador com modelo adaptativo foi avaliado com relação a resistência aos distúrbios não modelados e não perceptíveis ao modelo de predição. Para isso, o controlador foi submetido ao teste regulatório mantendo o *setpoint* em 60 % no tanque V-101 e 40 % no tanque V-102 e o distúrbio realizado nos instantes 1000 s (abertura da válvula HV-101), 2000 s (segunda abertura da válvula HV-101) e 3000 s (retorna a válvula à posição inicial). Os resultados são apresentados na Figura 4.12.

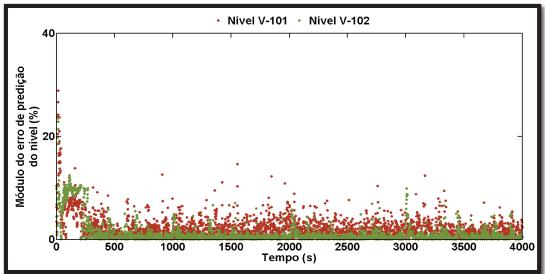
Observa-se que o controlador em seu instante inicial, antes de ativar a adaptação, não consegue atingir o *setpoint* do tanque V-102. Seu desempenho é então melhorado, assim como no teste servo, com a adaptação do modelo. Com relação ao distúrbio, observa-se que a bomba P-101 quase não sofre alteração e o nível do tanque V-101 é mantido sem sofrer interferência.

Figura 4.12. Teste Regulatório do controlador MPC Neural com adaptação do modelo, processo real de tanques comunicantes.



Por outro lado, o controlador consegue notar a variação de comportamento no tanque V-102 e atua de forma a corrigir o distúrbio. É importante destacar que com a alteração da abertura da válvula era esperado o surgimento de desvio do *setpoint*, devido ao erro de predição do modelo. Esse *offset* não foi detectado por conta da adaptação do modelo que adequou o modelo ao novo comportamento do processo, como visualizado na Figura 4.13.

Figura 4.13. Módulo do erro de predição da rede neural (com adaptação) em relação à resposta da planta de tanques de nível comunicantes (processo real) para o teste regulatório.



O teste regulatório com o controlador sem adaptação não foi possível devido ao modelo não conseguir manter o *setpoint* (60 % e 40 %) mesmo antes de realizar os distúrbios, como pode ser comprovado no resultado do teste servo (intervalo entre 1600 e 2000 s mostrado no corte do gráfico da Figura 4.8).

4.2 Planta de Controle de pH

A mesma metodologia para o desenvolvimento do controlador MPC adaptativo foi experimentada em uma planta de controle de pH. Os resultados deste estudo de caso são divididos em: modelagem e identificação; testes em malha fechada.

4.2.1 Modelagem e Identificação

Conforme discutido na Seção 3.2.1.2 a identificação do modelo das bombas (Apêndice D) foi realizada no processo real. Os parâmetros determinados juntamente com outras constantes, são apresentados na Tabela 4.5.

A modelagem do processo foi desenvolvida a partir do balanço de massa no tanque de mistura, como descrito na Seção 3.2.1.2.

Tabela 4.5. Parâmetros do Modelo.

Parâmetro	Valor
k_{P202}	38,8 l/h
$u_{0_{P202}}$	-5,26 l/h
k_{P203}	64,4 l/h
$u_{0_{P203}}$	-4,11 l/h
k_{P204}	58,8 1/h
$u_{0_{P204}}$	-11,09 l/h
S_{V201}	2,01 dm²

A determinação do tempo de amostragem foi realizada usando as simulações com o modelo fenomenológico. A Figura 4.14 apresenta os resultados do experimento simulado com degrau de tensão enviado à bomba de ácido e a Figura 4.15 o experimento simulado com degrau na tensão da bomba de base. Em ambos os casos, a simulação foi realizada partindo do tanque cheio somente com a solução tampão e mantendo o sinal enviado a bomba em 20 %. Após 2000 s foi realizada a perturbação, elevando a tensão da bomba para 30 % e, no instante de 4000 s, o sinal de tensão foi retornada à 20 %, avaliando a constante tempo como sendo o tempo necessário para que o pH variasse 63 % da diferença entre os dois regimes estacionários.

Figura 4.14. Experimento de identificação da planta controle de pH com perturbação degrau na vazão de ácido - simulação fenomenológica.

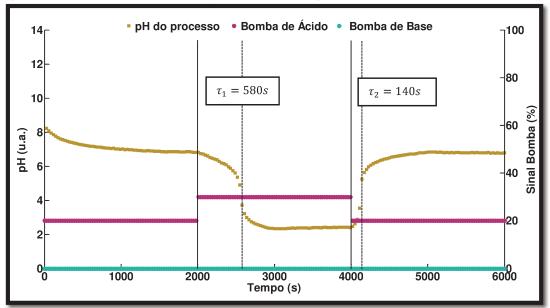
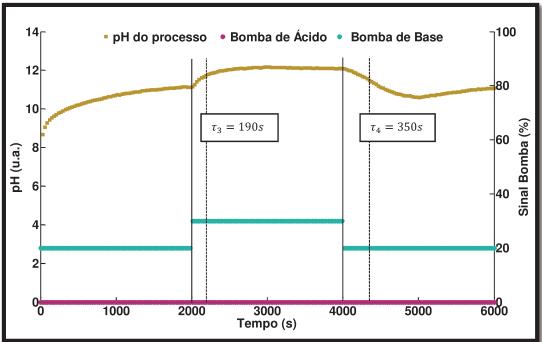


Figura 4.15. Experimento de identificação da planta controle de pH com perturbação degrau na vazão de base - simulação fenomenológica.



Os valores numéricos determinados para a constante de tempo são expostos na Tabela 4.6. Com base nos resultados, o valor médio da constante de tempo foi calculado $\tau_{médio}=315~s$. Como recomendado por STEPHANOPOULOS (1984), o intervalo de amostragem do controlador foi definido como 30 s, aproximadamente como 10 % da constante de tempo média do processo.

Tabela 4.6. Valores das constantes de tempo do processo de controle de pH para cada uma das perturbações de identificação.

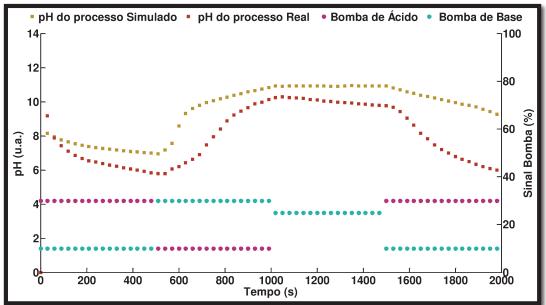
Distúrbio	2000 s	4000 s
P-201	$\tau_1 = 580 \ s$	$\tau_2 = 140 \ s$
P-202	$\tau_3 = 190 \ s$	$\tau_4 = 350 s$

Um experimento para validação do modelo foi realizado e os dados comparativos entre a resposta do modelo e os dados da planta são apresentados na Figura 4.16.

Na Figura 4.16 observa-se que a simulação apresentou a mesma tendência de comportamento que do processo real, mas existe um desvio entre os valores observados e os simulados. A esse desvio muitas hipóteses podem ser atribuídas, sendo as principais com relação ao aparato experimental: 1- Imperícia na preparação das

soluções; 2- Existência de contaminantes nas soluções, uma vez que a água utilizada provém da rede de distribuição da cidade e o bicarbonato de sódio usado é tipo comercial.

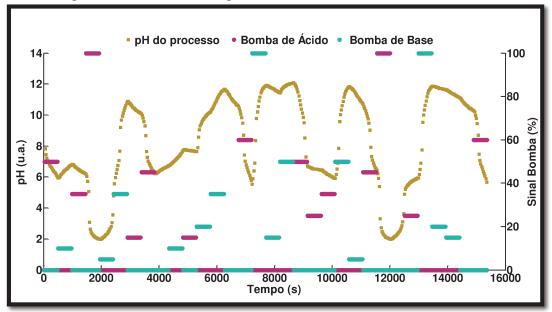
Figura 4.16. Comparação entre simulação computacional e experimento real da planta de controle de pH.



