

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

RAFAEL AKIRA AKISUE

**DESENVOLVIMENTO DE SISTEMA NEBULOSO (*FUZZY*) PARA
CONTROLE DO OXIGÊNIO DISSOLVIDO NO CULTIVO DE
Escherichia coli PARA EXPRESSÃO DE PROTEÍNAS
RECOMBINANTES**

São Carlos
2018

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

RAFAEL AKIRA AKISUE

**DESENVOLVIMENTO DE SISTEMA NEBULOSO (*FUZZY*) PARA
CONTROLE DO OXIGÊNIO DISSOLVIDO NO CULTIVO DE
Escherichia coli PARA EXPRESSÃO DE PROTEÍNAS
RECOMBINANTES**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Química da Universidade Federal de São Carlos, como parte dos requisitos necessários à obtenção do título de mestre em Engenharia Química, área de concentração em Pesquisa e Desenvolvimento de Processos Químicos

Orientador: Prof. Dr. Ruy de Sousa Jr.

São Carlos
2018

DEDICATÓRIA

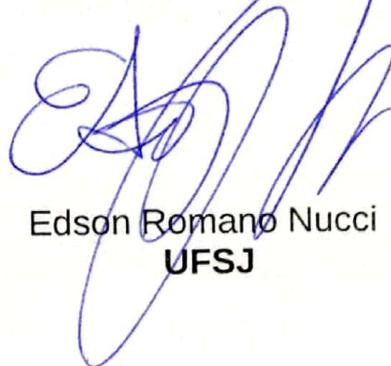
Dedico este trabalho a meus pais Lauro e Maura. Sem o apoio incondicional de vocês jamais chegaria tão longe. Se hoje posso sonhar alto, é porque tive alguém que me deu asas. Este vôo também é de vocês.

MEMBROS DA BANCA EXAMINADORA DA DEFESA DE DISSERTAÇÃO DE RAFAEL AKIRA AKISUE APRESENTADA AO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA DA UNIVERSIDADE FEDERAL DE SÃO CARLOS, EM 26 DE FEVEREIRO DE 2018.

BANCA EXAMINADORA:



Ruy de Sousa Júnior
Orientador, UFSCar



Edson Romano Nucci
UFSJ



Antonio Carlos Luperni Horta
UFSCar

AGRADECIMENTOS

A meus amados pais, Lauro e Maura, meus modelos de integridade, moralidade e conduta. Sua coragem, força e fé foram meus alicerces durante esta jornada.

Ao Professor Dr. Ruy de Sousa Júnior, para quem não há agradecimentos que bastem, pela orientação e amizade durante estes anos juntos.

Ao Professor Dr. Antonio Carlos Luperni Horta que gentilmente cedeu os dados de cultivo de *E.coli* e forneceu valorosas contribuições ao trabalho.

Ao meu companheiro Wagner Santiago cujas palavras de carinho e apoio incondicional foram fundamentais nesta jornada.

Aos amigos Ana Elisa Achilles, Ana Elisa Lista, Andreza Longati, Ariane Sbrice, Fernando de Mello, Camila Lacerda, Juliana Silva, Fabiana Massini, Lais Federici, Nathália Mello e Paula Satie, Rauber Pereira, Ricardo Arbach, pelas boas recordações vividas.

A todos os integrantes do laboratório KYATERA que me receberam durante o mestrado.

À minha família.

Aos demais amigos que estiveram comigo durante esta jornada.

Ao CNPq pelo financiamento.

“Neste mundo não existem coincidências, apenas o inevitável”

(YUUKO ICHIHARA)

RESUMO

Um bioprocesso de extrema importância é o cultivo da bactéria *Escherichia coli* recombinante para expressão de proteínas heterólogas. Para isso, o cultivo celular de alta densidade é uma das técnicas mais utilizadas. Embora resulte em densidades celulares acima de 100g/L, também está sujeita a desafios. Assim pesquisadores do Laboratório de Desenvolvimento e Automação de Bioprocessos (LaDABio) do Departamento de Engenharia Química da Universidade Federal de São Carlos (UFSCar) desenvolveram um programa de computador robusto (SUPERSYS_HCDC) que além de outras funções, dispõe de um sistema híbrido de controle com um controlador Proporcional, Integral e Derivativo para manipulação da velocidade de agitação e uma árvore de decisão para manipulação das vazões de ar e oxigênio que controlam a porcentagem de oxigênio dissolvido (atualmente alguns controladores comerciais também oferecem este controle em cascata). No entanto, em particular, podem ocorrer atrasos nos dispositivos responsáveis pela injeção de ar e oxigênio no biorreator, uma vez que a árvore de decisão não fornece respostas suaves (ou seja, não há transições graduais na ação de controle). O sistema opera introduzindo degraus nas vazões de ar e oxigênio. Sob a luz dos fatos acima mencionados, a lógica *fuzzy* foi utilizada para desenvolver um controlador nebuloso com o objetivo de melhorar o controle do oxigênio dissolvido no cultivo de *E. coli* para expressão de proteínas recombinantes. Primeiramente foi utilizado o *Fuzzy Logic Toolbox* para gerar o algoritmo de controle implementado via código em MATLAB. Em seguida os parâmetros das funções de pertinência foram otimizados utilizando a ferramenta ANFIS. Por fim, para a realização de testes do controlador nebuloso, o mesmo foi acoplado a um modelo neural do processo. Este foi criado utilizando o *toolbox* de redes neurais artificiais e dados de cultivo de *E. coli*. Utilizando o controle *fuzzy* foi possível manter a porcentagem de oxigênio dissolvido próximo ao valor de *setpoint* de 30%. Em geral as respostas ao controlador nebuloso foram mais suaves em relação às respostas fornecidas pela árvore de decisão, de modo que os picos de oxigênio dissolvido foram suavizados.

Palavras-chave: Lógica *Fuzzy*. Oxigênio Dissolvido. Redes Neurais Artificiais. *Escherichia coli* recombinante. SUPERSYS_HCDC

ABSTRACT

One very important bioprocess is the cultivation of recombinant *E. coli* for expression of heterologous proteins. For this, High Cell Density Culture is one of the most widely used technique. Although it results in cell densities above 100 g/L, it also has its challenges. Therefore, researchers from the Laboratory of Development and Automation of Bioprocesses (LaDABio) at Chemical Engineering Department of Federal University of São Carlos (UFSCar) developed a robust computer program (SUPERSYS_HCDC) that, among other functions, presents a hybrid system with a Proportional-Integral-Derivative (PID) controller for agitation speed control and a decision tree to manipulate air and oxygen flow rates that control the percentage of dissolved oxygen in the cultivation (nowadays some commercial controllers also offer this cascade control). However, in particular, delays may occur in the devices responsible for air and oxygen injection in the bioreactor, since the decision tree provides no smooth responses (that is, no gradual transitions in the control action). The system presented operates by introducing steps in the air and oxygen flow rates. Under the light of the above-mentioned facts, fuzzy reasoning was used to develop a fuzzy controller, aiming to improve dissolved oxygen control in recombinant *E. coli* cultivation for heterologous protein production. At first, fuzzy logic toolbox was used to generate a control algorithm implemented in a MATLAB code. Secondly, the membership functions parameters were optimized using ANFIS tool. Finally, in order to perform tests using the fuzzy controller, it was coupled to a neural network model of the process. This was created using artificial neural network toolbox and *E. coli* cultivation data. Results for oxygen and air flow rates indicated that the trends of aeration required by *E. coli* cultivation were fulfilled. Using fuzzy controller, it was possible to maintain the percentage of dissolved oxygen around the setpoint value of 30%. In general, the fuzzy controller responses were smoother than those provided by the decision tree, in a way that the dissolved oxygen peaks were softened.

Keywords: Fuzzy Logic. Dissolved Oxygen. Artificial Neural Networks. Recombinant *Escherichia coli*. SUPERSYS_HCDC

LISTA DE FIGURAS

Figura 1 - Variável linguística temperatura	35
Figura 2 - Estrutura básica de um controlador nebuloso	37
Figura 3 - Representação da função f e seu Fuzzy Graph.....	39
Figura 4 - Processo de avaliação de um sistema nebuloso	40
Figura 5 - Processo de inferência nebuloso	40
Figura 6 - Gráfico da vazão de oxigênio (l/min) em função do tempo (h).....	51
Figura 7 - Gráfico da vazão de ar (l/min) em função do tempo (h).....	51
Figura 8 - Principais componentes de um neurônio biológico	52
Figura 9 - Estrutura básica de um neurônio artificial	54
Figura 10 - Três camadas de neurônios - configuração (3-2-2).....	55
Figura 11 - Rede Feedforward de uma única camada.....	56
Figura 12 - Rede Feedforward com múltiplas camadas.....	57
Figura 13 - Redes recorrentes	57
Figura 14 - Interface do Fuzzy logic Designer (FIS editor).....	66
Figura 15 - Interface do Membership Function Editor	67
Figura 16 - Interface do Rule editor.....	67
Figura 17 - Interface do Rule Viewer	68
Figura 18 - Interface do Surface Viewer.....	69
Figura 19 - ANFIS editor	71
Figura 20 - Interface inicial (esquerda) e escolha dos dados de entrada e saída do Time series app (direita).....	73
Figura 21 - Da esquerda para a direita tem-se a interface de escolha da divisão dos dados; escolha dos parâmetros da arquitetura da rede e por fim um breve resumo da rede.	74
Figura 22 - Da esquerda para a direita tem-se a interface de escolha do método de treinamento da rede; resultados do treinamento da rede e por fim a escolha dos gráficos que podem ser plotados para auxiliar na avaliação da rede treinada.....	75
Figura 23 - Interface de ajuste da rede neural treinada.....	75
Figura 24 - Interface para exportar e implementar a rede neural treinada.....	76
Figura 25 - Velocidade específica de crescimento experimental e a suavizada em função do tempo.....	83
Figura 26 - Diagrama do simulador pretendido.....	85
Figura 27 - Resultado do ajuste das funções de pertinência de entrada do sistema nebuloso Delta realizado pelo ANFIS.....	92
Figura 28 - Ajuste das funções de pertinência de entrada do sistema de inferência nebuloso "Princ1" realizada pelo ANFIS.....	94
Figura 29 - Ajuste das funções de pertinência de entrada do sistema de inferência nebuloso Princ2 realizada pelo ANFIS.	97
Figura 30 - Vazão de ar (l/min) em função do tempo (h).....	100
Figura 31 - Vazão de oxigênio (l/min) em função do tempo (h).....	100
Figura 32 - Vazão de ar (l/min) em função do tempo (h) utilizando o simulador simples (primeiro conjunto de dados).....	102
Figura 33 - Vazão de oxigênio (l/min) em função do tempo (h) utilizando o simulador simples (primeiro conjunto de dados).....	102
Figura 34 - Comparação entre as porcentagens de oxigênio dissolvido simulado e experimental utilizando o simulador simples (primeiro conjunto de dados).....	103
Figura 35 - Vazão de ar (l/min) em função do tempo (h) utilizando o simulador avançado (primeiro conjunto de dados).....	104

Figura 36 - Vazão de oxigênio (l/min) em função do tempo (h) utilizando o simulador avançado (primeiro conjunto de dados).....	105
Figura 37 - Comparação entre as porcentagens de oxigênio dissolvido simulado e experimental utilizando o simulador avançado (primeiro conjunto de dados).....	106
Figura 38 - Vazão de ar (l/min) em função do tempo (h) utilizando o simulador simples (segundo conjunto de dados).....	107
Figura 39 - Vazão de oxigênio (l/min) em função do tempo (h) utilizando o simulador simples (segundo conjunto de dados).....	108
Figura 40 - Comparação entre as porcentagens de oxigênio dissolvido simulado e experimental utilizando o simulador simples (segundo conjunto de dados).....	109
Figura 41 - Vazão de ar (l/min) em função do tempo (h) utilizando o simulador avançado (segundo conjunto de dados).....	110
Figura 42 - Vazão de oxigênio (l/min) em função do tempo (h) utilizando o simulador avançado (segundo conjunto de dados).....	110
Figura 43 - Comparação entre as porcentagens de oxigênio dissolvido simulado e experimental utilizando o simulador avançado (segundo conjunto de dados).....	112
Figura 44 - Dados experimentais do cultivo A16 (controlado pela árvore de decisão).....	128
Figura 45 - Dados experimentais do cultivo A12 (controlado pela árvore de decisão).....	129

LISTA DE QUADROS

Quadro 1 - Pseudocódigo de controle de vazões de ar e oxigênio.....	32
Quadro 2 - Modus ponens generalizado	39
Quadro 3 - Variáveis da sub-rotina “Delta”	49
Quadro 4 - Variáveis da sub-rotina “Princ”	49
Quadro 5 - Valores linguísticos das variáveis da sub-rotina “Delta”	49
Quadro 6 - Valores linguísticos das variáveis da sub-rotina “Princ”	49
Quadro 7 - Regras para a sub-rotina "Delta"	50
Quadro 8 - Regras para a sub-rotina "Princ"	50
Quadro 9 - Variáveis de entrada e saída das novas sub-rotinas e seus respectivos valores linguísticos.	80
Quadro 10 - Faixas de QarEscolha e QO ₂ Escolha na rotina em MATLAB.....	81
Quadro 11 - Excerto da rotina em MATLAB para o cálculo de QarSe.....	84
Quadro 12 – Chamada à função correspondente à rede 1.....	89
Quadro 13 - Chamada à função correspondente à rede 2.	90
Quadro 14 - Excerto da programação para o cálculo da velocidade específica de crescimento	91
Quadro 15 - Faixas de valores ótimos para QarEscolha e QO ₂ Escolha utilizados no simulador simples	99
Quadro 16 - Faixas de valores ótimos para QarEscolha e QO ₂ Escolha utilizados no simulador simples	99
Quadro 17 - Faixas de valores ótimos para QarEscolha e QO ₂ Escolha utilizados no simulador simples	103
Quadro 18 - Faixas de valores ótimos para QarEscolha e QO ₂ Escolha utilizando o simulador avançado.	105
Quadro 19 - Faixas de valores ótimos para QarEscolha e QO ₂ Escolha utilizados no simulador simples	108
Quadro 20 - Faixas de valores ótimos para QarEscolha e QO ₂ Escolha utilizados no simulador avançado.	111

LISTA DE TABELAS

Tabela 1: Valores de R para diferentes arquiteturas de rede para treinamento da Rede 1 (A16)	87
Tabela 2: Valores de R para diferentes arquiteturas de rede para treinamento da Rede 2 (A16)	87
Tabela 3: Valores de R para diferentes arquiteturas de rede para treinamento da Rede 1 (A12)	88
Tabela 4: Valores de R para diferentes arquiteturas de rede para treinamento da Rede 2 (A12)	89
Tabela 5 - Valores dos parâmetros das funções de pertinência de entrada do sistema de inferência nebuloso Delta ajustados pelo ANFIS.	93
Tabela 6 - Valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Delta ajustados pelo ANFIS.	93
Tabela 7 - Valores dos parâmetros das funções de pertinência de entrada do sistema de inferência nebuloso Princ1 ajustados pelo ANFIS.	95
Tabela 8 - Valores dos parâmetros das funções de pertinência de entrada do sistema de inferência nebuloso Princ2 ajustados pelo ANFIS.	98
Tabela 9 - Resultados do treinamento da rede neural (Rede 1) para o primeiro conjunto de dados	101
Tabela 10 - Resultados do treinamento da rede neural (Rede 2) para o primeiro conjunto de dados	104
Tabela 11 - Resultados do treinamento da rede neural (Rede 1) para o segundo conjunto de dados	107
Tabela 12 - Resultados do treinamento da rede neural (Rede 2) para o segundo conjunto de dados	109
Tabela 13 - Valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Princ1 ajustados pelo ANFIS.	124
Tabela 14 - Valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Princ2 ajustados pelo ANFIS.	126

LISTA DE SIGLAS

CDC: *Center for Disease Control and Prevention*

PCV: *Pneumococcal Conjugate Vaccine*

PS: Polissacarídeo Capsular

HCDC: High Cell Density Cultivation

CADC: Cultivo e Alta Densidade Celular

LaDABio: Laboratório de Desenvolvimento e Automação de Bioprocessos

PID: Proporcional Integral e Derivativo

DNA: Ácido Desoxirribonucleico

RNA: Ácido Ribonucleico

TCA: *Tricarboxylic acid*

NADH: Nicotinamida Adenina Dinucleotídeo (Reduzida)

FIS: Fuzzy Inference System

ANFIS: Adaptive Neuro Fuzzy Inference System

Trimf: *triangular membership function*

Trapmf: *trapezoidal membership function*

Gaussmf: *Gauss membership function*

Gbellmf: *G-bell membership function*

Sigmf: *sigmoidal membership function*

Zmf: *Z-polynomial membership function*

NAR: *Nonlinear Autoregressive*

NARX: *Nonlinear Autoregressive with Exogenous Input*

PGA: Penicilina G-Acilase

BP: Baixo e Positivo

MP: Médio e Positivo

AP: Alto e Positivo

MAP: Muito Alto e Positivo

R: Regressão

LS_{agit}: Limite Superior de Agitação

LI_{OD}: Limite Inferior de Oxigênio Dissolvido

RQ: *Respiratory Quotient*

PC: *Personal Computer*

SCADA: *Supervisory Control and Data Acquisition*

RTS: *Resistance Temperature Sensor*

DLL: *Dynamic Link Library*

LISTA DE ABREVIATURAS

E. coli: *Escherichia coli*

S. pneumoniae: *Streptococcus pneumoniae*

B. megaterium: *Bacillus megaterium*

C. acremonium: *Cephalosporium acremonium*

Soft Sensor: *Software Sensor*

CX: Concentração Celular (g/l)

OD: Porcentagem de saturação de oxigênio dissolvido (%)

QarP: Vazão de ar (l/min)

QO₂P: Vazão de oxigênio (l/min)

Q_{máx}: Vazão máxima de gás permitida no sistema (l/min)