Com relação à modelagem, considerações como: 1- A hipótese de homogeneização perfeita e imediata das substâncias no reator; 2- As constantes de dissociação, obtidas da literatura, não sofreriam interferência da temperatura em que o processo estivesse ocorrendo. Mesmo observando o desvio, o modelo foi usado para treinar o modelo neural do controlador e para determinar o tempo de amostragem e os pesos da função custo do MPC.

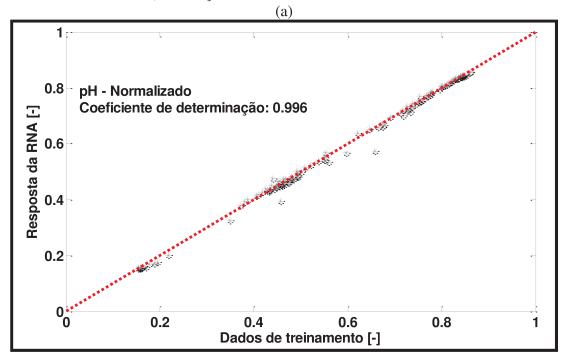
O treinamento da rede neural foi realizado conforme explicado na Seção 3.2.2, usando dados obtidos através da simulação (apresentados na Figura 4.17). Os dados foram normalizados e divididos aleatoriamente, separando 75 % para o treinamento, e os 25 % restantes foram usados para a validação da rede.

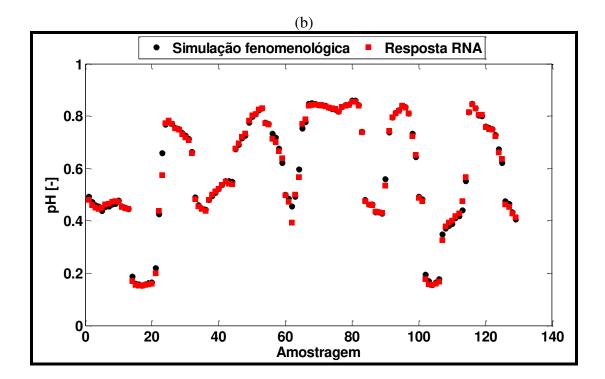
Figura 4.17. Experimento simulado *staircase* para obtenção de dados para treinamento da RNA do processo de controle de pH.



Os gráficos da Figura 4.18 apresentam a comparação da resposta da rede com os dados de treinamento e o resultado da validação da rede treinada, respectivamente. A rede resultante do treinamento *offline* tem boa aderência aos dados de treinamento e representa de forma satisfatória o processo, como visualizado na curva de validação

Figura 4.18. Resultado do treinamento da RNA. a) Comparação da resposta da rede e dados de treinamento; b) Validação da RNA.





Observa-se nos dados de treinamento que na faixa de pH entre 0,2 e 0,4 (na escala normalizada) não foram apresentados dados à rede. Nesses valores, correspondentes a 2,8 e 5,6 na escala de pH, não é possível garantir que a rede fará predições coerentes sobre o processo.

Vale destacar que o bom ajuste em relação aos dados simulados não indica que essa é a melhor rede ao processo, uma vez que, como observado na Figura 4.16, há desvios entre o modelo fenomenológico e o processo real.

4.2.2 Testes em Malha fechada

Com base nos resultados de constante de tempo do processo apresentados nas Figuras 4.14 e 4.15 e resumidos na Tabela 4.6 e na metodologia descrita na Seção 3.3, o horizonte de controle (HC) foi fixado em 3 amostragens (90 s) e horizonte de predição (HP) fixado em 5 amostragens (150 s).

A definição dos parâmetros do controlador MPC para o processo de controle de pH se deu, primeiramente, pela definição do fator de amortecimento da trajetória de referência (α). Por ser um processo de dinâmica mais lenta, optou-se por ignorar a recomendação de usar a Equação 4.1 e adotou-se o valor de 0,2. Dessa forma o controlador tem resposta mais rápida ao desvio de *setpoint*.

Os pesos de penalização da função objetivo (Equação 3.33) foram definidos, assim como no processo de tanques comunicantes, por meio de testes servos usando o

MPC sem adaptação do modelo na simulação do processo. Os resultados de SSE e SVM são mostrados na Tabela 4.7.

Tabela 4.7. Experimentos para definição dos pesos de penalização da função objetivo do processo de controle de pH.

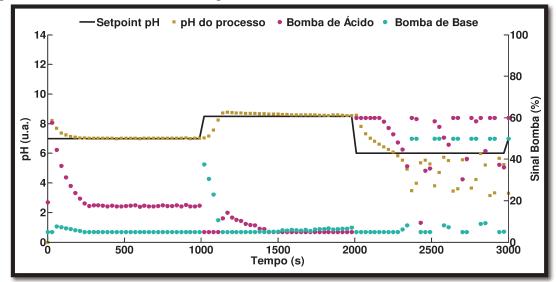
Experimentos	$P_{erro,SP}$	$P_{\Delta u}$	SSE	SVM	SVM	Soma	Soma
	1 erro,SP	¹ Δu	JJL	SV IVI	3 V IVI	SVM	Normalizada
1	1	1	173,9	0,72	4,21	4,93	1,50
2	1	10	255,4	0,33	3,51	3,84	1,55
3	1	20	314,8	0,10	2,93	3,03	1,59
4	10	1	173,5	0,79	4,36	5,15	1,55
5	10	20	175,4	0,68	4,06	4,74	1,47
6	20	1	173,4	0,80	4,37	5,17	1,55
7	20	10	173,5	0,75	4,29	5,04	1,53
Máximo			314,8			5,17	

A escolha foi realizada com base na normalização dos SSE e SVM. Foi escolhido o par de parâmetros que resultou em menor soma dos coeficientes de desempenho $(P_{\Delta u}=20~{\rm e}~P_{erro,SP}=10)$. Durante os testes a penalização da alimentação com ácido e base, simultaneamente, foi fixada igual à penalização do erro de setpoint $(P_u=P_{erroSP})$.

Usando esses parâmetros para o controlador, os resultados do experimento simulado para o controlador MPC Neural sem adaptação do modelo são apresentados na Figura 4.19.

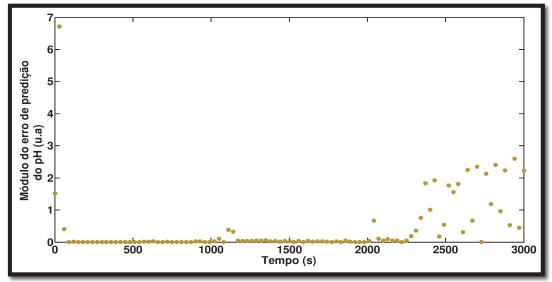
Como é possível observar, o controlador sem adaptação apresenta bom resultado no controle do pH nos *setpoint* de 7 e 8,5. Porém, quando o *setpoint* é levado à 6 ocorrem oscilações e o controlador não consegue fazer com que o nível permaneça em novo estado estacionário. Mesmo com esse resultado os parâmetros do controlador foram mantidos já que para os outros valores de *setpoint* o controle acontece de forma estável.

Figura 4.19. Teste servo do controlador MPC Neural sem adaptação do modelo, processo simulado de controle de pH.



Observando o erro de predição da rede neural (Figura 4.20) pode-se atribuir a dificuldade em realizar o controle no *setpoint* de 6 à dificuldade em predizer a dinâmica do processo nessa faixa de operação. Em outras faixas de operação (pH próximo de 7 e 8,5) a rede consegue predizer bem o processo e, portanto, o controle é realizado com sucesso.

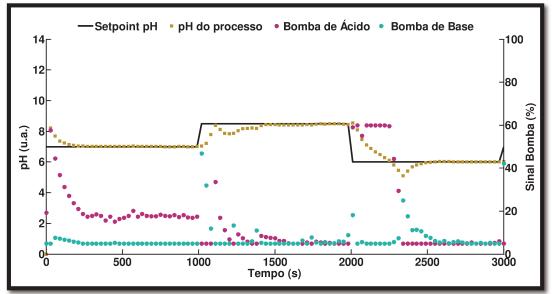
Figura 4.20. Módulo do erro de predição da rede neural (sem adaptação) em relação à resposta do modelo fenomenológico do tanque de controle de pH.



Para melhorar a predição da rede seria necessário um novo treinamento usando mais dados do processo na faixa de operação de pH próximo de 6.

Entretanto, o mesmo experimento foi realizado, apenas usando o algoritmo de adaptação do modelo neural. Os resultados são apresentados na Figura 4.21.