Qarant: Vazão de ar anterior (l/min)

QO₂ant: Vazão de oxigênio anterior (l/min)

QarEscolha: Variável auxiliar de saída da subrotina Princ1

QO₂Escolha: Variável auxiliar de saída da subrotina Princ2

QarSe: Variável auxiliar de entrada das subrotinas Princ1 e Princ2

AgitMed: Velocidade de Agitação (rpm)

LISTA DE SÍMBOLOS

F: Vazão de alimentação do meio suplementar	(l/h)
CO ₂ : Dióxido de carbono	
μ : Velocidade específica de crescimento	(h ⁻¹)
$\mu_{m\acute{a}x}$: Velocidade específica de crescimento máxima	(h ⁻¹)
μ_{crit} : Velocidade específica de crescimento crítica	(h ⁻¹)
Y _{XC} : Coeficiente de Rendimento	(g _x /g _s)
m: Manutenção	(g _x /(g _s h))
V ₀ : Volume no início da alimentação	(l)
C _{S0} : Concentração inicial de substrato no meio de alimentação	(g/l)
C _{SR} : Concentração de substrato residual	(g/l)
C _{X0} : Concentração celular inicial	(g/l)
t _{alimentação} : Tempo de alimentação	(h)
Kc: Ganho proporcional do controlador PID	
Ti: Ganho Integral do controlador PID	
Td: Ganho Derivativo do controlador PID	
W _{ij} : Vetor de pesos sinápticos	
X _i : Vetor de entrada da rede neural	
b: <i>bias</i>	
S _j : Vetor de saída da rede neural	
α : Grau de suavização	
A _i : Conjunto nebuloso A	
B _i : Conjunto nebuloso B	
ΔO_2 : Variação da vazão de ar e de oxigênio	(l/min)

SUMÁRIO

RESUMO	7
ABSTRACT	8
1. INTRODUÇÃO	19
2. REVISÃO BIBLIOGRÁFICA	23
2.1. <i>Escherichia coli</i>	23
2.2. Cultivos de Alta Densidade Celular (CADC)	24
2.2.1. Utilização dos cultivos de alta densidade celular para produção de <i>E. coli</i>	24
2.2.2. Fatores que afetam o crescimento em CADC	25
2.2.3. Estratégias para os cultivos de alta densidade celular	26
2.3. Monitoramento e controle de CADC	27
2.3.1. SUPERSYS_HCDC	28
2.3.2. Módulo de controle de fases por sistema de inteligência artificial	28
2.3.3. Módulo do controle automático da vazão de alimentação através da velocidade específica de crescimento dinâmica	29
2.3.4. Módulo de alarmes	30
2.3.5. Módulo de controle da concentração do oxigênio dissolvido	30
2.4. Fuzzy	33
2.4.1. Fuzzy: Fundamentos	33
2.4.2. Controle Nebuloso	37
2.4.3. Aplicações do controle nebuloso em bioprocessos	43
2.5. Redes Neurais	52
2.5.1. Redes Neurais Biológicas	52
2.5.2. Redes Neurais Artificiais	54
2.5.3. Aplicações de redes neurais em bioprocessos	61
2.6. Ferramentas computacionais	65
2.6.1. Fuzzy Inference System (FIS)	65
2.6.2. Adaptive Neuro Fuzzy Inference System (ANFIS)	69
2.6.3. Neural Network Time Series	71
3. METODOLOGIA	77
3.1. Dados experimentais	77
3.1.1. Microrganismo	77
3.1.2. Cultivo	77
3.2. Desenvolvimento do controlador nebuloso	78
3.2.1. Adaptações do sistema de inferência nebuloso	79
3.2.2. Inserção dos dados na ferramenta ANFIS	80
3.3. Simulador	85
3.3.1. Planejamento do simulador	85
3.3.2. Treinamento das redes neurais	86
3.3.3. Adaptação da rotina em MATLAB	89
4. RESULTADOS E DISCUSSÕES	92
4.1. Resultados com controle nebuloso otimizado via ANFIS	92
4.1.1. Otimização dos parâmetros do sistema de inferência nebuloso “Delta”	92
4.1.2. Otimização dos parâmetros dos sistemas de inferência nebulosos “Princ1” e “Princ2”	94
4.1.3. Resultados utilizando o controle nebuloso otimizado via ANFIS	98
4.2. Resultados com o simulador	101
4.2.1. Resultados utilizando o simulador simples	101
4.2.2. Resultados utilizando o simulador avançado	104
4.3. Resultados de teste de robustez do simulador	106
4.3.1. Resultados utilizando o simulador simples	107

4.3.2. Resultados utilizando o simulador avançado.....	109
5. CONCLUSÕES.....	113
6. SUGESTÕES PARA TRABALHOS FUTUROS	114
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	115
APÊNDICE.....	119
ANEXO	128

1. INTRODUÇÃO

De acordo com o Centro de Controle e Prevenção de Doenças dos Estados Unidos (ou do inglês, *Centers for Disease Control and Prevention - CDC*) o *Streptococcus pneumoniae* ou pneumococo é uma bactéria gram negativa e importante patógeno humano, responsável por diversas infecções. Elas podem ser caracterizadas em dois grupos principais:

- Doenças não invasivas: otite média aguda (infecção nos ouvidos), sinusite, conjuntivite, bronquite;
- Doenças invasivas: pneumonia (infecção nos pulmões), meningite (infecção nas membranas que revestem o cérebro e a medula espinhal), bacteremia (infecção na corrente sanguínea);

As doenças invasivas atingem majoritariamente crianças menores de 5 anos e idosos, maiores de 65 anos. De acordo com os dados do CDC, em 1998 foi estimado que a cada grupo de 100.000 crianças menores de 5 anos, cerca de 100 contraíram alguma destas doenças. Com a introdução de vacinas como a PCV7 e a PCV13 (2000 e 2010, respectivamente), esta estatística caiu para 9 a cada 100.000 crianças (2015). Já para o grupo dos idosos acima de 65 anos, em 1998 eram 59 casos de doenças invasivas para cada 100.000 pessoas. Em 2015 foram 26 casos para cada 100.000 pessoas (CDC, 2016).

De acordo com dados do Ministério da saúde, em 2016 foram reportados 881 casos confirmados de meningite pneumocócica no Brasil, sendo que a taxa de mortalidade foi de 29,4%, a mais alta dentre os tipos que foram incluídos no estudo (PORTAL DA SAÚDE, 2016).

Assim, o impacto destas doenças, aliada ao aumento da resistência bacteriana aos antibióticos, faz com que seja prioritário o desenvolvimento de vacinas.

O principal fator de virulência desta bactéria está na sua cápsula polissacarídica, que impede ou retarda a fagocitose. Assim foi criada uma vacina de polissacarídeo capsular (PS), baseada na ideia de que anticorpos formados contra o PS são efetivos na imunização. Porém observou-se que os anticorpos formados eram extremamente específicos a cada sorotipo, sendo contabilizados mais de 90 tipos. Além disso, o polissacarídeo capsular não possui a capacidade de ativar os linfócitos T, responsáveis pela memória imunológica. Uma solução encontrada foi a conjugação do polissacarídeo com uma proteína, que aumenta a resposta imune (gerando memória imunológica). Ela estimula a produção de anticorpos que se ligam ao polissacarídeo da bactéria, estimulando a opsonização, a fagocitose e a morte do pneumococo (GONÇALVES, 2002; HORTA, 2008).

Genética e fisiologia bem caracterizadas e tecnologia de recombinação genética sedimentada, foram alguns dos fatores que fizeram a bactéria *Escherichia coli* pioneira na área de produção de proteínas heterólogas. Desde então vários tipos de proteínas simples tais como hormônios de crescimento e albumina humana, são produzidos industrialmente. Ao serem produzidas, as proteínas se acumulam no meio intracelular da *E. coli* recombinante. Logo, a produtividade é proporcional à densidade celular final e à produtividade específica (quantidade de produto formado por unidade mássica celular por unidade de tempo) (LEE, 1996).

Uma das técnicas mais utilizadas para o cultivo de *E. coli* é o cultivo de alta densidade celular (CADC ou, do inglês, *High Cell Density Cultivation - HCDC*), cujo objetivo é a maximização da produção. Entretanto, a execução de um CADC é um desafio até hoje, pois variáveis como concentração de oxigênio dissolvido e fornecimento de substrato são difíceis de controlar. Na etapa de batelada alimentada é crítico manter um controle preciso da vazão de alimentação do meio suplementar, a fim de evitar a produção de metabólitos indesejáveis. Além disso, inibição pelo substrato, capacidade de transferência de oxigênio limitada e dissipação de calor limitada são alguns problemas enfrentados na fase de formação da biomassa e expressão de proteínas. Para solucionar tais problemas, pesquisadores do grupo LaDABio do Departamento de Engenharia Química da Universidade Federal de São Carlos (DEQ/UFSCar) desenvolveram o SUPERSYS_HCDC, que além de outras funções, controla a vazão de alimentação do meio suplementar. Ele demonstrou sua eficiência ao ser utilizado, com sucesso, em diversos cultivos celulares. Em sua versão atual o SUPERSYS_HCDC é capaz de conduzir uma batelada alimentada de *E. coli* recombinante até altas concentrações celulares (superiores a 100g/l). Toda a aquisição e controle foram desenvolvidos fazendo uso da plataforma LabView 8.0 (*National Instruments*) e do software MATLAB 6.5 (*MathWorks*). Além do controle preciso da vazão de alimentação do meio suplementar, a manutenção de níveis adequados da concentração de oxigênio dissolvido no meio de cultura é fundamental para o bom andamento do cultivo. Para isso o programa dispõe de um sistema híbrido de controle com um PID para a manipulação da velocidade de agitação e uma árvore de decisão para a manipulação das vazões de ar e oxigênio que controlam a porcentagem de oxigênio dissolvido. No entanto, em particular, podem ocorrer atrasos nos dispositivos responsáveis pela injeção de ar e oxigênio no biorreator, uma vez que a árvore de decisão não fornece respostas suaves (ou seja, não há transições graduais na ação de controle). O sistema original opera introduzindo degraus nas vazões de ar e oxigênio. Surge então a necessidade de melhoria deste sistema de controle automático das vazões de ar e oxigênio (HORTA, 2011).

Por muito tempo o controle e a otimização de processos biológicos foram baseados em modelos matemáticos fenomenológicos ligados à perspectiva de controle tradicional. Porém, devido a fatores como a complexidade do mecanismo de reação intracelular e mudanças nas características do microrganismo ao longo da reação (dinâmica não linear), o desenvolvimento de um modelo matemático não é trivial. Uma alternativa é a utilização da metodologia nebulosa (*fuzzy*), que integra a corrente das Metodologias Baseadas no Conhecimento. Ela é um formalismo matemático capaz de capturar informações vagas, descritas em uma linguagem natural e convertê-las para um formato numérico, de fácil manipulação pelos computadores (HORTA, 2011; SOUSA, 1999; TAKEMURA, 2016; KONSTANTINOV; YOSHIDA, 1992).

A teoria nebulosa é baseada nos conjuntos nebulosos (*fuzzy sets*) introduzidos pelo Professor Lotfali A. Zadeh em 1965. Ele observou que certos conjuntos, definidos até então pela lógica ordinária - em que um elemento pode pertencer ou não a certo conjunto (ou classe) - eram extremamente imprecisos e ainda assim tinham papel fundamental nas comunicações, abstrações e reconhecimento de padrões. Assim, ele estabeleceu que, em um conjunto nebuloso, seria permitido ao elemento um determinado grau de pertinência em relação ao conjunto. O grau de pertinência é um valor numérico entre 0 e 1 (em que 0 significa completa exclusão, e 1 completa pertinência) que traduz a possibilidade de um certo elemento de pertencer ou não a um determinado conjunto. Quando se considera todos os elementos de um conjunto, tem-se as funções de pertinência, através das quais obtém-se o grau de pertinência de cada elemento. Para fins de projeto, a escolha adequada dos parâmetros e das formas das funções de pertinência é essencial para a formulação das regras nebulosas (do tipo “Se - Então”). Além disso é necessário escolher corretamente as variáveis nebulosas a serem utilizadas (GOMIDE; GUDWIN, 1994; SOUSA, 1999).

Porém, quando não é possível discernir a forma e os parâmetros das funções de pertinência, simplesmente observando os dados, utiliza-se a ferramenta ANFIS (*Adaptive Neuro Fuzzy Inference System*), pertencente ao *Fuzzy Logic Toolbox*. Ela realiza esta escolha automaticamente, utilizando um conjunto de funções, a fim de construir um sistema de inferência nebuloso a partir de dados de entrada e saída (MATHWORKS, 2016).

Por fim, para testar o controlador nebuloso criado, foram utilizadas redes neurais artificiais para modelar o processo de cultivo de *E. coli* para produção de proteínas recombinantes. São muitos os exemplos na literatura aplicando com sucesso as redes neurais para modelagem de processos biológicos complexos (Glassey *et al* (1994), Silva *et al* (2008) e Ahmad *et al* (2015)). As redes neurais mimetizam a capacidade do cérebro humano de executar tarefas de grande complexidade, não lineares e com processamento paralelo da informação.

Podem ser definidas como uma rede de múltiplos processadores simples (cada um deles possuindo uma certa quantidade de memória) que são conectados através de meios de comunicação (conexões), aos quais são associados valores (pesos) numéricos (PALMA NETO; NICOLETTI, 2005).

Assim, este trabalho teve como objetivo o desenvolvimento de um sistema nebuloso para controle do oxigênio dissolvido no cultivo de *E. coli* para expressão de proteínas recombinantes. Para isso, foram cumpridas as seguintes etapas:

- A. A *fuzzyficação* do pseudocódigo de controle existente, fazendo uso de dados de controle de cultivo disponíveis.
- B. Modelagem dinâmica do processo de produção por redes neurais artificiais.
- C. Criação de um simulador que integra o controlador nebuloso otimizado (via ferramenta ANFIS) e o modelo neural do processo a fim de testar o sistema de controle *fuzzy*.

2. REVISÃO BIBLIOGRÁFICA

Nesta seção serão elucidados os principais conceitos utilizados no desenvolvimento desta dissertação. Serão abordados conceitos gerais sobre o microrganismo em estudo, pressupostos teóricos da lógica nebulosa e das redes neurais artificiais e por fim um resumo das ferramentas computacionais a serem utilizadas.

2.1. *Escherichia coli*

De acordo com o Centro de Controle e Prevenção de Doenças dos Estados Unidos a *Escherichia coli* é uma bactéria gram negativa que se encontra no trato gastrointestinal de seres endotérmicos (seres humanos e animais). Em geral ela é inofensiva e tem um papel fundamental na manutenção do bom funcionamento do sistema digestivo. Porém, algumas espécies são patogênicas, ou seja, podem causar doenças ao homem, tais como diarreia, pneumonia e infecções no trato urinário (CDC, 2015). Desde 1950 a *E. coli* vem sendo extensamente estudada, sendo este o organismo modelo procarioto mais bem documentado (TORTORA, 2012). Devido à caracterização detalhada de sua genética e fisiologia, e à sedimentação da tecnologia de recombinação genética ao longo dos anos, a bactéria *E. coli* foi escolhida como pioneira na área de produção de proteínas heterólogas (RIESENBERG; GUTHKE, 1999; SHILOACH; FASS, 2005). Até hoje ela é um dos principais hospedeiros para estudos envolvendo tecnologia de recombinação genética. Em 1978, Arthur Riggs, Keiichi Itakura, Herbert Boyer e demais colaboradores sintetizaram a primeira insulina de origem não animal (insulina sintética ou insulina humana), utilizando para tal a *E. coli* modificada. Desde então vários tipos de proteínas simples tais como hormônios de crescimento e albumina humana são produzidos industrialmente (GENENTECH, 2016; RIGGS *et al.*, 1979).

Ao serem produzidas, as proteínas se acumulam no meio intracelular da *E. coli* recombinante. Logo, a produtividade é proporcional à densidade celular final e à produtividade específica (quantidade de produto formado por unidade mássica celular por unidade de tempo) (LEE, 1996). As estratégias de pesquisa para produção da *E. coli* baseiam-se em duas vertentes principais, uma que foca no desenvolvimento de técnicas de cultivo e outra que foca nas técnicas de manipulação das características fisiológicas da bactéria (SHILOACH; FASS, 2005). Uma das técnicas mais utilizadas para o cultivo de *E. coli* é o Cultivo de Alta Densidade Celular (CADC, do inglês, *High Cell Density Cultivation - HCDC*), cujo objetivo é a maximização da produção.

2.2. Cultivos de Alta Densidade Celular (CADC)

Nesta seção serão apresentadas as principais aplicações e os fatores que afetam o crescimento celular em cultivos de alta densidade celular.

2.2.1. Utilização dos cultivos de alta densidade celular para produção de *E. coli*

Os produtos comerciais produzidos pela *E. coli* se dividem em duas categorias: produtos de baixo-volume-alto-valor (proteínas recombinantes) e produtos de alto-volume-baixo-valor (ácido succínico). Seja para condições de viabilidade econômica (primeiro caso), ou para redução de custos e enxugamento do processo produtivo (segundo caso), para ambas as categorias, é necessária uma alta densidade celular final. Assim, a técnica de cultivo de alta densidade celular tem sido estudada desde 1970 (SHILOACH; FASS, 2005). Porém não há um consenso de qual seria a concentração celular final mínima para se caracterizar um cultivo de alta densidade celular (HORTA, 2011).

Mori e colaboradores (1979) analisaram a relação entre a densidade celular e a viscosidade do caldo de cultura. Eles observaram que para valores acima de 200g/l a viscosidade do caldo aumenta exponencialmente com o aumento da concentração celular. Este praticamente perdia sua fluidez em concentrações celulares acima de 220g/l, inviabilizando o cultivo em reatores (MORI *et al.*, 1979).