Figura 4.21. Teste servo do controlador MPC Neural com adaptação do modelo, processo simulado de controle de pH.

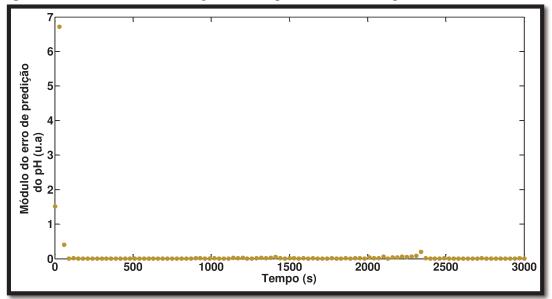


No caso do controlador com adaptação *online* do modelo é observada a extinção da oscilação das variáveis manipuladas na faixa de operação de pH = 6. Esse fato é atribuído diretamente ao aprendizado do modelo durante o processo, sendo possível confirmar a afirmação analisando o erro de predição do modelo neural (Figura 4.22).

Todavia, a alteração do modelo original causou dificuldades no início do controle no pH = 8,5 (Figura 4.21). Porém, o algoritmo de adaptação logo se adaptou à nova faixa de operação do sistema e corrigiu o modelo de forma a eliminar qualquer desvio entre variável medida e *setpoint*.

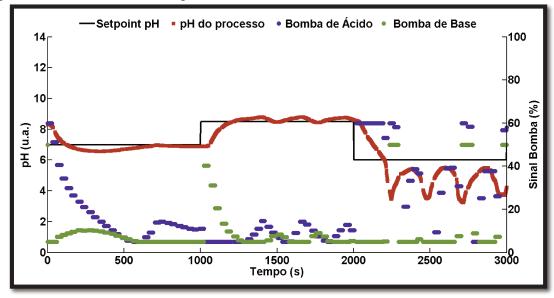
Com relação às variáveis manipuladas, os gráficos apresentam que o controlador com modelo adaptativo não aumenta muito as variações nas variáveis manipuladas. Em especial na faixa em que o controlador sem adaptação não consegue manter o *setpoint*, a adaptação reduziu a oscilação do sinal enviado às bombas.

Figura 4.22. Módulo do erro de predição da rede neural (com adaptação) em relação à resposta do modelo fenomenológico do tanque de controle de pH.



Os mesmos testes foram realizados no processo real, mantendo os parâmetros que apresentaram bons resultados na simulação. A Figura 4.23 apresenta os resultados do teste do controlador sem adaptação *online*.

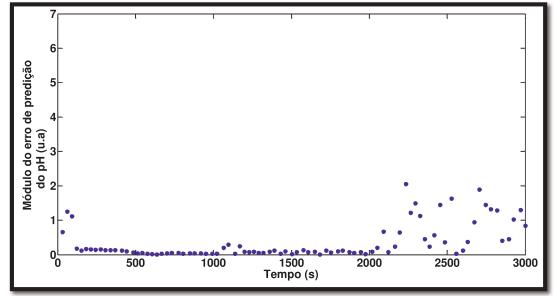
Figura 4.23. Teste servo do controlador MPC Neural sem adaptação do modelo, processo real de controle de pH.



Assim como no teste por simulação computacional o controlador sem adaptação conseguiu manter o controle nos setpoint de 7 e 8,5. Porém, no *setpoint* de pH = 6, o controlador não consegue encontrar uma solução estacionária e começa a oscilar as

variáveis manipuladas. Da mesma forma que na simulação, no processo real, o modelo neural de predição apresenta grandes erros na faixa de pH igual a 6 (Figura 4.24).

Figura 4.24. Módulo do erro de predição da rede neural (sem adaptação) em relação à resposta da planta de controle de pH (processo real).



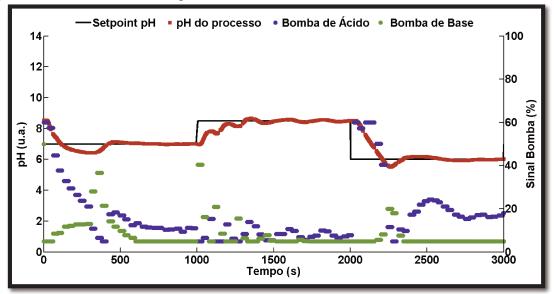
Destaca-se também o bom ajuste do modelo às outras condições testadas para o processo, resultando em um controle sem *offset*.

A Figura 4.25 apresenta os resultados do experimento servo do controlador MPC Neural com adaptação do modelo no processo real.

Mais uma vez, o resultado do processo corrobora o observado na simulação. A técnica de adaptação do modelo foi capaz de eliminar o *offset* depois de iniciada a adaptação. Além disso, fez com que o processo pudesse ser controlado no *setpoint* de pH = 6, cujo controlador com modelo original não era capaz. Assim como na simulação, houve maior dificuldade em atingir o *setpoint* de pH = 8,5 comparado com o controlador original.

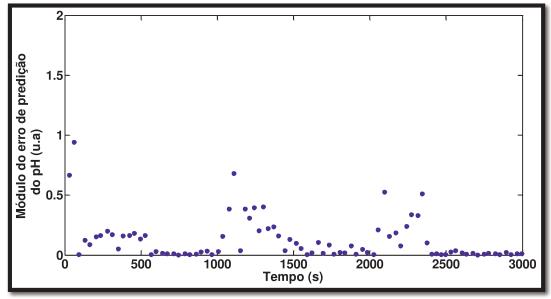
Vale ressaltar que ambos os experimentos foram iniciados com o reator cheio somente com a solução tampão e mantido o controle usando o controlador sem adaptação até o tempo de 300 s, por essa razão as curvas coincidem no início do período.

Figura 4.25. Teste servo do controlador MPC Neural com adaptação do modelo, processo real de controle de pH.



O erro de predição do modelo é apresentado na Figura 4.26. Assim como no processo simulado o algoritmo de adaptação ajustou o modelo às condições de operação, o que tornou possível o controle mesmo na faixa de pH igual a 6.

Figura 4.26. Módulo do erro de predição da rede neural (com adaptação) em relação à resposta da planta de controle de pH (processo real).



Com relação às variáveis manipuladas, é possível observar na Figura 4.25 que o controlador com correção do modelo provocou maior esforço das variáveis manipuladas, com exceção do período em que o *setpoint* foi de 6 (de 2000 a 3000 s), e o

controlador sem adaptação não consegue encontrar uma solução adequada. Nesse período, o MPC sem adaptação fez com que as bombas fossem ligadas e desligadas provocando grande oscilação do pH do processo.

O comprometimento do desempenho do controlador MPC sem a adaptação do modelo era esperada, uma vez que não havia nenhum outro mecanismo para corrigir o desvio de predição do modelo. Com os desvios de predição do modelo acentuados nas proximidades de pH igual a 6, o controlador não consegue manter o pH em um valor constante e faz com que a variável controlada oscile entre 3,6 e 5,7 (setpoint de 6).

Melhores resultados poderiam ter sido obtidos usando dados do processo real para treinamento da RNA, uso de mecanismos de correção da ação de controle do MPC ou do estado calculado pelo modelo de predição. Entretanto, o objetivo dessa pesquisa é mostrar a melhoria do controle devido à adaptação do modelo de predição por meio de um algoritmo de aprendizado em tempo real.

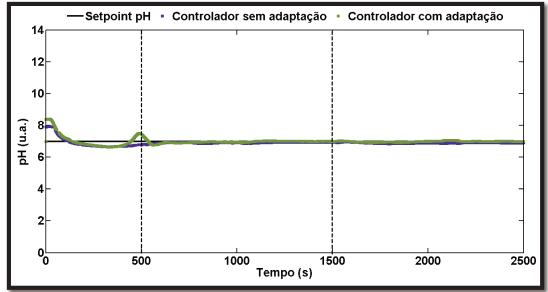
O algoritmo de adaptação da RNA baseado em FKE teve êxito realizando ajustes ao modelo e melhorando o desempenho do controlador. Esse resultado é especialmente importante uma vez que o processo real está sujeito a condições que, eventualmente, não foram levadas em consideração durante o desenvolvimento do controlador. E mesmo sem que o controlador tenha sido preparado para essas novas condições, esperase que ele seja capaz de manter o controle do processo.