Märkl e colaboradores (1992) estimaram a concentração celular máxima que poderia se atingir utilizando cultivos de alta densidade celular. Chegaram a um valor entre 160 e 200g/l (massa celular seca), resultado bastante próximo ao de Mori e colaboradores (1979). Para corroborar seus cálculos, os próprios cientistas conseguiram chegar a uma densidade celular de 174 g/l, utilizando para isso um reator de diálise bicameral com membrana. A fonte de carbono escolhida foi o glicerol (meio definido) e a alimentação era constante (MÄRKEL *et al.*, 1992).

2.2.2. Fatores que afetam o crescimento em CADC

A. Produção de acetato

Em cultivos aeróbios o excesso de glicose pode aumentar a produção de intermediários ácidos, como por exemplo o acetato. Essa acidogênese é conhecida como efeito *Crabtree*. A formação do acetato é causada por um desbalanceamento entre o metabolismo da glicose e a respiração (o fluxo de carbono para a célula excede a capacidade de síntese da mesma), causando a saturação do ciclo de Krebs. Outros fatores que intensificam esse processo são o excesso de Dinucleótido de nicotinamida e adenina em sua forma reduzida (NADH) e a repressão de enzimas do ciclo de Krebs (Isocitrato Desidrogenase e Alfa Cetoglutarato Desidrogenase) (LULI e STROHL, 1990 ; Kopot, 2015).

Documentou-se diversos efeitos nocivos do acúmulo de acetato no microrganismo. Em concentrações acima de 2 g/l observa-se uma redução no crescimento da *E. coli*, queda na produção de biomassa e na densidade máxima de células que pode ser atingida (SHILOACH; FASS, 2005). O acetato tem efeitos mais expressivos em células recombinantes do que em não recombinantes. Isto se deve ao efeito repressor do acetato na síntese de DNA, RNA, proteínas e lipídios (LEE, 1996).

Uma solução encontrada para inibir a formação de acetato é a alteração da composição do meio. Utilizando-se glicerol ao invés de glicose como fonte de carbono reduz-se o fluxo de carbono pela via glicolítica, evitando a saturação do Ciclo de Krebs (LEE, 1996).

B. Meio de Cultura

Estudos mostram que a densidade celular final não é diretamente proporcional à quantidade de substrato inoculado. Isso se deve à inibição pelo substrato, principalmente das fontes de carbono e nitrogênio. Logo, uma estratégia é começar o cultivo apenas com as quantidades mínimas de cada nutriente e adicioná-los conforme a demanda para o crescimento celular. É desejável também manter a solução de alimentação a mais simples possível (LEE, 1996).

Em geral, utiliza-se glicerol como substituto da glicose. Isto se deve à presença de transportadores de glicerol na membrana, garantindo uma alta velocidade de absorção do mesmo. Além disso, o glicerol não tem efeito repressor na expressão de proteínas, tal como exercido pela glicose (HORTA, 2011).

C. Temperatura

A temperatura é um importante fator de controle do metabolismo celular. Reduzindo-se a temperatura de 37°C para uma faixa de 26 a 30°C reduz-se o consumo de nutrientes e a taxa de crescimento. Isto afeta diretamente a produção de intermediários tóxicos e diminui o calor gerado pelo metabolismo celular. Nessa faixa de temperatura a demanda por oxigênio é menor, o que pode evitar a utilização de oxigênio puro (LEE, 1996).

D. Oxigênio

Devido à sua baixa solubilidade em água, o oxigênio é outro fator limitante em cultivos de alta densidade celular. Em cultivos aeróbios é essencial a manutenção do abastecimento de oxigênio. Isto pode ser realizado aumentando-se a vazão de alimentação de ar ou a frequência de agitação do biorreator e/ou aumentando a pressão. Também é possível injetar oxigênio puro no reator (LEE, 1996).

2.2.3. Estratégias para os cultivos de alta densidade celular

Em cultivos descontínuos alimentados é crítico o controle da velocidade específica de crescimento uma vez que esta está diretamente relacionada com a formação de subprodutos inibitórios e a produtividade celular. A fim de obter altos níveis de expressão de proteínas recombinantes, três fases caracterizam a estratégia de produção de proteínas heterólogas utilizando cultivos em batelada alimentada:

- a) Cultivo descontínuo ou batelada: até que a fonte de carbono seja consumida, com μ atingindo $\mu_{m\acute{a}x}$.
- b) Alimentação da fonte de carbono de forma controlada a fim de evitar o acúmulo de acetato, com $\mu < \mu_{crit}$.
- c) Indução da síntese de proteína recombinante, com $\mu \ll \mu_{m\acute{a}x}$ (*Stress metabólico*).

Na primeira fase há o consumo de toda a fonte de carbono no meio, de forma que o microrganismo atinge a velocidade específica de crescimento máxima $\mu_{m\acute{a}x}$. Esta fase de

crescimento ilimitado é necessária para permitir a ambientação das células no biorreator e posteriormente sua estabilização antes de iniciar a fase caracterizada pela escassez de substrato.

Tal como elucidado na seção anterior, a formação de acetato durante o cultivo é umas das principais preocupações ao utilizar cultivos descontínuos alimentados. Assim, inicia-se a segunda fase com a alimentação da fonte de carbono de forma controlada, para que não haja saturação do ciclo de Krebs e, conseqüentemente, a escolha pela via fermentativa (efeito *Crabtree*) com produção de acetato. Nesta fase a velocidade de crescimento é mantida abaixo de um valor crítico (μ_{crit}), ou seja, a vazão de alimentação do meio deve ser mantida acima de um valor necessário para a manutenção do crescimento celular e abaixo de um limiar, em que verifica-se o acúmulo de acetato. Riesenber (1991) reportou que, para cultivos de alta densidade celular de *E. coli*, a formação de acetato é inibida com $\mu_{crit} < 0,35h^{-1}$ (meio definido) e $\mu_{crit} < 0,20h^{-1}$ (meio complexo). A manutenção da velocidade específica de crescimento abaixo de um valor crítico (para a formação de acetato) permite ao cultivo crescer conforme o limite imposto por outro parâmetro, no caso a fonte de carbono.

Após a obtenção de altas concentrações celulares, inicia-se a terceira fase, em que ocorre a indução da síntese de proteína recombinante. Células em cultivos de alta densidade celulares estão sujeitas a *stress* devido à falta de nutrientes e à indução da síntese de proteínas. Nesta fase toda a fonte de carbono é utilizada apenas para a manutenção e a produção da proteína heteróloga. Logo, a velocidade específica de crescimento é mantida abaixo da velocidade máxima (RIESENBERG, 1991; SHILOACH e FASS, 2005; BABAEIPOUR *et al.*, 2006; FAULKNER *et al.*, 2006; CARNEIRO; FERREIRA; ROCHA, 2013).

2.3. Monitoramento e controle de CADDC

Além do controle preciso da vazão de alimentação, outros fatores podem influenciar o sucesso e o bom andamento dos cultivos de alta densidade celular. Tal como elucidado em seções anteriores, tem-se a produção de metabólitos indesejados, alteração na temperatura, variações na concentração de oxigênio dissolvido, formação de espuma, entre outros. Assim, é necessário um operador experiente e treinado para supervisionar o cultivo durante todo o tempo. Porém cultivos celulares podem levar mais de 30 horas. Logo, uma ferramenta baseada em inteligência artificial poderia auxiliar e até substituir a supervisão humana. Desde a época da concepção do SUPERSYS_HCDC programas de monitoramento e controle de biorreatores já eram vendidos por fabricantes como Fermentec e Marconi. Todavia, ressalta-se que nenhum

deles tinha a capacidade de conduzir um cultivo em batelada alimentada automaticamente até altas densidades (HORTA, 2011).

Todos apresentam as mesmas funcionalidades básicas: controle de pH, oxigênio dissolvido, temperatura, agitação e vazão de líquidos e gases. Com o avanço das telecomunicações, também foram incluídas funções de acesso remoto pela rede, alarmes sonoros e por mensagens (SMS) (HORTA, 2011).

2.3.1. SUPERSYS_HCDC

Um dos sistemas de automação para cultivos de alta densidade celular criados no Brasil é o SUPERSYS_HCDC. Ele foi desenvolvido no Laboratório de Desenvolvimento e Automação de Bioprocessos (LaDABio) do Departamento de Engenharia Química da Universidade Federal de São Carlos. Já foi empregado com sucesso em cultivos de diversos microrganismos, tais como *Saccharomyces cerevisiae*, *Pichia pastoris*, *Bacillus megaterium* e *Escherichia coli* recombinante. O programa foi desenvolvido em LABVIEW e MATLAB (versões 8.0 e 6.5 respectivamente) e dispõe de um controle híbrido hierárquico automático da concentração de oxigênio dissolvido, atuando na manipulação da velocidade de agitação e na manipulação das vazões de ar e oxigênio. Ele possui também um comitê de redes neurais responsáveis por identificar o fim da batelada e iniciar a vazão de alimentação do meio suplementar, além dos demais controles, como o de pH e o de temperatura (HORTA, 2011).

Utilizando o sistema com todas as suas funcionalidades chegou-se a uma concentração celular máxima de $155 \text{ g (massa seca)l}^{-1}$ em apenas 17 horas de cultivo. Ou seja, obteve-se uma produtividade de biomassa superior a $9 \text{ g (massa seca)l}^{-1}\text{h}^{-1}$. Além disso houve significativa diminuição dos problemas operacionais e erros de operação (HORTA, 2011).

2.3.2. Módulo de controle de fases por sistema de inteligência artificial

O SUPERSYS_HCDC possui módulos que atuam desde as funções básicas até a inferência de importantes variáveis em tempo real. Um desses módulos é o controle da vazão de alimentação do meio suplementar.

Três módulos de controle especiais compõem o controle da vazão de alimentação do meio suplementar.

Num primeiro nível, o controle do acionamento automático da bomba está vinculado ao reconhecimento da fase do cultivo. Este é feito por três algoritmos: comitê de redes neurais, lógica nebulosa e derivada do CO₂. As três técnicas foram implementadas (de forma redundante) para evitar falsos positivos.

O comitê de redes neurais foi escolhido como algoritmo principal de reconhecimento de fases. A arquitetura da rede adotada foi: 9 neurônios na camada de entrada (velocidade de agitação, vazões de ar e oxigênio, oxigênio dissolvido, permissividade, condutividade, fração molar de CO₂ nos gases de exaustão, dentre outras), 10 neurônios na camada oculta e um neurônio na camada de saída. Foram realizados três cultivos em batelada alimentada para aquisição de dados e treinamento das redes neurais. Cada cultivo realizado foi utilizado para treinar uma rede neural. A resposta do comitê foi feita pela média da saída de classificação de cada rede neural.

Para acompanhar o comitê de redes neurais, foi utilizada a lógica nebulosa. Foram criadas três variáveis nebulosas de entrada: tempo, fração molar de CO₂ e derivada da fração molar de CO₂, cujos valores linguísticos foram: Tempo Ótimo (TO), Tempo Cedo (TC) e Tempo Tarde (TT), Fração molar Média (CM), Fração molar Baixa (CB) e Fração molar Alta (CA) e por fim Derivada Negativa da fração molar de CO₂ (DCN) e Derivada Positiva da fração molar de CO₂ (DCZP). Para a única variável de saída foram adotados os seguintes valores linguísticos: Fase de batelada (NP) e início da alimentação (PC).

Por fim, se a derivada da fração molar de oxigênio dissolvido fosse menor que -2 a bomba de alimentação seria acionada.

A decisão final de ativar ou não a bomba de alimentação foi realizada através da média entre os três métodos utilizados.

2.3.3. Módulo do controle automático da vazão de alimentação através da velocidade específica de crescimento dinâmica.

O módulo de controle automático da vazão de alimentação pode ser desmembrado em três partes principais. Num primeiro nível, o controle da vazão de alimentação F é feito por uma árvore de decisão simples. Caso a concentração de oxigênio dissolvido estiver próxima ao seu valor no *set point*, a vazão de alimentação do meio suplementar é calculada pela Equação 1:

$$F = \left(\frac{\mu}{Y_{XC}} + m \right) \frac{C_{X0}V_0}{C_{S0} - C_{SR}} e^{(\mu_{SET}t_{alim})} \quad (1)$$

Em que μ é a velocidade específica (h^{-1}), F é a vazão de alimentação do meio fresco (l/h), Y_{XC} é o coeficiente de rendimento (gx/gs), m é a manutenção (gx/gs.h), C_{X0} é a concentração celular no início da alimentação (g/l), V_0 é o volume no início da alimentação (l), t_{alim} representa o tempo de alimentação (h), C_{S0} a concentração de substrato no meio de alimentação (g/l) e C_{SR} a concentração de substrato residual (g/l) (HORTA, 2011).

Caso contrário a vazão de alimentação é calculada pela Equação 2

$$F_t = 0,9 * F_{t-1} \quad (2)$$

2.3.4. Módulo de alarmes

Um recurso importante implementado no SUPERSYS_HCDC foi um conjunto de alarmes para detecção de problemas durante o cultivo. Foram monitorados: pH (mal funcionamento das bombas de ácido ou base), temperatura (aumento ou diminuição acentuada da temperatura do banho), concentração de oxigênio dissolvido (aumento da vazão de gás ou da velocidade de agitação), vazão de ar e oxigênio (aumento da pressão nas linhas de ar e oxigênio), pressão (aumento da pressão no reator) e viabilidade do microrganismo (μ negativo ou μ positivo) (HORTA, 2011).

2.3.5. Módulo de controle da concentração do oxigênio dissolvido

Outro módulo principal é o controle da concentração de oxigênio dissolvido. A fim de compatibilizar a alta velocidade de consumo de oxigênio de *E. coli* em CADC, o controle da concentração de oxigênio dissolvido foi feito utilizando um sistema híbrido. Este combina um controlador PID digital (utilizando um algoritmo de velocidade), atuando diretamente na velocidade de agitação, com um algoritmo heurístico que modulava as vazões de ar e oxigênio (HORTA, 2011).

A estratégia de controle PID para a velocidade de agitação é descrita pela Equação 3:

$$Agit_t = Agit_{t-1} + K_c \{ [(SetP - OD_t) - (SetP - OD_{t-1})] + \frac{\Delta t}{T_i} (SetP - OD_t) + \frac{T_d}{\Delta t} [(SetP - OD_t) - 2(SetP - OD_{t-1}) + (SetP - OD_{t-2})] \} \quad (3)$$

Em que Agit é a velocidade de agitação (rpm), K_c é a constant proporcional, T_d é a constante derivativa, T_i é a constante integral, SetP é o *set point* da concentração de oxigênio dissolvido, OD é a concentração de oxigênio dissolvido e t é o tempo (s).

Esta estratégia de controle PID digital advém da equação do PID clássico mostrado na Equação 4.

$$p_k = \bar{p} + K_c[\varepsilon_t + \frac{1}{T_i} \int_0^t \varepsilon(t) dt + T_d \frac{d\varepsilon(t)}{dt}] \quad (4)$$

Em que p_k é o valor da variável de saída do controlador e ε é o valor do erro.

Para obtenção da equação do PID digital, basta substituir os termos integral e derivativo da Equação 4 por suas respectivas aproximações por diferenças finitas, dadas pelas Equações 5 e 6 (SEBORG, EDGAR e MELLICHAMP, 2004).

$$\int_0^t \varepsilon(t) dt \cong \sum_{j=1}^k \varepsilon_j \Delta t \quad (5)$$

$$\frac{d\varepsilon(t)}{dt} \cong \frac{\varepsilon_k - \varepsilon_{k-1}}{\Delta t} \quad (6)$$

Obtendo-se a Equação 7 do controlador PID digital (de posição).

$$p_k = \bar{p} + K_c[\varepsilon_t + \frac{\Delta t}{T_i} \sum_{j=1}^k \varepsilon_j + \frac{T_d}{\Delta t} (\varepsilon_k - \varepsilon_{k-1})] \quad (7)$$

Para se obter a Equação 7 na forma de velocidade, calcula-se a variação da variável de saída do controlador PID, tal como mostra a Equação 8, a partir dos termos p_k e p_{k-1} (SEBORG, EDGAR e MELLICHAMP, 2004).

$$\Delta p_k = p_k - p_{k-1} = K_c[\varepsilon_k - \varepsilon_{k-1} + \frac{\Delta t}{T_i} * \varepsilon_k + \frac{T_d}{\Delta t} (\varepsilon_k - 2 * \varepsilon_{k-1} + \varepsilon_{k-2})] \quad (8)$$

O controle das vazões de ar e oxigênio é feito por um algoritmo independente baseado em uma heurística. O código para a realização deste controle consiste de uma árvore de decisão apresentada no Quadro 1 (HORTA, 2011).

Quadro 1 - Pseudocódigo de controle de vazões de ar e oxigênio

<p><i>Início</i></p> <p><i>Se</i> $Agit_{média} < 0,70 * LS_{Agit}$ e $Q_{O_2 anterior} > 0$</p> $Q_{Ar} = Q_{Ar} + \Delta O_2^-$ $Q_{O_2} = Q_{O_2 anterior} - \Delta O_2^-$ <p><i>Se</i> $Agit_{média} \geq 0,99 * LS_{Agit}$ e $OD < LI_{OD}$</p> <p><i>Se</i> $Q_{Ar} \geq (Q_{max} - Q_{O_2 anterior})$</p> $Q_{Ar} = Q_{Ar} - \Delta O_2^+$ $Q_{O_2} = Q_{O_2} + \Delta O_2^+$ <p><i>Senão</i></p> $Q_{Ar} = Q_{Ar} + \Delta O_2^+$ $Q_{O_2} = Q_{O_2 anterior}$ <p><i>Fim</i></p>
--

Fonte: HORTA, 2011

No Quadro 1, $Agit_{média}$ é a velocidade de agitação média (com controle PID - Proporcional, Integral Derivativo), LS_{Agit} é o limite superior de agitação, Q são as vazões de ar ou oxigênio, ΔO_2^- e ΔO_2^+ são valores constantes e LI_{OD} é o limite inferior de oxigênio dissolvido.

O pseudocódigo é interpretado da seguinte forma. Caso a velocidade de agitação esteja baixa (inferior a 70% do limite superior de agitação) em um cenário em que já esteja sendo injetado oxigênio no sistema (Q_{O_2}), deve-se aumentar a vazão de ar, e diminuir o enriquecimento de oxigênio no sistema. Todavia, caso a velocidade de agitação já esteja próxima ao seu valor máximo (maior ou igual a 99% do limite superior de agitação) e ainda assim, a concentração de oxigênio dissolvido no sistema estiver baixa, tem-se duas opções. Observa-se se a soma das vazões de ar e oxigênio é maior que a vazão máxima permitida. Se sim, então diminui-se a vazão de ar, e aumenta-se o enriquecimento de oxigênio no sistema. Caso contrário aumenta-se a vazão de ar e mantém-se a vazão de oxigênio constante (HORTA, 2011).

A implementação em LabVIEW desse pseudocódigo, além de não ser trivial, pode levar a resultados não plenamente satisfatórios. Podem ocorrer atrasos na ação de controle e também na resposta dos fluxômetros de massa, responsáveis pela injeção de ar ou oxigênio no biorreator, já que a árvore de decisão fornece respostas abruptas (ΔO_2^- e ΔO_2^+) condicionadas

às mudanças nas condições comparativas. Ou seja, não se tem transições suaves, graduais, na ação de controle. O sistema apresentado opera introduzindo degraus nas vazões de ar e oxigênio (HORTA, 2011).

Além disso o SUPERSYS_HCDC possui um dispositivo de corte da vazão de alimentação do meio suplementar sempre que a concentração de oxigênio dissolvido cai a um valor limite. Isto ocorre para evitar a formação de metabólitos indesejáveis. Esta privação de nutrientes pode desencadear graves problemas tanto no próprio cultivo como na fase de síntese da proteína recombinante (HORTA, 2011).

Surge então a necessidade de melhoria deste sistema de controle automático das vazões de ar e oxigênio. Uma alternativa atraente é utilizando a lógica nebulosa (*fuzzy*).

2.4. *Fuzzy*

Nesta seção serão abordados os pressupostos teóricos da lógica nebulosa, do controle nebuloso e algumas aplicações da lógica nebulosa em controle de bioprocessos.

2.4.1. *Fuzzy*: Fundamentos

Nesta seção serão apresentados os principais conceitos relacionados à teoria *fuzzy*. Serão abordadas as teorias de lógica nebulosa e conjuntos nebulosos e, em seguida, abordar-se-á os conceitos de funções de pertinência, variáveis linguísticas e operadores lógicos.

2.4.1.1. Lógica Nebulosa

Em um sentido estrito, a lógica nebulosa é um sistema lógico que visa a formalização de um raciocínio aproximado. “Andar alguns metros” e “copo meio vazio” são expressões que mostram a comunicação de modo vago e a incerteza do pensamento humano. São situações que violam a suposição dicotômica da lógica clássica, em que uma informação é verdadeira ou falsa, não podendo ser parcialmente falsa ou parcialmente verdadeira. A lógica nebulosa surge como uma alternativa para lidar com essa incerteza. É um formalismo matemático capaz de capturar informações vagas, descritas em uma linguagem natural e convertê-las para um formato numérico, permitindo a sua manipulação pelos computadores (ZADEH, 1994; SOUSA, 1999; TAKEMURA, 2016).

2.4.1.2. Conjuntos Nebulosos

Em 1965 o professor da Universidade de Berkley, Lotfali Askar Zadeh, introduziu o conceito de conjuntos nebulosos (*Fuzzy Sets*), que como o próprio nome sugere, é uma teoria de conjuntos (ou classes) cujas fronteiras são indefinidas. Zadeh observou que tanto no plano físico (mundo real), quanto no plano matemático, houve um aumento no número de conjuntos que não possuíam um critério de pertinência claramente definido. Conjuntos como “Temperatura Alta” ou “Pressão Baixa”, definidos até então pela lógica ordinária - em que um elemento pode pertencer ou não a certo conjunto - eram extremamente imprecisos e ainda assim tinham papel fundamental nas comunicações, abstrações e reconhecimento de padrões. Assim, ele estabeleceu que um conjunto nebuloso seria uma classe com diversos graus de pertinência. Ou seja, a cada elemento seria permitido um determinado grau de pertinência em relação ao conjunto. O grau de pertinência é um valor numérico entre 0 e 1 (em que 0 significa completa exclusão, e 1 completa pertinência) que traduz a possibilidade de um certo elemento de pertencer ou não a um determinado conjunto (ZADEH, 1965; SOUSA, 1999).

2.4.1.3. Funções de Pertinência

Quando se considera todos os elementos de um conjunto, tem-se então as funções de pertinência, através das quais tem-se o grau de pertinência de cada elemento (SOUSA, 1999). Elas se assemelham as funções de probabilidade, que descrevem a probabilidade relativa de uma variável tomar um certo valor. Porém há diferenças conceituais bastante expressivas principalmente nas regras de combinação das funções de pertinência. Além disso, a noção de um conjunto nebuloso é completamente não estatística por natureza (ZADEH, 1965).

As funções de pertinência podem assumir diversas formas. Elas podem ser: quadradas, retangulares ou trapezoidais (estas podem levar a perda de informações em regiões onde sua inclinação é igual a 0) triangulares, da forma-Z ou forma-S (que são formas mais suaves e reduzem a perda de informação) entre outras (NELLES, 2001).

2.4.1.4. Variáveis Linguísticas

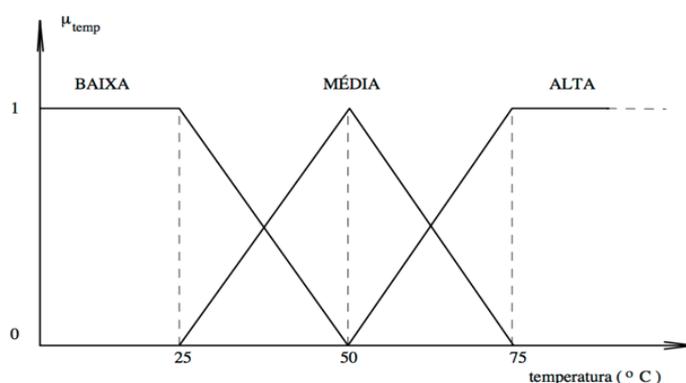
Outro conceito importante dentro da teoria da lógica nebulosa são as variáveis linguísticas. Enquanto que na matemática clássica são utilizadas variáveis numéricas, a lógica

nebulosa utiliza variáveis linguísticas (variáveis cujos valores não são números, mas palavras ou expressões), por exemplo: temperatura, vazão, pressão, etc. Isto decorre da facilidade da utilização destes termos para expressão e processamento das relações no formato de regras, haja vista que a caracterização linguística é, em geral, menos específica que a numérica (ZADEH,1975; NELLES, 2001).

Variáveis linguísticas correspondem a variáveis descritas, a priori, por expressões como “devagar”, “rápido”, “alto”, “baixo”, que correspondem a valores linguísticos da variável linguística. Zadeh (1994) destaca um dos aspectos mais importantes do conceito de variável linguística ligado aos valores linguísticos, o de considera-los como uma forma de compressão de dados. Esta forma de compressão dos dados é chamada de granulação. Os conjuntos nebulosos podem ser vistos como grânulos. Embora a granulação seja uma generalização da forma clássica de quantização, na primeira tem-se que os valores são uma sobreposição de conjuntos nebulosos. Enquanto que na segunda os valores são intervalos. As vantagens da granulação frente a quantização são: maior generalização, maior semelhança com o pensamento humano e a transição de um valor linguístico para outro é mais suave e gradual, resultando em maior robustez (ZADEH,1994; BELOHLAVEK; DAUBEN; KLIR, 2017).

Toma-se como exemplo a variável linguística temperatura, que pode admitir 3 valores linguísticos: “baixa”, “média” e “alta”, tal como mostra a Figura 1.

Figura 1 - Variável linguística temperatura



Fonte: GOMIDE; GODWIN, 1994 - modificado

Toma-se três valores arbitrários: 20, 30 e 80°C. Observando as funções de pertinência tem-se que 20°C é uma temperatura baixa com grau de pertinência 1, 30°C é uma temperatura baixa com grau de pertinência 0,8 ou uma temperatura média com grau de pertinência 0,2 e por fim 80°C é uma temperatura alta com grau de pertinência 1. A noção de elementos linguísticos

associados a um grau de pertinência com relação a valores linguísticos de variáveis linguísticas é a noção básica de sistemas nebulosos (SOUSA, 1999).

2.4.1.5. Operadores Lógicos

Muitas vezes observa-se que mais de uma variável linguística pode ser empregada em uma mesma colocação linguística. Por exemplo: “Se a temperatura for baixa e o vento forte, então a sensação é de muito frio”. “Temperatura”, “vento” e “sensação” são as variáveis linguísticas em questão. “Baixa”, “forte” e “muito frio” são os valores linguísticos das variáveis linguísticas. “Temperatura baixa e vento forte” é o antecedente, e “sensação de muito frio” é o consequente. Observe que as duas variáveis linguísticas no antecedente são ligadas por um conector, “e”, chamado operador lógico nebuloso de conjunção (NELLES, 2001).

A conjunção de dois ou mais conjuntos nebulosos é feita por operadores lógicos, chamados normas-t. Dentre elas estão:

- A) Mínimo: utiliza-se o valor mínimo entre os graus de pertinência.
- B) Produto: utiliza-se o produto dos graus de pertinência.
- C) Diferença limitada: utiliza-se o valor máximo entre 0 e a somatória dos graus de pertinência menos 1.

Para a disjunção (conector “ou”) de dois ou mais conjuntos nebulosos também são utilizados operadores lógicos, chamados conormas-t. Dentre elas estão:

- A) Máximo: utiliza-se o valor máximo entre os graus de pertinência.
- B) Soma algébrica: utiliza-se a soma dos graus de pertinência subtraindo-se o produto dos mesmos.
- C) Soma limitada: utiliza-se o mínimo entre 1 e a somatória dos graus de pertinência.

Por fim tem-se o operador de negação “não”, que nada mais é do que o complementar do grau de pertinência (ou seja, 1 menos o grau de pertinência).

Para modelos nebulosos, a operação de conjunção pelo operador “mínimo” é a mais utilizada (NELLES, 2001).

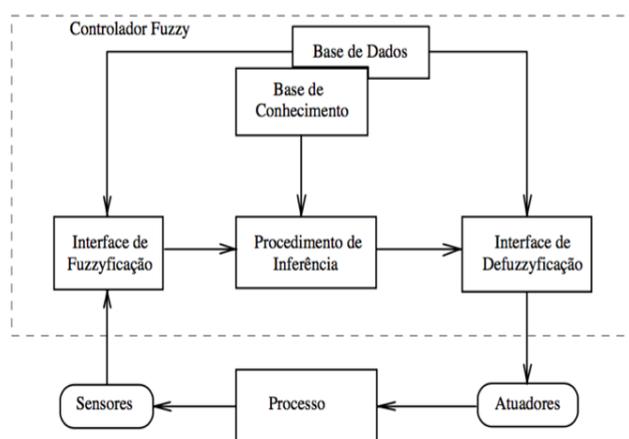
2.4.2. Controle Nebuloso

Nesta seção serão apresentados conceitos fundamentais relacionados ao funcionamento do controlador nebuloso tais como regras nebulosas, avaliação de sistemas nebulosos e tipos de sistemas nebulosos. Por fim serão abordados alguns exemplos de aplicações de controladores nebulosos.

2.4.2.1. Controlador Nebuloso

A ideia do controle *fuzzy* é modelar ações de controle baseadas no conhecimento do especialista. Ou seja, o controlador nebuloso fornece um algoritmo que pode converter estratégias de controle de natureza qualitativa (baseadas em horas de observação do processo) em uma estratégia de controle automática. O que difere da abordagem clássica, em que se busca desenvolver métodos de controle baseados na modelagem matemática do processo. A motivação dessa abordagem advém de casos em que o conhecimento de especialista era vasto e o sistema possuía modelagem matemática complexa. A Figura 2 ilustra a estrutura básica de um controlador nebuloso (LEE,1990; GOMIDE; GODWIN, 1994; SOUSA, 1999).

Figura 2 - Estrutura básica de um controlador nebuloso



Fonte: GOMIDE; GODWIN, 1994

Seus componentes básicos são:

- A. Interface de *fuzzyficação*: mensura os valores das variáveis de entrada e conduz um mapeamento que transfere o conjunto das variáveis de entrada aos seus

correspondentes no universo de discurso *fuzzyficado*. Por fim conduz a *fuzzyficação*, que converte números reais a valores linguísticos (conjuntos nebulosos).

- B. Base de conhecimento: compreende a base de regras que caracteriza a estratégia de controle e as metas de controle. Armazena toda a informação sobre o processo, que caracteriza as regras linguísticas e as funções de pertinência.
- C. Procedimento de Inferência: é o cerne do controle nebuloso. É onde ocorre o processamento dos dados de entrada *fuzzyficados*, junto com as regras de controle, a fim de inferir as ações de controle *fuzzy*, utilizando a lógica nebulosa.
- D. Interface de *Defuzzyficação*: transforma ações de controle *fuzzy* em ações não *fuzzy* através do mapeamento dos valores de saída *fuzzyficados* aos seus correspondes no universo de discurso real (LEE,1990; GOMIDE; GODWIN, 1994).

Durante a implementação do controlador nebuloso é necessária a definição de alguns parâmetros e de um protocolo de controle (com as principais estratégias de controle). Ambos advêm do conhecimento do especialista do processo. Alguns desses parâmetros são fixos (estruturais), outros são alterados de tempos em tempos (sintonização).

São exemplos de parâmetros estruturais: número de variáveis de saída e de entrada, variáveis linguísticas, funções de pertinência e regras de controle.

São exemplos de parâmetros de sintonização: parâmetros das funções de pertinência e universos de discurso das variáveis (GOMIDE; GUDWIN, 1994).

2.4.2.2. Regras Nebulosas

A forma mais usual de se expressar um conhecimento é utilizando as regras do tipo “Se-Então” ou regras do tipo “condição-ação”. Tomemos como exemplo um operador que monitora um reator químico. Uma das principais variáveis para o bom andamento de uma reação química é o controle preciso da temperatura. Logo, o operador tem em mente a seguinte regra: “Se a temperatura do reator aumentar, então abre-se a válvula do fluido refrigerante”. Observa-se que esta regra “se-então” tem uma premissa “Se a temperatura aumentar” e um conseqüente “então abre-se a válvula de fluido refrigerante”. Esta regra equivaleria a implicação $A_i \rightarrow B_i$, que por sua vez corresponde ao produto cartesiano $A_i \times B_i$ de dois conjuntos nebulosos. Dado que o comportamento do sistema é caracterizado por todas as regras nebulosas, o sistema de controle então é representado por uma única relação *fuzzy* (combinação linear das relações nebulosas provenientes das regras nebulosas). Ou seja, $R = \sum_i A_i \times B_i$. Conclui-se então que a descrição

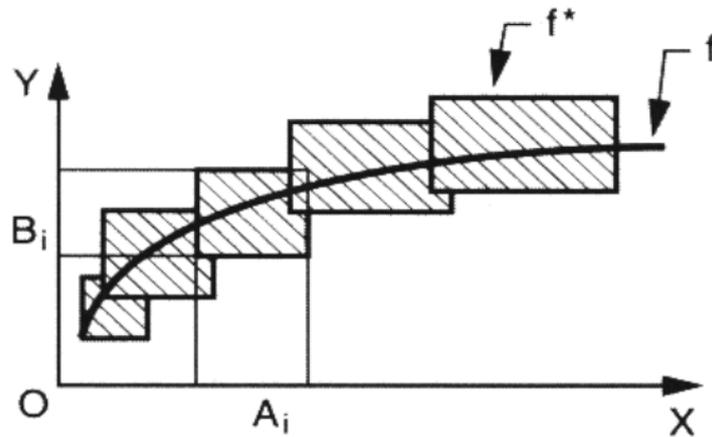
geral do sistema pode estar associada a um *fuzzy graph* (desde que a implicação se dê através de um produto cartesiano) (GOMIDE; GUDWIN, 1994; SOUSA, 1999).

O *fuzzy graph* pode ser interpretado como uma representação granular de dependências funcionais e relações.

Um *fuzzy graph* f^* de uma dependência funcional $f: X \rightarrow Y$, tal como mostram a Figura 3 (onde X e Y são variáveis linguísticas) serve para fornecer uma representação de f na forma da Equação 9 (ZADEH, 1965):

$$f^* = A_1 \times B_1 + A_2 \times B_2 + \dots + A_n \times B_n = \sum_i A_i \times B_i \quad (9)$$

Figura 3 - Representação da função f e seu Fuzzy Graph



Fonte: ZADEH, 1994 - Adaptado

Em que A_i e B_i são subconjuntos nebulosos contíguos de X e Y, respectivamente.

O procedimento de inferência anterior é decorrente do *modus ponens* apresentado no Quadro 2 (SOUSA, 1999).

Quadro 2 - *Modus ponens* generalizado

X é A	Antecedente
$(X,Y) \in \sum_i A_i \times B_i$	Implicação
Y é B	Consequente

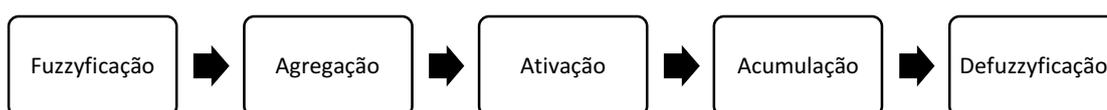
Fonte: SOUSA, 1999

As regras “se-então” são chamadas de declarações condicionais *fuzzy* ou regras *fuzzy*.

2.4.2.3. Avaliação de sistemas nebulosos

A avaliação de um sistema nebuloso é feita em cinco fases, com base no diagrama da Figura 4.