Com o intuito de verificar como o controlador lida com distúrbios e com condições não consideradas durante o desenvolvimento do controlador, três testes regulatórios foram realizados: 1- Variação na vazão da solução tampão; 2- Variação na concentração da solução tampão e 3- Variação nas concentrações do ácido e da base usados no controle de pH.

Os resultados do primeiro teste regulatório com variação da vazão do tampão são apresentados na Figura 4.27.

O teste foi realizado para ambos os controladores mantendo o *setpoint* do pH em 7 e as alterações de vazão foram realizadas nos instantes 500 s e 1500 s, iniciando com o sinal enviado a bomba P-204 igual a 40 %, depois aumentando para 60 % e por fim reduzindo de volta a 40 %. É importante observar que nenhum dos controladores apresentou desvio de *setpoint* com a mudança, indicando que essa perturbação muda pouco a eficiência dos controladores mesmo que no processo de desenvolvimento não tenha sido levada em consideração.

Figura 4.27. Comparação dos resultados do teste regulatório 1 (distúrbio na vazão da solução tampão) do controlador MPC Neural com e sem adaptação do modelo, processo real de controle de pH.



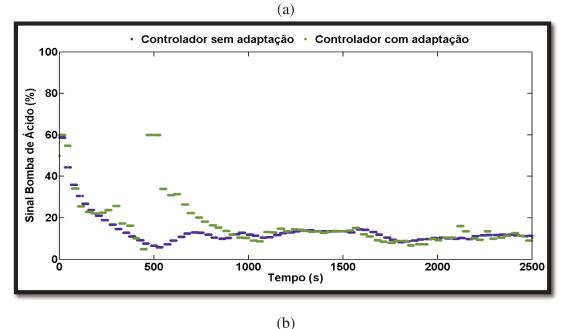
No instante de tempo igual a 500 s observa-se que o controlador com modelo adaptativo apresentou um *overshoot*, desviando-se do *setpoint*. Essa perturbação foi resultado de dois efeitos combinados: 1- O controlador havia iniciado seu processo de adaptação e por conta disso havia se adaptado às condições iniciais do processo, e a perturbação da vazão da solução tampão fez com que o modelo voltasse a não representar bem o processo; 2- Neste instante o controlador estava em processo de correção do *offset* de forma mais agressiva que o controlador sem adaptação, e com a alteração de vazão a correção calculada passou a não ser a ideal para o processo.

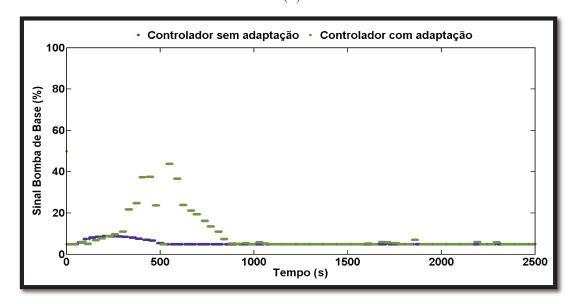
Em todo o caso, esse desvio foi logo corrigido pelo modelo e o pH de *setpoint* foi reestabelecido.

Observando as variáveis manipuladas (Figura 4.28), fica claro o esforço do controlador com modelo adaptativo no instante da primeira perturbação com a variação das variáveis manipuladas, o que ocasionou o desvio da variável controlada.

O efeito sentido no processo por conta da adição de base no instante de 500 s é semelhante ao observado no teste servo (Figura 4.25), em que observa-se o esforço do controlador em eliminar o desvio de *setpoint* existente logo após o algoritmo de adaptação do modelo ser iniciado. Entretanto, no teste regulatório houve ainda a alteração da vazão da solução tampão, provocando um somatório de efeitos que justifica o pico observado.

Figura 4.28. Comparação das variáveis manipuladas no teste regulatório 1 (distúrbio na vazão de tampão). a) Bomba de ácido; b) Bomba de base.

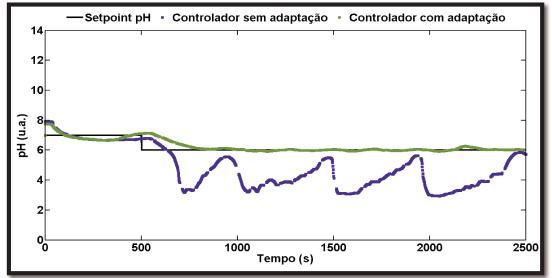




Durante o prosseguimento dos experimentos nota-se a similaridade dos dois controladores, indicando que essa perturbação regulatória não gerou grandes alterações na dinâmica do processo.

Os resultados do segundo teste regulatório (substituição do tampão por água pura) são apresentados na Figura 4.29. Esse experimento foi realizado mantendo o *setpoint* em 6 (nos 500 s iniciais manteve-se o *setpoint* em 7 para que os controladores se estabilizassem).

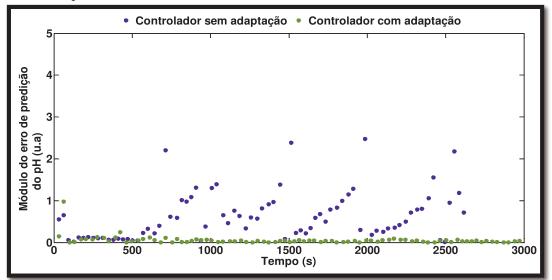
Figura 4.29. Comparação dos resultados do teste regulatório 2 (distúrbio na concentração da solução tampão) do controlador MPC Neural com e sem adaptação do modelo, processo real de controle de pH.



O teste com o controlador de modelo fixo não foi levado até o final do período por não ser capaz de encontrar uma solução e provocar oscilações nas variáveis manipuladas, como já apresentado no teste servo para esse *setpoint*.

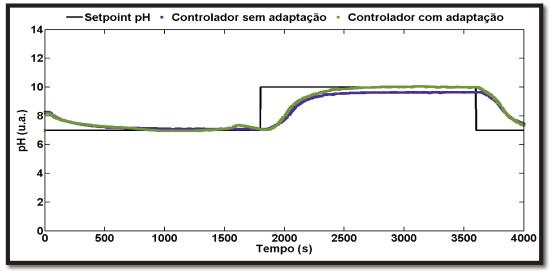
Porém, a adaptação do modelo, além permitir que o controlador encontrasse uma solução estacionária (como já observado no teste servo), conseguiu manter o *setpoint* mesmo após a substituição da solução tampão por água pura (realizada entre os instantes 1000 s e 2000 s). Os erros de predição dos modelos são apresentados na Figura 4.30.

Figura 4.30. Módulo do erro de predição da rede neural em relação à resposta da planta de controle de pH (processo real) no teste regulatório 2: a) Modelo sem adaptação; b) Modelo adaptativo.



Os resultados do terceiro teste regulatório, realizado variando as concentrações das soluções de ácido e base (concentrações sugeridas por SAFADY (2018)) são apresentados na Figura 4.31. Ambos os controladores mantiveram o *setpoint* de 7 para o pH mesmo com soluções diferentes, porém quando o *setpoint* foi alterado para 10, somente o controlador de modelo adaptativo eliminou o *offset*.

Figura 4.31. Comparação dos resultados do teste regulatório 3 (mudança das concentrações do ácido e base) do controlador MPC Neural com e sem adaptação do modelo, processo real de controle de pH.



Comparando os resultados obtidos com aqueles apresentados por SAFADY (2018) no mesmo processo, porém usando um controlador baseado na técnica de *Extremum-seeking* (Figura 4.32), o controlador MPC neural apresenta resposta mais rápida à variação de *setpoint*.

10.5 10.0 9.5 enviado a bomba 9.0 8.5 8.0 Sinal 150 200 250 Tempo (min) ----- P-202 - méd. setpoint P-202 P-203 P-203 - méd.

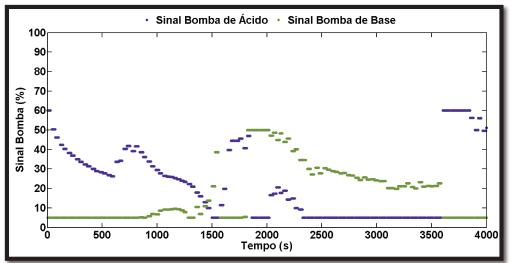
Figura 4.32. Resultados apresentados por SAFADY (2018) do controlador baseado na técnica de *Extremum-seeking* aplicado ao processo real de controle de pH.