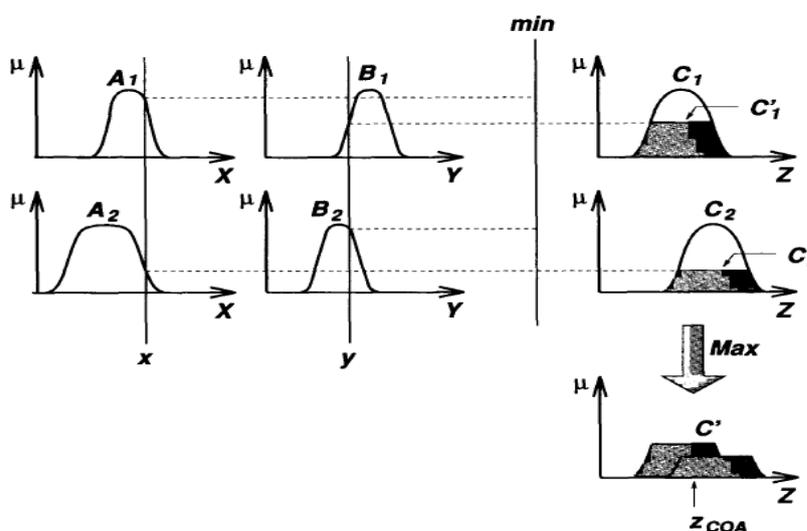
Figura 4 - Processo de avaliação de um sistema nebuloso



Fonte: NELLES, 2001 - adaptado

Para ilustrar as cinco fases do processo de avaliação de sistemas de inferência nebulosos, tem-se a Figura 5. O sistema é composto de duas entradas *crisp* (x e y), duas variáveis linguísticas de entrada (A e B) - e suas respectivas funções de pertinência - e duas regras nebulosas. A saída é um valor *crisp* z .

Figura 5 - Processo de inferência nebuloso



Fonte: JANG; SUN; MIZUTANI, 1997

A *Fuzzyficação* utiliza as funções de pertinência para mapear as entradas “crisp” aos graus de pertinência. Ou seja, toma-se os valores das variáveis de entrada do controlador

nebuloso e *fuzzyfica-os*, transformando números reais em conjuntos nebulosos, de modo que possam ser equiparados às categorias linguísticas (NELLES, 2001).

Ao conjunto das três fases seguintes (Agregação, Ativação e Acumulação) dá-se o nome de inferência. É nessa fase que se processam os dados de entrada, juntos com as regras, de modo a inferir ações de controle.

Na fase de Agregação são combinados os antecedentes de cada regra, utilizando operadores “ou” e “e”, ao grau de acionamento da regra.

Na fase de Ativação os graus de acionamento são utilizados para calcular a ativação das saídas das regras. A combinação dos graus de pertinência de todas as colocações linguísticas é chamada de grau ou força de acionamento da regra. Quando há mais de um antecedente a ação de controle é ativada de acordo com o menor grau de pertinência dentre os antecedentes. Esta operação é feita “cortando” as funções de pertinência de saída no grau de acionamento da regra (NELLES, 2001).

Na fase de Acumulação as ativações das saídas de todas as regras são combinadas, computando-se o máximo de todas as ativações das saídas. A união destes conjuntos nebulosos produz a figura final, que por sua vez também é um conjunto nebuloso.

Caso não seja necessário um valor *crisp* tal como na entrada, o processo de inferência termina. Isto acontece quando a resposta esperada não é um número, mas sim uma variável linguística (NELLES, 2001). Caso seja necessário um valor *crisp* de saída, tem-se a última fase, a *Defuzzyficação*. A técnica mais comumente empregada é a do centro de gravidade, cuja equação é dada pela Equação 10.

$$y_0 = \frac{\int y Y(y) dy}{\int Y(y) dy} \quad (10)$$

Em que y_0 é o valor *defuzzyficado*, y é o valor da variável linguística, $Y(y)$ é a função de pertinência.

Observa-se que os três últimos passos da análise dos sistemas nebulosos dependem do sistema em estudo (NELLES, 2001).

2.4.2.4. Tipos de sistemas nebulosos

Destacam-se três tipos de sistemas nebulosos: Mamdani (Linguístico), Takagi-Sugeno e Singleton (NELLES, 2001).

O sistema nebuloso do tipo Mamdani, também conhecido como sistema nebuloso linguístico é baseado na seguinte regra apresentada pela Equação 11:

$$\text{Se } u_1 = A_{i1} \text{ e } u_{i2} = A_{i2} \dots u_p = A_{ip} \text{ então } y = B_i \quad (11)$$

Sendo $u_{i1} \dots u_p$ as entradas, y a saída, A e B conjuntos nebulosos.

São sistemas mais gerais e mais complexos, pois tanto as entradas quanto as saídas são descritas por variáveis linguísticas.

Sistemas nebulosos Takagi-Sugeno por sua vez seguem a seguinte regra apresentada pela Equação 12:

$$\text{Se } u_1 = A_{i1} \text{ e } u_{i2} = A_{i2} \dots u_p = A_{ip} \text{ então } y = f_i(u_1, u_2, \dots, u_p) \quad (12)$$

Em que f_i é uma função qualquer.

Eles diferem dos sistemas nebulosos linguísticos quanto ao consequente. Enquanto que estes possuem uma saída na forma de conjunto nebuloso, os sistemas Takagi-Sugeno possuem como consequente uma função. Em geral são utilizados sistemas Takagi-Sugeno de primeira ordem, em que o consequente da regra é uma função linear das entradas (NELLES, 2001).

Algumas vantagens dos sistemas Takagi-Sugeno são: computacionalmente eficientes e trabalham bem com técnicas lineares (controlador PID), de otimização e adaptativas (ANFIS)

Os sistemas nebulosos Singleton são um caso específico tanto dos sistemas linguísticos, quanto dos sistemas Takagi-Sugeno. Eles seguem a seguinte regra apresentada pela Equação 13:

$$\text{Se } u_1 = A_{i1} \text{ e } u_{i2} = A_{i2} \dots u_p = A_{ip} \text{ então } y = S_i \quad (13)$$

Em que S_i é um valor real, chamado singleton.

Observa-se então que no caso de um sistema linguístico com um valor *crisp* de saída, tem-se então um sistema singleton. Ao mesmo tempo, um sistema Takagi-Sugeno, cujo consequente for uma função de ordem zero (uma constante) também se reduz a um sistema singleton (NELLES, 2001).

2.4.3. Aplicações do controle nebuloso em bioprocessos

Na área de controle de bioprocessos um exemplo bem-sucedido da utilização do controle nebuloso é no controle de reatores bioquímicos para produção de Cefalosporina C em regime de batelada alimentada, estudo realizado por SOUSA (1999).

A Cefalosporina C é um antibiótico produzido industrialmente utilizando o fungo *Cephalosporium acremonium*. Observou-se que sob altas concentrações de glicose, é repimida a produção do aminoácido metionina, resultando na redução da produção do antibiótico. Pode-se identificar duas fases distintas durante o processo de produção. A primeira é a de crescimento do microrganismo, caracterizada por altas concentrações de glicose no meio e a segunda é a fase de produção que promove a síntese de enzimas específicas, que catalisam a produção do antibiótico (SOUSA, 1999).

O trabalho teve como objetivo a implementação de um sistema para determinação do momento preciso em que se deve iniciar a alimentação de sacarose. Dadas as dificuldades da modelagem matemática do processo e das características altamente não lineares de um processo fermentativo, foi acertada a utilização de um controlador nebuloso. Este foi baseado na metodologia nebulosa que, por sua vez, integra a corrente geral das metodologias baseadas no conhecimento (*Knowledge-Based*) (SOUSA, 1999).

Os cultivos foram realizados em biorreatores Bioflo II-C da New Brunswick Scientific Co., Inc. Algumas condições adotadas para o cultivo e a descrição de parte da instrumentação utilizada são detalhadas nos itens subsequentes (SOUSA, 1999).

- A. Temperatura mantida numa faixa entre 25 e 30°C (utilizando-se um microprocessador baseado em um controlador PID) e aferida por um sensor RTS (*Resistance Temperature Sensor*), PT-100.
- B. Faixa de pH para o crescimento mantida entre 4 e 9, e para a produção, próxima de 7 (utilizando-se um controlador PID (pHC-200) para acionamento de duas bombas injetoras de ácido ou base) e aferida por um sensor amperométrico (pHE-200) imerso no meio fermentativo.
- C. Oxigênio dissolvido controlado em uma faixa de 5 a 95% da saturação (utilizando-se um controlador PID que atuava na velocidade de agitação) e medido através de um eletrodo imerso no reator.
- D. Adição de antiespumante (utilizando-se um controlador LC-100 que aciona a bomba injetora de antiespumante).

- E. Vazão do meio suplementar (utilizando um controlador YC-600 conectado a bomba injetora de meio suplementar - esta por sua vez foi manipulada por uma função lógica programável YIC-600).

Nos experimentos foi utilizado o microrganismo *C. acremonium* ATCC 48272 (C-10) cedido pela CODETEC (Campinas – SP). As principais fontes de carbono utilizadas foram a glicose (fase de crescimento) e a sacarose (fase de produção) (SOUSA,1999).

A principal ideia para se atingir os objetivos do trabalho foi baseada na informação de que o fim da trofofase pode ser caracterizado como um ponto de máximo. Este indica uma transição entre liberação crescente de gás carbônico e liberação decrescente. Porém, devido a problemas relacionados ao monitoramento da concentração de gás carbônico nos cultivos iniciais, optou-se por uma nova concepção do sistema de controle nebuloso. Este passaria a operar em três níveis, o primeiro chamado de nível de atenção (programado para agir apenas de modo a determinar o momento em que o nível de liberação de gás carbônico é alto suficiente para ser monitorado), o segundo chamado de nível de ação (programado para detectar o nível máximo de liberação de gás carbônico e iniciar a alimentação do meio suplementar) e por fim um terceiro nível, chamado nível de proteção (programado para evitar possíveis equívocos na identificação do final da trofofase) (SOUSA, 1999).

Para o primeiro nível de controle (nível de atenção) foram definidas duas variáveis linguísticas: tempo e nível de saturação de gás carbônico. Os valores linguísticos adotados para cada uma das variáveis linguísticas foram: “cedo”, “tarde” e “baixo”, “alto” (respectivamente).

As regras nebulosas criadas para o nível de atenção foram:

1. SE TEMPO CEDO E CO₂ BAIXO, ENTÃO NÃO ATIVA O NÍVEL DE AÇÃO.
2. SE TEMPO TARDE E CO₂ BAIXO, ENTÃO NÃO ATIVA O NÍVEL DE AÇÃO.
3. SE TEMPO CEDO E CO₂ ALTO, ENTÃO ATIVA O NÍVEL DE AÇÃO.
4. SE TEMPO TARDE E CO₂ ALTO, ENTÃO ATIVA O NÍVEL DE AÇÃO.

Para o segundo nível de controle (nível de ação) foram definidas duas variáveis linguísticas: “negativa” e “zero ou positiva”

As regras nebulosas criadas para o nível de atenção foram:

5. SE VARIAÇÃO DE CO₂ ZERO OU POSITIVA, ENTÃO NÃO ALIMENTA.
6. SE VARIAÇÃO DE CO₂ NEGATIVA, ENTÃO ALIMENTA.

Para o terceiro nível de controle (nível de proteção) foram definidas duas variáveis linguísticas: tempo e nível de variação de CO₂. Os valores linguísticos adotados para cada uma

das variáveis linguísticas foram: “cedo”, “tarde” e “Negativa”, “Positiva” (respectivamente) (SOUSA, 1999).

As regras nebulosas criadas para o nível de atenção foram:

7. SE TEMPO CEDO E VARIAÇÃO DE CO₂ NEGATIVA, ENTÃO ESPERA.
8. SE TEMPO TARDE E VARIAÇÃO DE CO₂ NEGATIVA, ENTÃO CONFIRMA ALIMENTAÇÃO.
9. SE TEMPO CEDO E VARIAÇÃO DE CO₂ POSITIVA, ENTÃO CANCELA ALIMENTAÇÃO.
10. SE TEMPO TARDE E VARIAÇÃO DE CO₂ POSITIVA, ENTÃO CANCELA ALIMENTAÇÃO.

Os dois primeiros experimentos permitiram estabelecer as funções de pertinência e a base de regras do controlador. O terceiro experimento foi utilizado para testar a viabilidade e a robustez do algoritmo. Os resultados foram bastante satisfatórios, obtendo-se um controlador eficaz na detecção do momento em que se deve iniciar alimentação do meio suplementar no processo de Cefalosporina C em regime de batelada alimentada (SOUSA, 1999).

Nucci e colaboradores (2005) desenvolveram um algoritmo baseado na lógica nebulosa para a determinação do momento de máxima concentração de enzimas durante um cultivo de *Bacillus megaterium* em um biorreator agitado e aerado automaticamente.

A Penicilina G-Acilase (PGA) é uma importante enzima industrial utilizada, principalmente como biocatalisador na produção de antibióticos Beta-lactâmicos semi-sintéticos. O microrganismo *Bacillus megaterium*, além de produzir a enzima, secreta-a no meio, o que simplifica os processos de separação e purificação. Notou-se que durante o processo produtivo da PGA pelo *B. Megaterium* a fração molar do dióxido de carbono atinge um máximo. Logo após este pico pode ser observado que a concentração enzimática atinge seu máximo. Assim os pesquisadores desenvolveram um sistema de lógica nebulosa baseado em Sousa (1999), para identificar o momento em que ocorre este pico. A determinação exata deste momento é fundamental para redução de custos do processo, redução de tempo morto e evitar a degradação do produto (NUCCI *et al.*, 2005).

O microrganismo utilizado foi *B. megaterium* ATCC 14945 cedido pela Fundação Tropical (Campinas – SP). A atividade enzimática foi medida através da hidrólise de 0,2 M de PGA em um pH 8 a 37°C (a hidrólise da penicilina G utilizando a PGA produz o ácido 6-aminopenicilânico – 6-APA). A concentração de 6-APA produzido foi determinada por espectrofotometria a 415 nm.

O experimento foi realizado em biorreator de 5 litros (Bioflo II-C), temperatura de 30°C e vazão específica de alimentação de ar de 2 l/min. A concentração de oxigênio dissolvido foi

mantida a 20% por um controle PID atuando diretamente na frequência de agitação. O biorreator foi acoplado a um sistema composto por um controlador lógico programável e por um sistema supervisão para aquisição de dados e controle de processo (NUCCI *et al.*, 2005).

O algoritmo *fuzzy* foi escrito em linguagem *Fortran*, compilado em formato DLL (*Dynamic Link Library*) e implementado numa plataforma desenvolvida no *Microsoft Visual Basic*.

O controlador *fuzzy* foi dividido em três níveis: atenção, identificação do pico e proteção (NUCCI *et al.*, 2005).

As regras nebulosas para o nível de atenção foram:

1. SE TEMPO É CEDO E A FRAÇÃO MOLAR DE CO₂ É BAIXA, ENTÃO NÃO ATIVA O NÍVEL DE AÇÃO.
2. SE TEMPO É TARDE E A FRAÇÃO MOLAR DE CO₂ É BAIXA, ENTÃO NÃO ATIVA O NÍVEL DE AÇÃO.
3. SE TEMPO É CEDO E A FRAÇÃO MOLAR DE CO₂ É ALTA, ENTÃO ATIVA O NÍVEL DE AÇÃO.
4. SE TEMPO É TARDE E A FRAÇÃO MOLAR DE CO₂ É ALTA, ENTÃO ATIVA O NÍVEL DE AÇÃO.

O nível de ativação era ativado quando o grau de pertinência excedia 0,8.

Em seguida, a derivada da fração molar de gás carbônico em relação ao tempo foi calculada (NUCCI *et al.*, 2005).

As regras nebulosas para o nível de identificação foram:

5. SE A VARIAÇÃO FRAÇÃO MOLAR DE CO₂ É ZERO OU POSITIVA, ENTÃO NÃO NOTIFICA O PICO DE FRAÇÃO MOLAR.
6. SE A VARIAÇÃO FRAÇÃO MOLAR DE CO₂ É NEGATIVA, ENTÃO NOTIFICA O PICO DE FRAÇÃO MOLAR.

Para evitar a identificação de falsos picos o nível de proteção foi elaborado com 4 regras nebulosas (NUCCI *et al.*, 2005).

7. SE O TEMPO É CEDO E A VARIAÇÃO DE CO₂ É NEGATIVA, ENTÃO ESPERE.
8. SE O TEMPO É TARDE E A VARIAÇÃO DE CO₂ É NEGATIVA, ENTÃO CONFIRMA PICO.
9. SE O TEMPO É CEDO E A VARIAÇÃO DE CO₂ É POSITIVA, ENTÃO CANCELA O PICO.
10. SE O TEMPO É TARDE E A VARIAÇÃO DE CO₂ É POSITIVA, ENTÃO CANCELA O PICO.

Três experimentos foram realizados para fornecer os dados para o ajuste dos parâmetros do controlador nebuloso. Em seguida, outros três experimentos (com comportamentos bastante distintos) foram utilizados para validação e testes de robustez do controlador.

Para os testes de ajuste dos parâmetros, os resultados foram positivos. O algoritmo nebuloso identificou, com precisão, a região de máxima concentração de enzimas. Já para os testes de robustez do controlador nebuloso os resultados também foram positivos. O algoritmo proveu estimativas de tempo para a máxima atividade enzimática dentro de uma faixa experimental correta (NUCCI *et al.*, 2005).

Reis e colaboradores (2009) acoplaram um sensor capacitivo a um algoritmo *fuzzy* para controle da taxa de alimentação do meio suplementar em um cultivo de *Saccharomyces cerevisiae*. Estratégias clássicas de controle apresentam problemas devido à não linearidade e à alta complexidade deste bioprocesso. Assim o emprego de dados obtidos a partir de sensores *in situ* para regulação dos parâmetros é imperativo para melhorar a estratégia de controle (REIS *et al.*, 2009).