Fonte: Adaptado de SAFADY (2018)

Comparando a constante de tempo dos controladores (tempo necessário para que a variável controlada atinja 63,2 % do novo *setpoint*) ambos os controladores são semelhantes, na Figura 4.32 o controlador de SAFADY (2018) atinge o pH de 8,9 em aproximadamente 6 minutos (360 segundos) e o MPC neural atinge o mesmo pH após 337 segundos (Figura 4.31). Entretanto, para atingir o valor do *setpoint* (pH = 10) o controlador MPC neural é mais rápido. O *setpoint* é atingido 992 segundos (cerca de 16 min) após o distúrbio, enquanto o controlador baseado em *Extremum-seeking* necessita de 75 minutos.

Outra vantagem apresentada pelo controlador MPC neural é a redução da alimentação simultânea das soluções de ácido e base (como apresentado na Figura 4.33). Como observado na Figura 4.32, o controlador de SAFADY (2018) não restringe a alimentação simultânea das duas soluções. Essa ocorrência é indesejada uma vez que as soluções produzem efeitos opostos no comportamento do pH, além de diluir o meio reacional, alterando o comportamento do processo.

Figura 4.33. Sinal enviado às bombas de ácido e base do controlador MPC neural com adaptação do modelo no experimento regulatório 3 (mudança das concentrações do ácido e base).



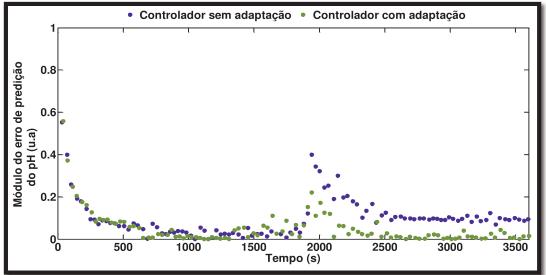
Como pode ser observado na Figura 4.33, o controlador MPC neural com adaptação do modelo evitou a alimentação simultânea com ácido e base devido à penalização adicionada à função custo (restrição flexível). Essa melhoria contribui para a maior velocidade de resposta do controlador, além de reduzir os custos de operação, por reduzir o consumo das soluções de neutralização.

Na Figura 4.34 são apresentados os módulos dos desvios do modelo de predição. Nota-se que a eliminação do *offset* do controlador de modelo adaptativo se deu à medida que o desvio do modelo foi reduzido.

O bom resultado do controlador neural sem adaptação confirma a robustez dessa técnica, desde que o modelo neural seja bem treinado e represente bem o processo, pois em quase todos os casos conseguiu lidar bem com as perturbações.

Os resultados ainda indicam que o algoritmo de adaptação do modelo conseguiu corrigir falhas no projeto do modelo de predição e ainda foi capaz de lidar com distúrbios invisíveis ao modelo.

Figura 4.34. Módulo do erro de predição da rede neural em relação à resposta da planta de controle de pH (processo real) no teste regulatório 3: a) Modelo sem adaptação; b) Modelo adaptativo.



De forma quantitativa os experimentos servo e regulatório foram analisados por meio do SSE e SVM, cujos resultados são apresentados na Tabela 4.8. Com exceção do teste regulatório 1 (variação da vazão do tampão) em que o SSE do controlador sem adaptação do modelo foi menor, o controlador com adaptação do modelo foi melhor em reduzir o desvio de *setpoint*. Esses resultados vão de acordo com os gráficos apresentados.

No caso do teste regulatório 1, o controlador sem adaptação do modelo já mantinha o pH no *setpoint* determinado, resultando no baixo SSE. Com isso, ao acionar a adaptação do modelo acrescentou-se uma incerteza quanto ao modelo resultando no desvio de setpoint (observado aos 500 s da Figura 4.27). Porém, pode ser observado nos resultados que esse desvio foi logo corrigido, retornando o pH ao *setpoint* definido para o processo.

Tabela 4.8. Índices de Desempenhos dos testes servo e regulatórios no processo real.

Experimento	Controlador	SSE (pH) [s]	SVM (P-201) [s]	SVM (P-202) [s]
Camro	Sem Adaptação	2631	6,80	5,14
Servo	Com Adaptação	940	3,06	1,60

Tabela 4.8. Índices de Desempenhos dos testes servo e regulatórios no processo real.(continuação)

Experimento	Controlador	SSE (pH) [s]	SVM (P-201) [s]	SVM (P-202) [s]
Regulatório 1 Distúrbio da vazão de	Sem Adaptação	89	0,07	0,002
Tampão	Com Adaptação	113	0,98	0,62
Regulatório 2 Distúrbio da concentração	Sem Adaptação	7897	9,03	6,58
de tampão	Com Adaptação	216	1,74	0,24
Regulatório 3 Distúrbio nas	Sem Adaptação	3726	2,43	1,56
concentrações de ácido e Base	Com Adaptação	3605	4,00	1,61

Além disso, nota-se também que a adaptação do modelo aumenta a agressividade do controlador na busca do *setpoint* aumentando a variação das variáveis manipuladas. Essa análise não pode ser levada em consideração nos experimentos em que o controlador sem adaptação do modelo não consegue encontrar uma solução para o controle, uma vez que nesses experimentos o sinal enviado às bombas oscilava.

Diante dos resultados nos testes servo e regulatório, o controlador MPC Neural com adaptação do modelo foi submetido a um experimento de longa duração com mudanças de setpoint e distúrbios na concentração e vazão da solução tampão. As condições de cada intervalo de testes são descritas na Tabela 4.9 e cada intervalo teve duração de 600 s (10 minutos).

Tabela 4.9. Condições do teste servo-regulatório de longa duração.

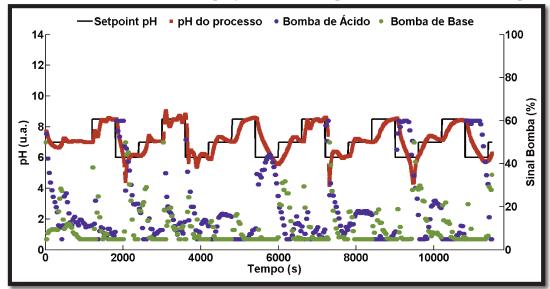
Intorvalo	Início	Setpoint	Concentração Tampão	Sinal enviado a
Intervalo [s] [-]	[-]	[mol/l]	bomba de Tampão [%]	
1	0	7	0,02	40%
2	600	7	0,02	40%
3	1200	8,5	0,02	40%
4	1800	6	0,02	40%
5	2400	7	0	60%
6	3000	8,5	0	60%
7	3600	6	0	60%
8	4200	7	0,02	60%
9	4800	8,5	0,02	60%

Tabela 4.9. Condições do teste servo-regulatório de longa duração. (continuação	Tabela 4.9.	Condições do	teste servo-regu	latório de lo	onga duração. ((continuação)
---	-------------	--------------	------------------	---------------	-----------------	---------------

Intervalo	Início	Setpoint	Concentração Tampão	Sinal enviado a
Intervalo	[s]	[-]	[mol/l]	bomba de Tampão [%]
10	5400	6	0,02	60%
11	6000	7	0	40%
12	6600	8,5	0	40%
13	7200	6	0	40%
14	7800	7	0,04	40%
15	8400	8,5	0,04	40%
16	9000	6	0,04	40%
17	9600	7	0,04	60%
18	10200	8,5	0,04	60%
19	10800	6	0,04	60%

A Figura 4.35 apresenta os resultados do experimento. O experimento repete o bom desempenho do controlador observado nos testes anteriores, pois em quase todos os casos, o controlador foi capaz de atingir e manter o *setpoint*.

Figura 4.35. Resultados do teste servo regulatório de longa duração usando o controlador MPC Neural com adaptação do modelo, processo real de controle de pH.

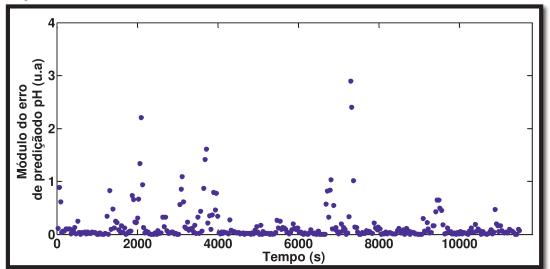


É possível notar a dificuldade do controlador em atingir e manter o *setpoint* de pH = 6 sendo observados *overshoot*, oscilações e, em alguns casos (intervalos de 5400 a 6000 s e 9000 a 9600 s), não houve tempo suficiente para observar se o *setpoint* seria mantido. Essa dificuldade observada é atribuída à dificuldade de controle do processo no *setpoint* de pH = 6 (observado com o controlador sem adaptação do modelo) e

devido ao degrau de *setpoint* de maior amplitude realizado, o que ocasionava atuação agressiva do controlador.

A dificuldade de atingir o *setpoint* é refletida na adaptação do modelo, sendo que os intervalos em que o erro do modelo é maior há maior dificuldade em atingir o valor objetivo (Figura 4.36). Observa-se que, em todo experimento, o erro do modelo é reduzido a quase zero pelo algoritmo, indicando que, mesmo com sucessivas perturbações e após um longo período de operação, o controlador continua funcionando adequadamente.

Figura 4.36. Módulo do erro de predição da rede neural (com adaptação) em relação à resposta da planta de controle de pH (processo real) em teste servo regulatório de longa duração.



5 CONCLUSÃO

As conclusões serão apresentadas seguindo a ordem de implementação do controlado MPC Neural: 1- desenvolvimento do modelo fenomenológico; 2- validação do modelo e obtenção da dinâmica do processo; 3- modelagem neural; 4- teste do controlador em simulação; 5- teste do controlador no processo real.

O modelo fenomenológico obtido para cada um dos processos foi suficiente para reproduzir o comportamento de cada um dos processos (como mostrado na Figura 4.2 e na Figura 4.16). Entretanto, existiram incertezas na obtenção de parâmetros dos modelos, provocando assim desvios entre os dados reais do processo e as respostas do modelo (como por exemplo: posição das válvulas da planta de tanques comunicantes; impurezas dissolvidas à água usada nas soluções do processo de controle de pH).

Apesar desses desvios os modelos foram usados para a obtenção do banco de dados para treinamento *offline* da rede neural (deixando a correção do modelo neural para o algoritmo de adaptação da rede) e para o ajuste de parâmetros do MPC. Além disso, os modelos fenomenológicos foram de grande importância no processo de desenvolvimento da lógica de controle.

A obtenção de dados para o treinamento da RNA, realizado em ensaios usando a modelagem fenomenológica, com perturbações do tipo *staircase*, foi suficiente para treinamento das redes, porém nota-se que em ambos os processos houve uma região com maior concentração de dados de treinamento (apresentado na Figura 4.3 e na Figura 4.18). Esse fato justifica a existência de condições no teste servo (usando o modelo fenomenológico como o processo) em que a predição da rede não era satisfatória (o controlador não conseguia manter a variável controlada em um valor fixo). O aumento do número de dados e a melhor distribuição desses dados no universo de condições poderia ter aumentado significativamente a capacidade de predição das redes.

Entretanto, optou-se em não aprofundar na melhoria do modelo neural treinado *offline* com o intuito de mostrar a capacidade do algoritmo de adaptação em suprir essas deficiências do modelo original.

A definição da arquitetura, dimensão e entradas da RNA de ambos os processos foi realizada empiricamente. A única limitação foi de manter o número de neurônios pequeno para que o algoritmo de adaptação não ficasse muito lento em decorrência do

5. Conclusão 91

grande número de parâmetros a serem ajustados. Fixar as duas redes com três neurônios apresentou resultados satisfatórios durante a modelagem, porém, há a possibilidade de melhorar a rede inicial, por exemplo melhorando a abrangência dos dados usados para treinamento.

Os testes em malha fechada usando os modelos fenomenológicos simulando os processos reais indicam a tensão que o MPC Neural tem de realizar o controle. Nas faixas em que a rede estava bem treinada, o controle era realizado com sucesso. Como haviam faixas em que a rede apresentava predições inexatas, o controlador apresentou desvio de *setpoint* e na planta de controle de pH não conseguiu manter um valor estacionário quando o *setpoint* desejado era pH = 6.

Por meio da análise dos gráficos de erro de predição da RNA, pode-se concluir que a dificuldade de controle ocorre quando o erro de predição é maior (Figura 4.5 e Figura 4.20).

Quando aplicado o algoritmo de adaptação do modelo, em ambos os processos, foi observada uma melhoria considerável em atingir e manter o *setpoint*. Mesmo no processo de controle de pH em que o controlador inicialmente não atingia um valor estacionário quando o *setpoint* de pH é 6, com a adaptação do modelo o controlador foi capaz de atingir e manter o pH igual a 6 (Figura 4.21). A melhoria da predição também pode ser observada nos gráficos de erro de predição (Figura 4.7 e Figura 4.22), em ambos os casos o erro foi reduzido consideravelmente.

Ao substituir os modelos pelos processos reais, foi observado comportamento semelhante ao que ocorreu com a simulação. O controlador de modelo fixo (não adaptativo) apresentou dificuldade em manter o *setpoint*, apresentando *offset* (principalmente no processo de tanques comunicantes) e, novamente no *setpoint* de pH = 6 na planta de controle de pH, o controlador não conseguiu manter um valor estacionário. Nos gráficos de erro de predição (Figura 4.9 e Figura 4.24) fica evidente a imprecisão do modelo neural em relação ao processo real.

O algoritmo de adaptação do modelo, assim como nos testes de simulação, foi capaz de melhorar o modelo neural, reduzindo os erros de predição (Figura 4.11 e Figura 4.26) e permitindo que o controlador atingisse o *setpoint* para os dois processos.

Todos esses resultados corroboram com a literatura, que relaciona o desempenho do MPC com o modelo de predição. Além disso, mostram que a técnica de adaptação do

5. Conclusão 92

modelo foi capaz de reduzir erros de predição e consequentemente melhorar o controlador.

O controlador MPC Neural de modelo adaptativo apresentou ainda resultados satisfatórios quando submetido a perturbações não modeladas. Nos testes regulatório e servo-regulatório apresentados, o controlador rapidamente identificou a mudança de comportamento do processo e se ajustou às novas condições mantendo o *setpoint* configurado. Esse resultado é especialmente relevante, uma vez que a literatura apresenta como uma das desvantagens do MPC a vulnerabilidade às incertezas do modelo.

6 SUGESTÕES DE TRABALHOS FUTUROS

A técnica de adaptação do modelo neural, usando um algoritmo de aprendizado em tempo real baseado no filtro de Kalman estendido se mostrou eficiente na melhoria do desempenho do MPC neural (observado nos resultados apresentados).

O controlador usado nos experimentos teve seus parâmetros ajustados de forma empírica, então para trabalhos futuros, sugere-se um estudo mais aprofundado dos efeitos desses parâmetros no desempenho do controlador e na adaptação do modelo de predição.

Sugere-se ainda, o estudo de outras técnicas de aprendizado em tempo real como a retro propagação ao longo do tempo (BPTT) ou o treinamento recorrente em tempo real (RTRL) e a adaptação dessas técnicas ao controlador MPC buscando comparar a eficiência da adaptação do filtro de Kalman estendido com outras técnicas.

É sugerido ainda o estudo dos efeitos do uso da técnica de adaptação do modelo de predição com outras técnicas de redução de desvio de *setpoint* normalmente usadas em controladores MPC, como por exemplo adicionar a ação integral ao controlador.

Com o objetivo futuro de aplicação da técnica em processos industriais, sugere-se a migração do *software* usado para desenvolver a pesquisa (*Matlab*), para uma linguagem livre, como por exemplo *Python*.