Foi utilizado um sistema de inferência nebuloso do tipo Mandani com duas entradas, o quociente respiratório (RQ) e o erro (e) – diferença entre a velocidade de crescimento específica de referência (μ_{REF}) e seu valor mensurado *on line* (μ) – a variável de saída foi a vazão de glicose (REIS *et al.*, 2009).

Foram definidos quatro valores linguísticos para cada variável linguística. Para o quociente respiratório foram definidos os seguintes valores linguísticos: baixo (RQL), bom (RQG), alto (RQH) e muito alto (RQVH). Para a variável erro foram definidos os seguintes valores linguísticos: diferença negativa (ND), diferença baixa (LD), diferença média (MD) e diferença alta (HD). Por fim, para a variável vazão de glicose foram definidos os seguintes valores linguísticos: vazão nula (NF), vazão baixa (LF), vazão média (MF) e vazão alta (HF).

O cultivo foi realizado em biorreatores tipo tanque agitado de 6 litros, operando em regime de batelada alimentada, e a porcentagem de oxigênio dissolvido foi mantida acima de 45% da saturação (Mettler Toledo). A concentração de células viáveis foi mensurada por um sensor de biomassa (Fogale Nanotech) e a fração molar dos gases de exaustão foi aferida utilizando um analisador de gases (SICK – MAIHAK) (REIS *et al.*, 2009).

A escolha do tipo de função de pertinência e seus parâmetros foi realizada de forma heurística. Foram utilizadas funções trapezoidais e triangulares. A partir das variáveis linguísticas e seus valores linguísticos, foram determinadas 16 regras nebulosas.

1. Se RQ é RQL e erro é ND, então vazão de alimentação de glicose é LF
2. Se RQ é RQG e erro é ND, então vazão de alimentação de glicose é LF

3. Se RQ é RQH e erro é ND, então vazão de alimentação de glicose é LF
4. Se RQ é RQV e erro é ND, então vazão de alimentação de glicose é NF
5. Se RQ é RQL e erro é LD, então vazão de alimentação de glicose é MF
6. Se RQ é RQL e erro é MD, então vazão de alimentação de glicose é MF
7. Se RQ é RQL e erro é HD, então vazão de alimentação de glicose é HF
8. Se RQ é RQG e erro é LD, então vazão de alimentação de glicose é LF
9. Se RQ é RQG e erro é MD, então vazão de alimentação de glicose é LF
10. Se RQ é RQG e erro é HD, então vazão de alimentação de glicose é MF
11. Se RQ é RQH e erro é LD, então vazão de alimentação de glicose é NF
12. Se RQ é RQH e erro é MD, então vazão de alimentação de glicose é NF
13. Se RQ é RQH e erro é HD, então vazão de alimentação de glicose é NF
14. Se RQ é RQV e erro é MD, então vazão de alimentação de glicose é NF
15. Se RQ é RQV e erro é ND, então vazão de alimentação de glicose é NF
16. Se RQ é RQV e erro é HD, então vazão de alimentação de glicose é NF

Com a implementação do algoritmo *fuzzy* foi possível manter uma baixa concentração de glicose no meio (próximo a zero) e, conseqüentemente, a concentração de etanol produzida também foi baixa. A viabilidade celular se manteve alta durante todo o cultivo (98%) e a produtividade celular obtida foi alta (0,45g/l/h). Demonstrou-se que o controle nebuloso foi eficiente para controlar a vazão de alimentação do meio suplementar a partir do sinal de um sensor (REIS *et al.*, 2009).

Visando aperfeiçoar a função de controle das vazões de ar e oxigênio no programa SUPERSYS_HCDC, SOUSA e SOUSA (2015) desenvolveram um controlador nebuloso (*fuzzy*) para controle das vazões de ar e oxigênio. Foram utilizados dados de cultivo de *Escherichia coli* para expressão de proteínas recombinantes doados pelo professor Dr. Antonio Carlos Luperni Horta (HORTA, 2011).

Inicialmente foram criadas duas sub-rotinas no editor FIS do MATLAB. Uma sub-rotina “Delta” que analisa o “tamanho” da variação das vazões de ar e oxigênio, e uma sub-rotina “Princ” que verifica se haverá um incremento ou decréscimo de delta nas vazões de ar e/ou oxigênio (SOUSA; SOUSA, 2015). As variáveis linguísticas de entrada e saída criadas para cada sub-rotina, bem como seus respectivos valores linguísticos se encontram nos Quadros de 3 a 6.

Quadro 3 - Variáveis da sub-rotina “Delta”

Sub-rotina “Delta”	
Variáveis de entrada	Variáveis de saída
QO ₂ ant: vazão anterior de oxigênio (l/min)	Delta: variação das vazões (l/min)

Fonte: Sousa; Sousa (2015)

Quadro 4 - Variáveis da sub-rotina “Princ”

Sub-rotina “Princ”	
Variáveis de entrada	Variáveis de saída
CX: concentração de microrganismos (g/l)	QarEscolha: variável auxiliar
AgitMed: Agitação (rpm)	QO ₂ Escolha: variável auxiliar
OD: oxigênio dissolvido (%)	Qmáx: vazão máxima do sistema (l/min)
QarSe: variável auxiliar	

Fonte: Sousa; Sousa (2015)

Quadro 5 - Valores linguísticos das variáveis da sub-rotina “Delta”

Sub-rotina “Delta”	
QO ₂ ant	BP: baixa e positiva MP: média e positiva AP: alta e positiva
Delta	BP: baixa e positiva AP: alta e positiva

Fonte: Sousa; Sousa (2015)

Quadro 6 - Valores linguísticos das variáveis da sub-rotina “Princ”

Subrotina “Princ”	
CX	BP: baixa e positiva AP: alta e positiva
AgitMed	BP: baixa e positiva AP: alta e positiva MAP: muito alta e positiva
OD	BP: baixo e positivo MP: médio e positivo AP: alto e positivo
QarEscolha	Ganha Perde
QO ₂ Escolha	Ganha Perde Nulo
Qmáx	MP: média e positiva AP: alta e positiva
QarSe	Sim Não

Fonte: Sousa; Sousa (2015)

A partir das variáveis linguísticas e de seus respectivos valores linguísticos, foram definidas as funções de pertinência para cada uma das variáveis em questão. Tanto os parâmetros numéricos quanto as formas de cada uma das funções de pertinência foram obtidos heurísticamente, com base nos dados experimentais previamente obtidos em laboratório (SOUSA; SOUSA, 2015).

Por fim, a partir das variáveis linguísticas (entrada e saída) de seus valores linguísticos e das respectivas funções de pertinência, foram definidas as regras nebulosas do controlador, apresentadas nos Quadros 7 e 8. Para sub-rotina “Delta” foram criadas três regras nebulosas e para a sub-rotina “Princ” foram criadas sete regras nebulosas (SOUSA; SOUSA, 2015).

Enfatiza-se que tudo foi baseado no sistema previamente desenvolvido para controle das vazões de ar e oxigênio (Quadro 1), incluindo uma medida de CX.

Quadro 7 - Regras para a sub-rotina "Delta"

- | |
|--|
| <ol style="list-style-type: none"> 1. Se (QO₂ant é BP) então (Delta é AP). 2. Se (QO₂ant é AP) então (Delta é BP). 3. Se (QO₂ant é MP) então (Delta é BP). |
|--|

Fonte: Sousa e Sousa (2015)

Quadro 8 - Regras para a sub-rotina "Princ"

- | |
|---|
| <ol style="list-style-type: none"> 1. Se (CX é BP) e (AgitMed é BP) então (QarEscolha é GANHA) (QO₂Escolha é PERDE)(Qmáx é MP). 2. Se (CX é AP) e (AgitMed é AP) então (QarEscolha é GANHA) (QO₂Escolha é PERDE)(Qmáx é AP). 3. Se (CX é AP) e (AgitMed é BP) então (QarEscolha é GANHA) (QO₂Escolha é PERDE)(Qmáx é AP). 4. Se (CX é AP) e (AgitMed é MAP) e (OD é BP) e (QarSe é SIM) então (QarEscolha é PERDE) (QO₂Escolha é GANHA)(Qmáx é AP). 5. Se (CX é BP) e (AgitMed é AP) e (OD é BP) e (QarSe é SIM) então (QarEscolha é PERDE) (QO₂Escolha é GANHA)(Qmáx é MP). 6. Se (CX é AP) e (AgitMed é MAP) e (OD é BP) e (QarSe é NÃO) então (QarEscolha é GANHA) (QO₂Escolha é NULO)(Qmáx é AP). 7. Se (CX é BP) e (AgitMed é AP) e (OD é BP) e (QarSe é Não) então (QarEscolha é GANHA) (QO₂Escolha é NULO)(Qmáx é MP). |
|---|

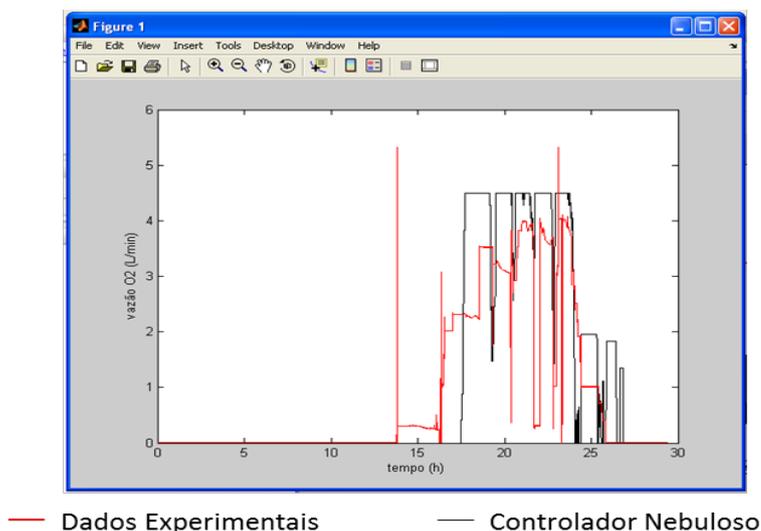
Fonte: Sousa e Sousa (2015)

Para gerenciar o uso do sistema de inferência nebuloso criado, foi desenvolvido um código em MATLAB. Este procurou aproximar ao máximo a tendência das vazões de ar e oxigênio observadas em um cultivo. Foram necessários diversos ajustes, realizados

heuristicamente, para que o sistema funcionasse de uma forma coerente com os resultados experimentais (SOUSA; SOUSA, 2015).

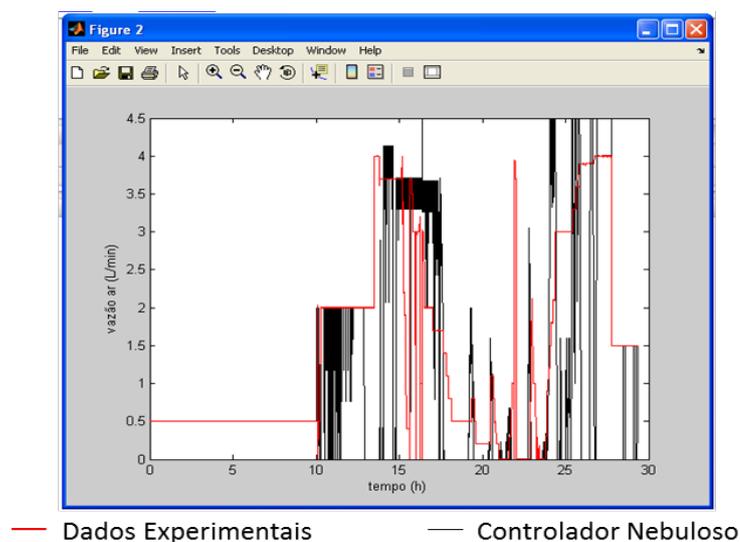
As Figuras 6 e 7 mostram os gráficos das vazões de ar e de oxigênio em função do tempo. A linha vermelha representa os dados experimentais (controlados utilizando a árvore de decisão) e a linha preta representa os dados obtidos com o controlador *fuzzy*.

Figura 6 - Gráfico da vazão de oxigênio (l/min) em função do tempo (h)



Fonte: Sousa; Sousa (2015)

Figura 7 - Gráfico da vazão de ar (l/min) em função do tempo (h)



Fonte: Sousa; Sousa (2015)

Observou-se que as tendências de necessidade de diminuição ou aumento das vazões de ar e oxigênio foram alcançadas pelo sistema. A lógica nebulosa foi capaz de indicar as

necessidades do processo estudado (cultivo de *E. coli* para expressão de proteínas recombinantes). Porém também foi observado que o controle apresentou algumas oscilações no decorrer da simulação. Assim surgiu a necessidade de melhoria deste sistema de inferência nebuloso.

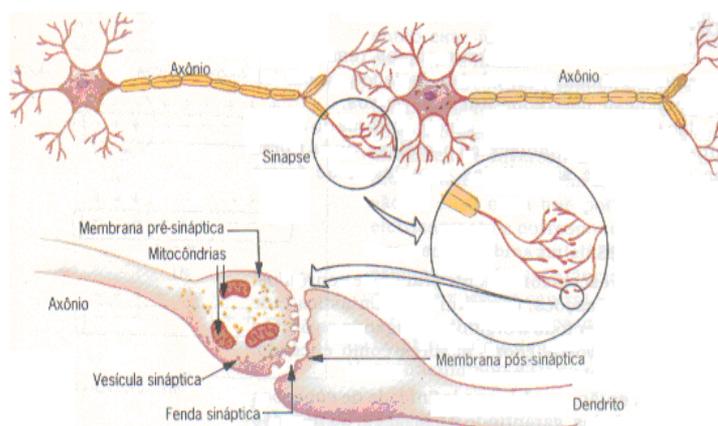
2.5. Redes Neurais

Nesta seção serão abordados os principais conceitos relacionados às redes neurais artificiais bem como algumas aplicações das mesmas em bioprocessos.

2.5.1. Redes Neurais Biológicas

O cérebro desempenha funções tais como reconhecimento de padrões, controle motor, percepção, inferência, intuição dentre outras. A unidade básica do processamento do cérebro é chamada de neurônio. O cérebro possui em média cerca de 100 bilhões de neurônios, cada unidade está interconectada através de sinapses, formando a rede neural biológica. O sistema nervoso humano possui três estágios: cérebro (centro do sistema, representado pela rede neural que recebe informações e afere decisões), receptores (convertem os estímulos do corpo ou do ambiente em impulsos elétricos) e os atuadores (convertem impulsos elétricos em respostas). A Figura 8 contém a descrição dos principais componentes de um neurônio biológico (CASTRO; ZUBEN, 2003; FINOCCHIO, 2014):

Figura 8 - Principais componentes de um neurônio biológico



Fonte: FINOCCHIO, 2014

Os neurônios são constituídos basicamente por:

- Corpo Celular: região onde se encontra o núcleo.
- Dendritos: Prolongamentos, em geral ramificados, responsáveis pelo recebimento dos sinais.
- Axônio: Prolongamento do neurônio que possui ramificações apenas na extremidade, responsável por conduzir o sinal.
- Sinapses: Responsáveis por transmitir as informações de uma célula para outra.

O sentido do impulso nervoso é sempre do dendrito, passando pelo corpo celular e transmitido a outro neurônio pelo axônio.

Quando os impulsos nervosos atingem as extremidades do axônio da célula pré sináptica, um potencial de membrana aparece como resultado da integração dos sinais de entrada, que irão determinar se o neurônio produz um sinal de saída ou não. O potencial da ação resulta na liberação de substâncias químicas denominadas neurotransmissores. Estes têm a capacidade de se combinar com o receptor presente na membrana da célula pós-sináptica, desencadeando o impulso nervoso. Esta é chamada de sinapse química. Existe também a sinapse elétrica, em que íons conectam canais de um neurônio ao outro, de forma que alterações elétricas em uma célula são transmitidas instantaneamente à próxima. A ativação de um neurônio só ocorre se o potencial da membrana é maior que um dado limiar.

Uma das principais características da rede neural é a representação distribuída da informação e seu processamento paralelo (CASTRO; ZUBEN, 2003; FINOCCHIO, 2014).

Por fim, é interessante ressaltar o processo de aprendizagem do cérebro. Este é uma das bases da teoria de redes neurais artificiais.

O sistema nervoso está continuamente sofrendo modificações e atualizações. Foi observado que o mesmo possui graus de plasticidade: as mudanças podem ser rápidas e fáceis, lentas e profundas e mudanças que embora sejam caracterizadas como permanentes, ainda podem ser modificáveis. Em geral, esta aprendizagem é fruto de modificações dos neurônios:

- Nascimento ou morte de dendritos;
- Esticamento/encolhimento dos dendritos, que permitem/eliminam a conexão com outras células;
- Criação e alteração de novas sinapses. Esta modulação sináptica depende de mecanismos de adaptação de neurônios individuais e de redes como um todo.

2.5.2. Redes Neurais Artificiais

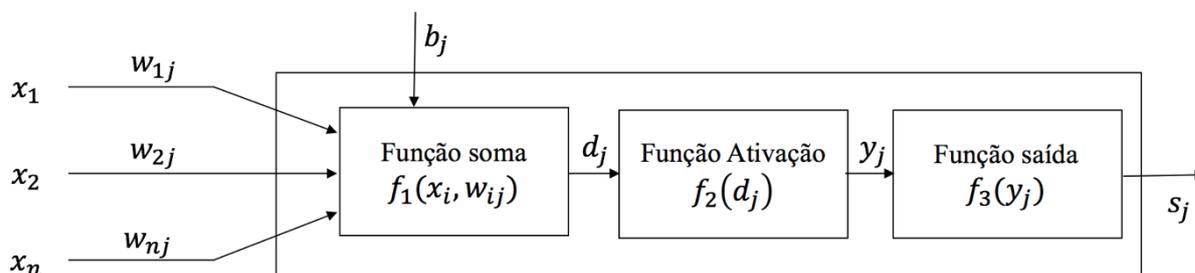
A criação das redes neurais artificiais foi motivada pela capacidade do cérebro humano de executar tarefas de grande complexidade, não lineares e com processamento paralelo da informação.