BIBLIOGRAFIA

- ABREU, R.; STICH, D.; MORALES, J. On the generalization of the Complex Step Method. **Journal of Computational and Applied Mathematics**, v. 241, p. 84-102, 15/03. 2013.
- ATKINS, P.; DE PAULA, J. **Physical chemistry for the life sciences**. Oxford University Press, USA, 2011. 300 p.
- CAMACHO, E. F.; ALBA, C. B. **Model predictive control**. Springer Science & Business Media, 2013. 280 p.
- CAO, Y. Neural Network training using the Extended Kalman Filter. p. A function using the extended Kalman filter to train MLP neural networks, 02/07/2008 2008. Disponível em: < https://www.mathworks.com/matlabcentral/fileexchange/18289-neural-network-training-using-the-extended-kalman-filter >. Acesso em: 03/05/2017.
- COSTA, T. V. D. Controle preditivo baseado em rede de modelos lineares locais aplicado a um reator de neutralização. 2010. 139 f. Dissertação (Mestrado). Faculdade de Engenharia Química, UNICAMP, Campinas.
- DULĂU, M.; DULĂU, T.-M. Multivariable System with Level Control. **Procedia Technology**, v. 22, p. 614-622, 01/01. 2016.
- FORSGREN, A.; GILL, P. E.; WRIGHT, M. H. Interior methods for nonlinear optimization. **SIAM review**, v. 44, n. 4, p. 525-597. 2002.
- GALÁN, O.; ROMAGNOLI, J. A.; PALAZOGLU, A. Real-time implementation of multi-linear model-based control strategies—an application to a bench-scale pH neutralization reactor. **Journal of Process Control**, v. 14, n. 5, p. 571-579, 01/08. 2004.
- HAYKIN, S. **Kalman filtering and neural networks**. John Wiley & Sons, 2004. 284 p.
- HAYKIN, S. S. Redes Neurais 2ed. BOOKMAN COMPANHIA ED, 2001. 842 p.
- HAYKIN, S. S. Neural networks and learning machines. New York: Prentice Hall, 2009. 906 p.
- HENSON, M. A.; SEBORG, D. E. Adaptive nonlinear control of a pH neutralization process. **IEEE Transactions on Control Systems Technology,** v. 2, n. 3, p. 169-182. 1994.
- HERMANSSON, A. W.; SYAFIIE, S. Model predictive control of pH neutralization processes: a review. **Control Engineering Practice**, v. 45, p. 98-109. 2015.
- IIGUNI, Y.; SAKAI, H.; TOKUMARU, H. A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter. **IEEE Transactions on Signal Processing,** v. 40, n. 4, p. 959-966. 1992.

- KHALID, M. U.; KADRI, M. B. Liquid level control of nonlinear Coupled Tanks System using linear model predictive control. 2012 International Conference on Emerging Technologies, 2012, 8-9 Oct. 2012. p.1-5.
- ŁAWRYŃCZUK, M. Computationally efficient model predictive control algorithms. Springer, 2014. 316 p.
- LIAN, S. T.; MARZUKI, K.; RUBIYAH, Y. Tuning of a neuro-fuzzy controller by genetic algorithms with an application to a coupled-tank liquid-level control system. **Engineering Applications of Artificial Intelligence,** v. 11, n. 4, p. 517-529, 01/08. 1998.
- LUYBEN, W. L. **Process Modeling, Simulation and Control for Chemical Engineers**. McGraw-Hill Higher Education, 1989. 725 p.
- MACIEJOWSKI, J. M. **Predictive control: with constraints**. Pearson education, 2002. 331 p.
- MORARI, M.; LEE, J. H. Model predictive control: past, present and future. **Computers & Chemical Engineering,** v. 23, n. 4-5, p. 667-682. 1999.
- NG, B. C.; DARUS, I. Z. M.; JAMALUDDIN, H.; KAMAR, H. M. Application of adaptive neural predictive control for an automotive air conditioning system. **Applied Thermal Engineering**, v. 73, n. 1, p. 1244-1254, 05/12. 2014.
- NOCEDAL, J.; WRIGHT, S. **Numerical optimization**. 2nd. New York: Springer Science & Business Media, 2006. 664 p.
- OLIVEIRA, A. F.; DE SILVA, A.; TENAN, M. A.; OLIVO, S. L. TitGer-uma planilha eletronica para simulacao de titulacao de mistura de compostos poliproticos. **Química Nova,** v. 30, n. 1, p. 224. 2007.
- PUSKORIUS, G. V.; FELDKAMP, L. A. Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. **IEEE Transactions on Neural Networks**, v. 5, n. 2, p. 279-297. 1994.
- RAMLI, M. S.; ISMAIL, R. M. T. R.; AHMAD, M. A.; NAWI, S. M. Improved coupled tank liquid levels system based on swarm adaptive tuning of hybrid proportional-integral neural network controller. **American J. of Engineering and Applied Sciences,** v. 2, n. 4, p. 669-675. 2009.
- RICE, R.; COOPER, D. J. Improve control of liquid level loops. **Chemical Engineering Progress,** v. 104, n. 6, p. 8. 2008.
- SAFADY, L. F. Controle de um reator de neutralização utilizando o método extremum-seeking. 2018. f. Dissertação (Mestrado). Faculdade de Engenharia Química, UNICAMP, Campinas, SP.

SEBORG, D. E.; MELLICHAMP, D. A.; EDGAR, T. F.; DOYLE III, F. J. **Process dynamics and control**. John Wiley & Sons, 2010. 783 p.

SENCIO, R. R. Reconfigurable Control Of Processes Subjected To Actuator Faults: A Two-Layer Mpc-Based Approach. 2017. 75 f. Dissertação (Mestre em Engenharia Química). Faculdade de Engenharia Química, UNICAMP, Campinas.

STEPHANOPOULOS, G. Chemical process control. Prentice hall New Jersey, 1984. 696 p.

WANG, L. Model predictive control system design and implementation using MATLAB®. Springer Science & Business Media, 2009. 375 p.

YU, H.; ZHANG, Z. Predictive Control Based on Neural Networks of The Chemical Process. 2006 Chinese Control Conference, 2006, 08/08. p.1143-1147.

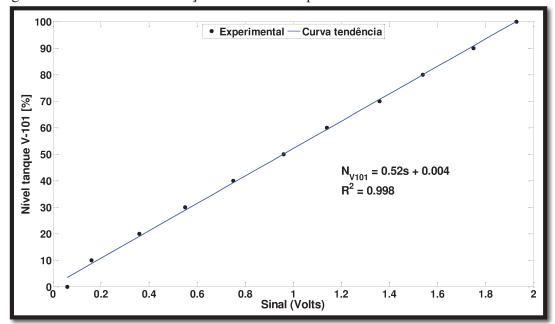
Apêndice A. Calibração dos Sensores de Nível

A calibração dos transdutores piezoelétricos foi realizada com testes em batelada, em que os tanques eram enchidos até o nível especificado e o sinal (em volts) foi aquisitado.

Tabela A.1. Dados de calibração dos transdutores piezoelétricos.

Nível	Sinal [Volts]				
[%]	$L_{V ext{-}101}$	$L_{V ext{-}102}$			
0%	0,06	0,09			
10%	0,16	0,20			
20%	0,36	0,41			
30%	0,55	0,60			
40%	0,75	0,79			
50%	0,96	0,99			
60%	1,14	1,18			
70%	1,36	1,38			
80%	1,54	1,57			
90%	1,75	1,78			
100%	1,93	1,96			

Figura A.1. Gráfico de calibração do transdutor piezoelétrico do vaso V-101.



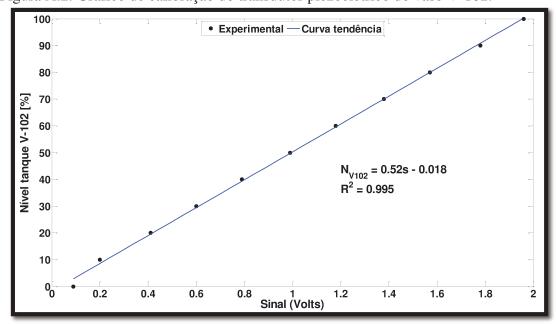


Figura A.2. Gráfico de calibração do transdutor piezoelétrico do vaso V-102.

Apêndice B. Identificação das bombas

A identificação do modelo das bombas se deu por experimentos em batelada, em que as bombas eram ligadas com o sinal indicado por um intervalo de tempo com as válvulas de drenagem dos tanques fechadas. Decorrido o tempo, o sinal dos transdutores de pressão era lido e convertido em volume (usando as dimensões dos tanques). Com o volume bombeado em um intervalo de tempo, foi mensurado a vazão média da bomba referente ao sinal enviado. Os dados foram usados para ajustar um modelo logarítmico (não linear).

Tabela B.1. Dados de identificação do modelo da bomba P-101 da planta de tanques comunicantes.