De acordo com Palma Neto e Nicoletti (2005), as redes neurais artificiais podem ser definidas como uma rede contendo múltiplos processadores simples (cada um deles possuindo uma certa quantidade de memória) que são conectados através de meios de comunicação (conexões), aos quais são associados valores (pesos) numéricos. Uma rede neural pode ser classificada com base em: seus processadores básicos (neurônios), pela sua função de ativação (representa o estado de um neurônio), pelo padrão da conexão existente entre os neurônios e pelo seu algoritmo de treinamento (algoritmo de aprendizado) (PALMA NETO; NICOLETTI, 2005).

2.5.2.1. Estrutura básica de uma rede neural artificial

A Figura 9 mostra a estrutura básica de um neurônio artificial.

Figura 9 - Estrutura básica de um neurônio artificial



Fonte: PALMA NETO; NICOLETTI, 2005 - modificado

O neurônio é a base da rede neural. Ele também é chamado de elemento processador uma vez que ele recebe o vetor de entradas x , com componentes x_i ($i = 1$ até n) e manipula estas entradas (atividades) para fornecer a saída s_j (BAUGHMAN; LIU, 1995).

Num primeiro momento, a função soma combina as entradas que chegam ao neurônio em um valor. Esta soma depende tanto das entradas (x_i) quanto dos pesos associados (w_{ij}) às conexões por meio das quais os valores são transmitidos. Esta operação corresponde a uma

combinação linear das entradas com os pesos associados, mais o limiar ou *bias* b_j , tal como mostra a Equação 14 (CASTRO; ZUBEN, 2003; PALMA NETO; NICOLETTI, 2005).

$$d_j = \sum_{i=0}^n w_{ij}x_i + b_j \quad (14)$$

Em que b_j é chamado *bias* que auxilia na diminuição ou no aumento da influência do vetor de entrada. Ele é principalmente utilizado na rede tipo *Perceptron* (que será abordada em seção posterior) (PALMA NETO; NICOLETTI, 2005).

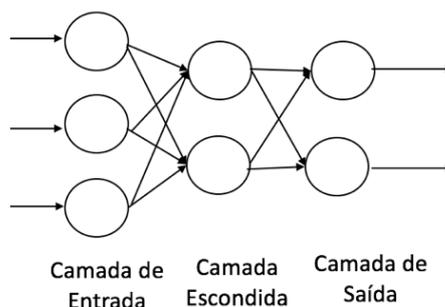
Os pesos podem ter tanto um fator inibitório, quanto um fator excitador do neurônio. Observa-se que se o valor produto $w_{ij}x_i$ é alto tende-se a excitar o neurônio. Ao contrário, se o valor do produto for baixo, o sinal de entrada x_i terá praticamente nenhum efeito (BAUGHMAN; LIU, 1995).

Na fase seguinte é calculado o nível ou estado de atividade do neurônio. Esta operação é feita utilizando uma função de ativação, que pode ser do tipo linear, degrau, degrau-reverso, sigmoide, gaussiana, entre outras. Por fim a função de saída associa a ativação de um neurônio a um valor de saída. Usualmente a função de saída é uma função identidade.

2.5.2.2. Camadas

Uma rede neural é organizada basicamente em três camadas, com unidades que podem estar conectadas às unidades da camada posterior, tal como mostra a Figura 10.

Figura 10 - Três camadas de neurônios - configuração (3-2-2)



Fonte: BAUGHMAN; LIU, 1995

São elas:

- Camada de entrada: formada por neurônios cujas funções soma, ativação e de saída são representadas pela função identidade. Ou seja, cada neurônio dessa camada apenas replica o valor que recebe. Como consequência, para cada entrada da rede neural tem-se um neurônio da camada de entrada a ela associada.
- Camada(s) intermediária(s) ou oculta(s): são as responsáveis pela maior parte do processamento. Através das conexões ponderadas, elas são consideradas como extratores de características.
- Camada de saída: responsável pela apresentação dos resultados finais.

2.5.2.3. Arquitetura da rede neural

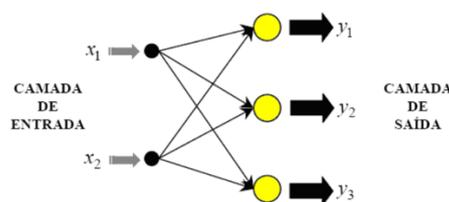
Ainda que pouco é sabido sobre os padrões de conexão entre neurônios biológicos, a maioria das redes neurais artificiais possui uma arquitetura padronizada dedicada à resolução de uma determinada classe de problemas.

Existem basicamente três tipos principais de arquiteturas em redes neurais artificiais:

A. Redes *Feedforward* de uma única camada:

É o tipo mais simples de rede, que consiste de uma camada de entrada e uma camada de saída. Suas principais aplicações se encontram na memória associativa e no reconhecimento de padrões (FINOCCHIO, 2014). Sua estrutura é mostrada na Figura 11.

Figura 11 - Rede *Feedforward* de uma única camada



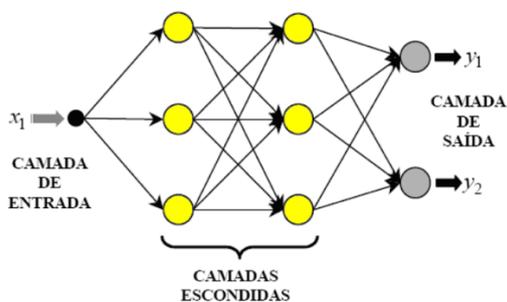
Fonte: FINOCCHIO, 2014

B. Redes *Feedforward* com múltiplas camadas:

As redes de múltiplas camadas possuem uma ou mais camadas intermediárias. Adicionando-se camadas intermediárias não lineares, pode-se aumentar a capacidade de

processamento da rede neural. A saída da camada intermediária é utilizada como entrada para a próxima camada. Suas principais aplicações se encontram no reconhecimento de padrões, atuação como aproximador universal de funções e em controle (FINOCCHIO, 2014). Sua estrutura é mostrada na Figura 12.

Figura 12 - Rede *Feedforward* com múltiplas camadas

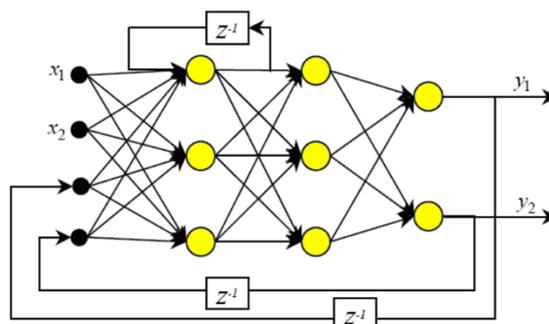


Fonte: FINOCCHIO, 2014

C. Redes Recorrentes

A principal característica deste tipo de rede é a presença de realimentação entre neurônios de camadas diferentes. Suas principais aplicações se encontram em sistemas dinâmicos, memória associativa, previsão e estimação, otimização e em controle (FINOCCHIO, 2014). Sua estrutura é mostrada na Figura 13.

Figura 13 - Redes recorrentes



Fonte: FINOCCHIO, 2014

2.5.2.4. Treinamento de Redes Neurais Artificiais

Uma das características mais importantes das redes neurais artificiais é a habilidade de aprender por intermédio de seu ambiente (através da alimentação de dados reais) e com isso melhorar seu desempenho. Isto é feito através de um processo iterativo de ajustes (treinamento) aplicado a seus parâmetros (pesos e limiar). Este treinamento envolve a alimentação de um conjunto de entradas e saídas conhecidos e o ajuste dos parâmetros da rede até que cada entrada produza uma saída apropriada.

Após o processo de treinamento, a rede neural adquire a sistemática necessária para executar o processamento desejado dos dados fornecidos. Ou seja, é através do algoritmo de treinamento que a rede extrai um conjunto de regras básicas (conjunto de pesos) por meio das quais os pesos das conexões são ajustados. Este conjunto de “exemplos” é chamado de conjunto de treinamento (CASTRO; ZUBEN, 2003; FINOCCHIO, 2014; PALMA NETO; NICOLETTI, 2005).

O processo de aprendizagem encerra-se quando a rede neural consegue generalizar soluções para uma determinada classe de problemas.

Observa-se que os métodos de treinamento de redes não criam novas informações, ou seja, uma rede é reflexo do conjunto de treinamento utilizado para treiná-la.

Finocchio (2014) argumenta que uma porcentagem entre 60 a 90% do total de dados deve ser separada para o treinamento da rede neural. Estes devem ser escolhidos aleatoriamente, e o restante dos dados devem ser apresentados à rede apenas na fase de testes, a fim de verificar se as regras produzem saídas adequadas para os dados não utilizados.

Embora existam diversos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais, a maioria recai, basicamente, em dois grupos (BAUGHMAN; LIU, 1995; CASTRO; ZUBEN, 2003):

A. **Aprendizado supervisionado:** Um agente externo indica à rede um comportamento adequado ou inadequado em conformidade com o padrão de entrada. Ou seja, a descrição de cada exemplo de treinamento traz também a classe (vetor de saída) a ele associada. Neste tipo de aprendizado um vetor de entrada é aplicado, a saída da rede é calculada e comparada com o respectivo vetor de saída desejado. A diferença ou erro é então propagada no sentido contrário ao da informação (da saída para a entrada). Este procedimento iterativo é realizado até que o erro atinja um nível aceitável, determinando-se assim o conjunto de parâmetros da rede neural.

B. Aprendizagem não supervisionada: ao contrário da aprendizagem supervisionada, não existe um agente externo indicando a resposta desejada. Neste tipo de aprendizado a rede é treinada sem considerar a informação da classe associada a cada exemplo de treinamento. A rede se auto-organiza em relação a alguns subconjuntos de entrada cujos elementos possuem características similares. O algoritmo de treinamento então modifica os pesos das redes de forma a produzir vetores de saída consistentes.

2.5.2.5. Redes *Perceptrons*

O *Perceptron* foi proposto por Rosenblatt em 1962 e é considerado o primeiro modelo de redes neurais. Ele é capaz de aprender somente problemas linearmente separáveis. Existem dois tipos de *Perceptrons*: *Perceptrons* de uma camada e de multicamadas. Seu auge se deu através da associação do *Perceptron* multicamadas com o algoritmo de treinamento *Backpropagation* (ou propagação retrógrada). Através dessa combinação foi possível criar um modelo de rede rápido e poderoso para modelagem de dados, sendo considerada até hoje uma das mais importantes redes neurais artificiais (CASTRO; ZUBEN, 2003; FINOCCHIO, 2014; PALMA NETO; NICOLETTI, 2005).

A arquitetura da rede *Perceptron* segue o modelo proposto pela Figura 12, Rede *Feedforward* com múltiplas camadas. Algumas características da rede *Perceptron* multicamadas são:

- A função de ativação de cada um dos neurônios da rede é uma função não linear. Ela deve ser suave e diferenciável em todos os pontos.
- As redes são constituídas de uma ou mais camadas escondidas. Elas permitem que a rede possa aprender tarefas mais complexas progressivamente através do processo de aprendizado.
- A rede *Perceptron* é do tipo *Feedforward*, ou seja, a saída dos neurônios de uma camada se conectam unicamente às entradas da camada seguinte (fluxo de dados se dá apenas em uma única direção).
- A maioria das redes *Perceptron* são completamente conectadas.

Como enunciado anteriormente existem diversos algoritmos de treinamento referentes aos processos de aprendizagem supervisionada. Para a rede *Perceptron* de multicamadas, o

algoritmo de treinamento mais comum é o *backpropagation* (em conjunto com o algoritmo do gradiente descendente).

Este algoritmo consiste basicamente de duas fases distintas. A primeira é chamada de propagação. Um conjunto de padrões é apresentado à camada de entrada da rede. A atividade resultante é propagada pela rede, camada por camada, até que a resposta seja produzida pela camada de saída (durante essa etapa os pesos sinápticos são fixos). A saída obtida é então comparada à saída desejada. Se a saída não estiver correta, o erro é calculado tal como mostra a Equação 15 (BAUGHMAN; LIU, 1995)

$$\epsilon_j = d_j - c_j \quad (15)$$

Em que ϵ_j é o erro da saída, d_j é a saída desejada e c_j é a saída calculada pela rede.

Inicia-se a fase de retro-propagação. O erro é então propagado a partir da camada de saída até a camada de entrada, fazendo com que os pesos das conexões das unidades sejam alterados. Este processo é repetido para todos os vetores de entrada da rede até que o erro quadrático médio das saídas da rede esteja num valor aceitável, tal como mostra a Equação 16.

$$E = \frac{1}{2} \sum_j \epsilon_j^2 = \frac{1}{2} \sum_j (d_j - c_j)^2 \quad (16)$$

Sabendo o valor do erro quadrático pode-se calcular a mudança no fator de pesos Δw_{ij} para a i -ésima conexão ao j -ésimo nó tal como mostra a Equação 17.

$$\Delta w_{ij} = -\eta_j \frac{\partial E}{\partial w_{ij}} \quad (17)$$

Em que η_j é a constante de proporcionalidade do j -ésimo nó, também chamada de taxa de aprendizagem (normalmente $0 < \eta_j \ll 1$).

2.5.2.6. Processo de desenvolvimento de uma rede neural artificial

A sistematização do processo de desenvolvimento de uma rede neural é elucidada a seguir (FINOCCHIO, 2014):

1. Coleta dos dados: nesta etapa tem-se a coleta, cuidadosa, dos dados relativos ao problema. Eles devem ser significativos, cobrir o domínio do problema (sem que haja ambiguidades) e devem abordar as exceções às condições (limites do domínio).
2. Separação em conjuntos: nesta etapa visa-se à separação dos dados coletados em dois conjuntos: treinamento e teste. O primeiro é direcionado ao treinamento da rede neural, e o segundo é direcionado à verificação do desempenho e da capacidade de generalização da rede.
3. Configuração da rede: em geral, esta fase é feita de forma empírica, e compreende três etapas: seleção do paradigma neural apropriado; determinação da topologia da rede (número de camadas e de unidades neuronais); determinação dos parâmetros do algoritmo da rede e das funções de ativação.
4. Treinamento: fase de ajuste dos pesos das conexões e do limiar.
5. Teste e integração: na última etapa, o conjunto de dados de teste é utilizado para determinar a capacidade de generalização da rede criada. Por fim ela deve ser integrada em um sistema de operação (real ou simulado) da aplicação.

2.5.3. Aplicações de redes neurais em bioprocessos

Glassey e colaboradores (1994) descrevem uma estratégia utilizando redes neurais como uma abordagem alternativa para a otimização do bioprocessos. É uma técnica capaz de contornar as não linearidades do processo e ao mesmo tempo pode ser implementada em uma simulação para análise da performance para diferentes entradas. A abordagem utilizando redes neurais se torna atraente na medida em que elas possuem a habilidade de aprender relações não lineares complexas com um conhecimento limitado da estrutura do processo (GLASSEY *et al.*, 1994).

Para a rede escolhida, os autores utilizaram uma função sigmoïdal como função de ativação. Foi utilizado o algoritmo quimiotático (*chemotaxis algorithm*), em que os pesos são ajustados através da adição de valores aleatórios de uma distribuição gaussiana aos pesos anteriores. Os novos pesos serão aceitos se o erro resultante for menor que o erro anterior. Este processo é repetido até que o erro seja minimizado (GLASSEY *et al.*, 1994).

O microrganismo utilizado foi a bactéria *E. coli* K-12 com plasmídeo de uma proteína recombinante utilizando um promotor constitutivo.

O bioprocesso foi realizado em vasos de 30 litros em modo batelada alimentada. Uma porcentagem de oxigênio dissolvido foi fixada através da alimentação de meio suplementar contendo uma quantidade limitada de carbono. Foram coletados 8112 dados de 12 bateladas (GLASSEY *et al.*, 1994).

Na estimativa de dados da biomassa a topologia da rede escolhida foi a 5-2-3-1 (5 neurônios de entrada, 2 na primeira camada escondida, 3 na segunda camada escondida e 1 na camada de saída). Embora tenha-se obtido predições bastante interessantes, alguns problemas ocorreram. Um deles foi a não variação da concentração da biomassa em relação a mudanças na temperatura (o que foi decorrente da falta de dados sob esta condição). As variáveis de entrada escolhidas foram taxa de alimentação, ano da realização da batelada, concentração na batelada, tempo de início da alimentação e temperatura do bioprocesso (GLASSEY *et al.*, 1994).

Na estimativa de dados da proteína recombinante, a topologia da rede escolhida foi a 5-6-1. Embora também se tenha obtido resultados promissores, uma dificuldade encontrada foi a instabilidade do plasmídio (após decorrido o ponto de coleta de amostras), o que resultou numa perda de produtividade. Além disso o mesmo problema com os dados de temperatura também afetou o desempenho da rede neural. As variáveis de entrada escolhidas foram as mesmas utilizadas na estimativa dos dados de biomassa.

Para a estimativa de dados de acúmulo da proteína recombinante a topologia da rede escolhida foi a 5-4-1. Observou-se que a variável que mais influenciou o acúmulo de proteínas recombinantes foi a temperatura do processo (GLASSEY *et al.*, 1994).

Também foi criada uma rede neural para prever a instabilidade das variáveis relacionadas. Foi utilizada uma topologia 4-3-1. Como variáveis de entrada foram utilizadas tempo de início da alimentação, taxa de alimentação, temperatura e ano da realização da batelada. Como variável de saída foi utilizada a variável concentração de células P* (células que sofreram rearranjos estruturais e não produziram a proteína recombinante). Os resultados foram bastante positivos e predisseram com precisão a concentração de células P* (GLASSEY *et al.*, 1994).