			P-101		
Tensão [Volts]	Sinal da Bomba [u.a.]	Tempo [s]	Sinal do nível [Volts]	Volume [dm ³]	Vazão [dm³/s]
1	0,2	10	0,00	0,00	0,00
1,2	0,24	30	0,41	0,29	0,01
2	0,4	20	0,85	0,59	0,03
3	0,6	20	1,33	0,92	0,05
4	0,8	15	1,27	0,88	0,06
5	1	10	0,94	0,65	0,07
5	1	15	1,43	0,99	0,07

Figura B.1. Curva de identificação do modelo da bomba P-101.

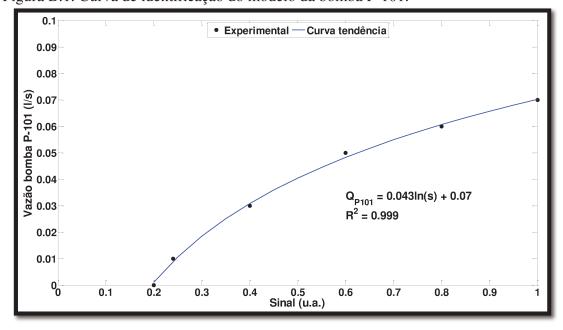
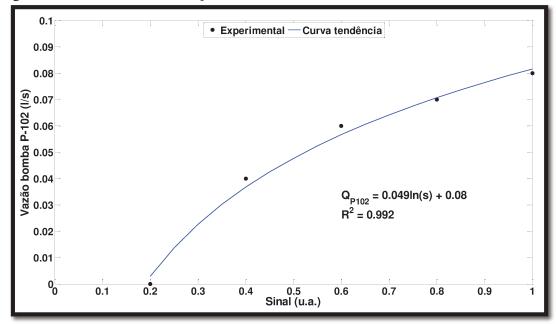


Tabela B.2. Dados de identificação do modelo da bomba P-102 da planta de tanques comunicantes.

P-102							
Tensão [Volts]	Sinal da bomba [u.a.]	Tempo [s]	Sinal do nível [Volts]	Volume [dm ³]	Vazão [dm³/s]		
1	0,2	30	0,22	0,13	0,00		
2	0,4	20	1,14	0,77	0,04		
3	0,6	15	1,34	0,91	0,06		
4	0,8	15	1,60	1,09	0,07		
5	1,0	15	1,73	1,18	0,08		
5	1,0	14	1,64	1,12	0,08		

Figura B.2. Curva de identificação do modelo da bomba P-102.



Apêndice C. Constantes das Válvulas

Com a Válvula de comunicação (HV-103) fechada, configurada a tensão fornecido às bombas e aguardado o estado estacionário, o balanço de massa em cada um dos tanques resulta em:

Tabela C.1. Dados do regime permanente obtido para cada um dos tanques (individualmente) para determinação da constante das válvulas de drenagem.

Vaso	Tensão [Volts]	Vazão [dm³/s]	Sinal Nível [u.a.]	Nível [u.a.]	Altura [dm]	Constante Válvula [dm ^{5/2} /s]
V-101	2	0,029	1,92	1,00	1,66	0,023
V-102	1,7	0,030	2,07	1,06	1,77	0,023

Com as três válvulas abertas é realizado o balanço de massa em cada um dos tanques.

$$k_{HV103} = \frac{Q_{P101} - k_{HV101} \cdot \sqrt{L_{V101}}}{\sqrt{L_{V101} - L_{V102}}}$$
 C.2

$$k_{HV103} = \frac{Q_{P102} - k_{HV102} \cdot \sqrt{L_{V102}}}{\sqrt{L_{V102} - L_{V101}}}$$
 C.3

Tabela C.2. Dados do regime permanente obtido nos experimentos de determinação da constante da válvula de comunicação.

	Q_{P101} [dm ³ /s]	Q_{P102} [dm ³ /s]	<i>N</i> _{V101} [%]	<i>N</i> _{V102} [%]	L _{V101} [dm]	L _{V102} [dm]	$k_{\rm HV103} = [{\rm dm}^{5/2}/{\rm s}]$
Exp. 1	0,052	0	92,4%	20,5%	1,543	0,343	0,022
Exp. 2	0	0,051	36,0%	93,5%	0,601	1,561	0,022

Apêndice D. Bombas planta de pH

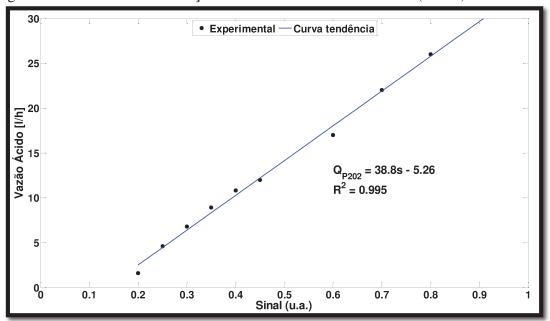
A identificação das bombas do processo de controle de pH foi realizada através dos dados de vazão retornados pelos medidores de vazão instalados em cada uma das linhas em função do sinal enviado a cada uma das bombas.

Tabela D.1. Dados de identificação das bombas P-202 (ácido), P-203 (base) e P-204

(tampão) da planta de controle de pH.

Sinal	•	Vazão [l/h]	
[u.a]	Ácido	Base	Tampão
0,2	1,6	7,8	0
0,25	4,6	11,6	0
0,3	6,8	15,3	7,1
0,35	8,9	19,1	11,1
0,4	10,8	22,2	14,1
0,45	12,0	25,5	17,0
0,6	17,0	34,8	25,0
0,7	22,0	41,0	30,1
0,8	26,0	46,7	34,0

Figura D.1. Curva de identificação do modelo da bomba de ácido (P-202).



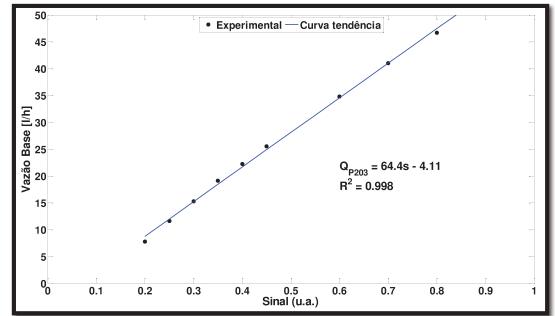
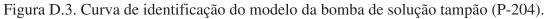


Figura D.2. Curva de identificação do modelo da bomba de base (P-203).



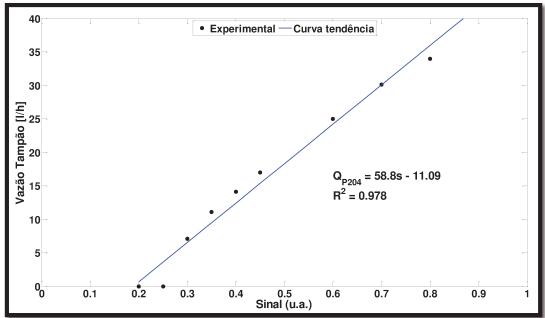


Tabela D.1. Constantes dos modelos das bombas de alimentação do processo de controle de pH de efluente.

Bombas	k_{Pi} [dm 3 /s]	$u_{0_{Pi}}\ [ext{dm}^3/ ext{s}]$
P-202 (ácido)	38,8	-5,26
P-203 (base)	64,4	-4,11
P-204 (tampão)	58,8	-11,09

Apêndice E. Artigos publicados

- 1- SENA, H. J; RAMOS, V. S.; SILVA, F. V.; FILETI, A. M. F.. Adaptive Offset Remover Based on Kalman Filter Integrated to a Model Predictive Controller. Chemical Engineering Transactions, 2017.
- 2- SENA, H. J.; SILVA, F. V.; FILETI, A. M. F.. Aplicação de Controlador Preditivo Baseado em Modelo Adaptativo em um Sistema de Tanques Comunicantes em Escala de Bancada. Blucher Chemical Engineering Proceedings, 2018.
- 3- SENA, H. J; FILETI, A. M. F.. Controlador MPC Neural Com Adaptação Online do Modelo Aplicado ao Processo de Controle de pH. In: CONGRESSO BRASILEIRO DE INSTRUMENTAÇÃO, SISTEMAS E AUTOMAÇÃO, 2019, Campinas. Anais eletrônicos. Campinas, GALOÁ, 2019.