Concluiu-se que as redes neurais conseguiram capturar relações não lineares entre as entradas e as saídas do bioprocesso. Os resultados mostraram que elas podem prever concentrações de biomassa e de proteína com níveis de precisão aceitáveis. O esquema de otimização proposto foi testado industrialmente e a comparação entre a predição do modelo e os dados coletados foi promissora (GLASSEY *et al.*, 1994).

Silva e colaboradores (2008) realizaram um estudo para utilização de *soft sensors* para identificar a fase de um cultivo celular. *Soft sensors (software sensors)* são algoritmos matemáticos que podem estimar variáveis não mensuráveis (concentração de substrato e produto) a partir de variáveis mensuráveis (pH, fração molar de oxigênio). Dentre eles, encontra-se a modelagem com redes neurais artificiais. Ela tem ganhado espaço pela sua habilidade de representar relações não lineares complexas sem o conhecimento prévio da estrutura do modelo (SILVA *et al.*, 2008).

Um processo essencialmente complexo e não linear são os cultivos celulares em batelada (ou batelada alimentada). No referido artigo foi analisado o cultivo de *Bacillus megaterium* para produção da enzima penicilina G acilase (PGA).

Os experimentos foram realizados em um biorreator de 5 litros (New Brunswick) operado em modos batelada e batelada alimentada. A temperatura foi mantida a 30 °C, a vazão de ar em 2 l/min e a concentração de oxigênio em 20% de saturação com ar. Os cultivos tiveram duração de 48 horas. A atividade da PGA foi determinada através da hidrólise de 0,2 mol/l de penicilina G a um pH 8,0 e 37 °C e foi determinada por espectrofotometria de massa em 415 nm (SILVA *et al.*, 2008).

O conjunto de dados para treinamento e validação das redes consistiu de seis experimentos em batelada e em batelada alimentada. Um filtro de média móvel foi acoplado para filtrar possíveis ruídos nas medidas antes do treinamento.

Para a identificação das fases, a arquitetura da rede neural utilizada foi a 11-10-3 (11 neurônios na camada de entrada, 10 neurônios na camada escondida e 3 neurônios na camada de saída). As variáveis de entrada escolhidas foram tempo do experimento e a concentração de gás carbônico nos gases de exaustão (valor corrente e mais nove valores anteriores). Como variáveis de saída foram utilizadas três fases características de cultivos, fase *lag*, fase exponencial e fase estacionária. Para cada uma foi atribuído um valor para a saída da rede neural, fase *lag* (1), fase exponencial (2) e fase estacionaria (3) (SILVA *et al.*, 2008).

Para a inferência da concentração celular, a arquitetura da rede neural utilizada foi a 3-10-1. As variáveis de entrada escolhidas foram concentração de gás carbônico, velocidade de agitação e tempo. Como variável de saída foi utilizada a concentração celular.

No caso da rede neural para identificação das fases, os resultados indicaram que o algoritmo foi eficiente. Sobretudo, os autores ressaltam que a grande vantagem desta proposta foi que ao tornar a saída da rede neural números reais contínuos, tem-se a possibilidade de se acoplar a resposta a outros algoritmos. Os resultados da rede neural para estimativa da concentração celular também foram bons. Porém resalta-se que como o conjunto de

treinamento consistiu apenas de dados das fases *lag* e exponencial, não foi possível estimar os dados da fase estacionária. Resultados mostraram que a rede neural seguiu a dinâmica do modelo até o final da fase exponencial. Ao se iniciar a fase estacionária, tem-se um desvio dos resultados do modelo em relação aos dados experimentais. Ressalta-se que na prática, ao fim da fase exponencial o processo seria interrompido para iniciar a coleta da enzima produzida. Assim, é de extrema importância identificar com precisão o momento do término da fase exponencial, uma vez que logo em seguida atinge-se a concentração máxima de enzimas produzidas (SILVA *et al.*, 2008).

Ahmad e colaboradores (2015) investigaram a eficácia de três modelos de redes neurais artificiais para estimar a produção de biossurfactantes pela bactéria *Klebseilla sp.* FKOD36, a redução da tensão superficial e o índice de emulsificação do meio de cultura (AHMAD *et al.*, 2015).

Surfactantes são moléculas que aumentam a solubilidade da classe de químicos hidrofóbicos em meio aquoso. Através da formação de micelas e emulsões, ocorre a suspensão de hidrocarbonetos, solventes e até metais em água. Uma alternativa para otimização do processo de produção reside na utilização das redes neurais artificiais (AHMAD *et al.*, 2015).

Para o experimento foi coletada uma amostra de 5g de solo contaminado com óleo. As bactérias mostraram intensa atividade produtora de biossurfactantes (AHMAD *et al.*, 2015).

A arquitetura das redes utilizadas foi 6-10-1 (6 neurônios na camada de entrada, 10 na camada escondida e 1 na camada de saída). Foram utilizadas como variáveis de entrada, temperatura, pH, tempo de incubação e fontes de carbono, nitrogênio e hidrocarboneto. Como variáveis de saída foram utilizadas produtividade de biossurfactante, índice de emulsificação e redução da tensão superficial (separadamente) (AHMAD *et al.*, 2015).

Os resultados obtidos foram:

- Produtividade de biossurfactante: a melhor estimativa foi obtida para a amostra 23, cujos parâmetros de entrada foram: (40, 7, 2, 3, 1, 3 – respectivamente: temperatura, pH, tempo de incubação e fontes de carbono, nitrogênio e hidrocarboneto). A produtividade aferida foi de 0,03 g/l e a predita pelo modelo foi de 0,038 g/l.
- Índice de emulsificação: a melhor estimativa foi obtida para a amostra 1, cujos parâmetros de entrada foram: (30, 5, 2, 1, 1, 1). O índice de emulsificação aferido foi de 31,67% e o predito pelo modelo foi de 31,68%.
- Redução da tensão superficial: a melhor estimativa foi obtida para a amostra 17, cujos parâmetros de entrada foram: (35, 9, 2, 2, 2, 3). A tensão superficial aferida foi de 21,5 dyna/cm e a predita pelo modelo foi de 21,5 dyna/cm.

Conclui-se que os resultados provenientes da rede neural aumentaram a precisão da estimativa dos três parâmetros desejados (produtividade de biossurfactante, índice de emulsificação e redução da tensão superficial). Além disso, o modelo de rede neural proposto explorou as condições ótimas, tanto do meio quanto do ambiente, para maximizar a produtividade de biossurfactante (AHMAD *et al.*, 2015).

2.6. Ferramentas computacionais

O desenvolvimento do sistema de controlador nebuloso (estabelecimento das variáveis linguísticas, funções de pertinência e regras de controle) pode ser feito através do uso de duas ferramentas computacionais: *Fuzzy Inference System* (FIS) e o *Adaptive Neuro Fuzzy Inference System* (ANFIS). Ambas pertencem ao *Fuzzy Logic Toolbox* do MATLAB. O *Fuzzy Logic Toolbox* provém um conjunto de funções e aplicações em linguagem MATLAB para análise, *design* e simulação de sistemas baseados na lógica nebulosa. Ele permite a modelagem de sistemas de comportamento complexo que se utilizam de regras lógicas simples, para assim implementá-las em um sistema de inferência nebuloso (MATHWORKS, 2016). Já, para a modelagem do processo de cultivo de *E. coli* para produção de proteínas recombinantes, foi utilizado o *Neural Network Time Series* (ou *Time series app*). Este integra o *Neural Network Toolbox*, composto por três outras ferramentas: *Fitting app* (ajuste de curvas de dados de entrada e saída), *Pattern Recognition app* (reconhecimento de padrões e classificação) e *clustering app* (classificação e agrupamento) (BEALE; HAGAN; DEMUTH, 2017).

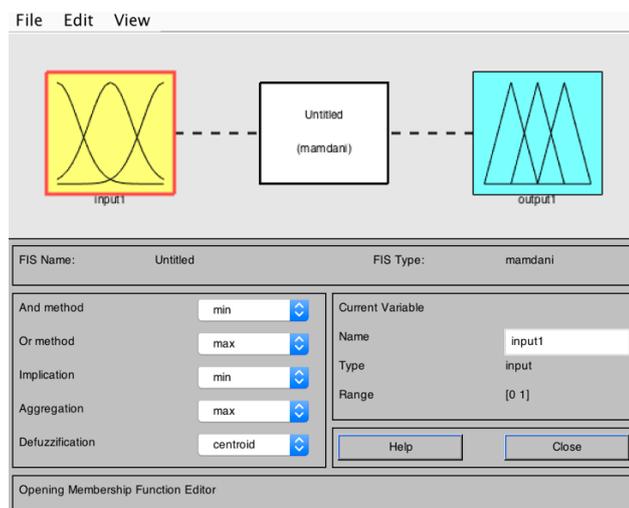
2.6.1. *Fuzzy Inference System* (FIS)

O *Fuzzy inference System* possui ferramentas gráficas para construir, editar e visualizar sistemas de inferência nebulosa. Elas podem ser divididas em dois grupos, um de edição e outro de visualização. O primeiro grupo é composto pelas ferramentas *Fuzzy Logic Designer*, *Membership Function Editor* e *Rule Editor*. O segundo conjunto é composto pelo *Rule Viewer* e o *Surface Viewer*. Elas estão dinamicamente interligadas, ou seja, quaisquer alterações pelas ferramentas de edição serão refletidas nas ferramentas de visualização (MATHWORKS, 2016).

1) *Fuzzy Logic Designer*: lida com questões importantes para o sistema, tais como número de variáveis de entrada e saída, seus nomes, a escolha do método de agregação e

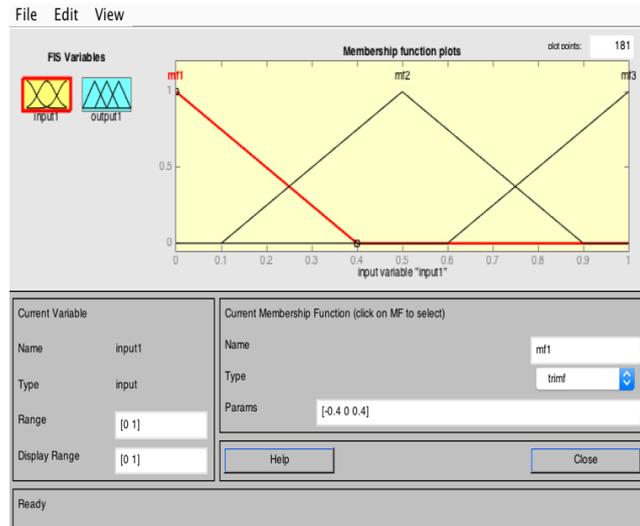
defuzzyficação, e o tipo de sistema nebuloso (Mamdani ou Sugeno). É importante ressaltar que o *Fuzzy Logic Toolbox* não limita o número de entradas. Este está condicionado à memória disponível no hardware (MATHWORKS, 2016). O *Fuzzy Logic Designer* é mostrado na Figura 14 a seguir.

Figura 14 - Interface do *Fuzzy logic Designer* (FIS editor)



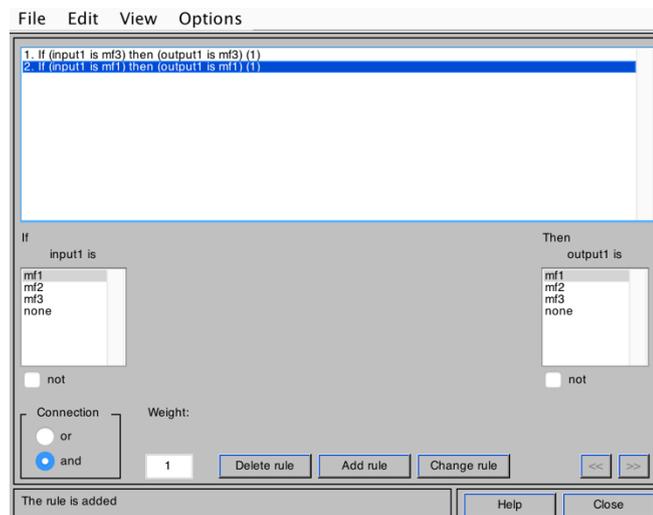
Fonte: MATHWORKS, 2016

2) *Membership Function Editor*: é a ferramenta de edição de funções de pertinência. Ela permite ao usuário incluir, excluir, editar e visualizar todas as funções de pertinência associadas tanto às entradas quanto às saídas. É possível determinar os limites inferior e superior (*range*) das funções de pertinência, nomeá-las, escolher o tipo de função - dentre elas destacam-se as triangulares (*trimf*), trapezoidais (*trapmf*), gaussianas (*gaussmf* e *gaussmf2*), em forma de sino (*gbellmf*), sigmodais (*sigmf*, *dsigmf*, *psigmf*) e polinomiais (*zmf*, *smf*, *pimf*) – e seus parâmetros. O programa também permite ao usuário que ele crie suas próprias funções de pertinência (*custmf*) (MATHWORKS, 2016). O *Membership Function Editor* é mostrado na Figura 15.

Figura 15 - Interface do *Membership Function Editor*

Fonte: MATHWORKS, 2016

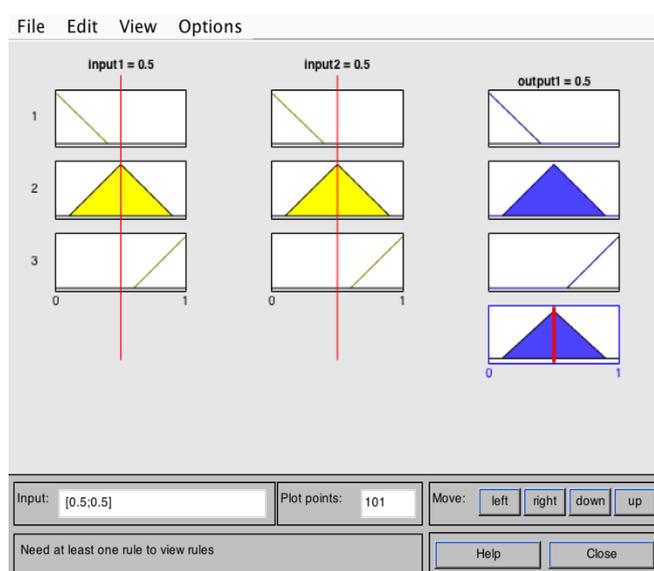
3) *Rule Editor*: é uma ferramenta que lida com a edição, inclusão ou exclusão de regras que determinam o comportamento do sistema. Sua interface é intuitiva e permite ao usuário criar regras (e modificá-las) escolhendo as variáveis nebulosas a partir de uma lista. O editor também possui duas outras opções: “None” e “Not”. A primeira é para excluir determinada variável da regra. A segunda é para negar determinada qualidade (complementar). Também é possível atribuir pesos (valores de 0 a 1) a determinadas regras. Se nada for especificado assume-se que o peso é igual a 1 (MATHWORKS, 2016). O *Rule editor* é mostrado na Figura 16.

Figura 16 - Interface do *Rule editor*

Fonte: MATHWORKS, 2016

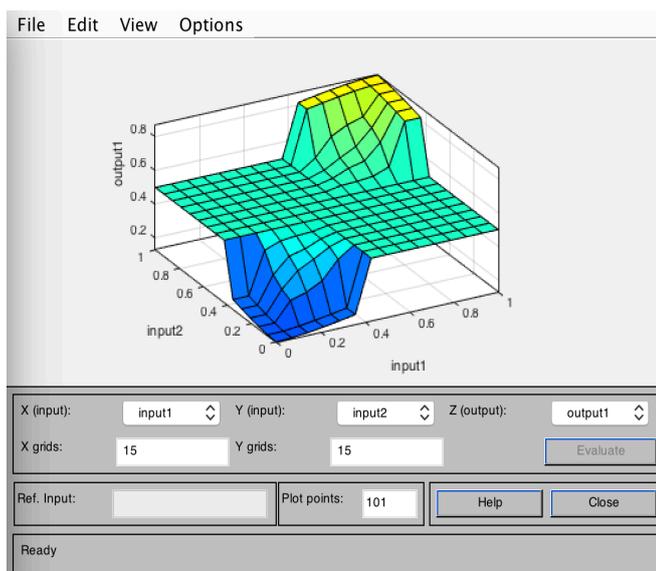
4) *Rule Viewer*: nesta ferramenta cada linha representa uma regra, sendo que a última linha representa o resultado da fase de acumulação. Nas colunas os gráficos amarelos representam as funções de pertinência referentes aos antecedentes. Já os gráficos azuis representam as funções de pertinência referentes aos consequentes. O valor *defuzzyficado* é representado por uma linha vermelha. É uma ferramenta utilizada para diagnóstico. Avalia-se, por exemplo, quais as regras estão ativadas e como a forma específica de função pode afetar o resultado final (MATHWORKS, 2016). O *Rule Viewer* é mostrado na Figura 17.

Figura 17 - Interface do *Rule Viewer*



Fonte: MATHWORKS, 2016

5) *Surface Viewer*: é uma ferramenta utilizada para avaliar a dependência de uma saída com relação a uma ou duas variáveis de entrada, através de gráficos de superfície. Os eixos x e y referem-se às entradas. Já o eixo z refere-se à saída. A ferramenta permite o ajuste do número de pontos a serem plotados (no formato padrão são utilizados 101 pontos) (MATHWORKS, 2016). O *Surface Viewer* é mostrado na Figura 18.

Figura 18 - Interface do *Surface Viewer*

Fonte: MATHWORKS, 2016

2.6.2. *Adaptive Neuro Fuzzy Inference System (ANFIS)*

Como descrito anteriormente, no caso do FIS, a forma das funções de pertinência e seus parâmetros são definidos pelo próprio usuário. Porém, quando não é possível discernir a forma e os parâmetros das funções de pertinência, simplesmente observando os dados, utiliza-se a ferramenta ANFIS (*Adaptive Neuro Fuzzy Inference System*), pertencente ao *Fuzzy Logic Toolbox*. Ela realiza esta escolha automaticamente, utilizando um conjunto de funções, a fim de construir um sistema de inferência nebuloso a partir de dados de entrada e saída.

Sua estrutura é muito similar a uma rede neural. O software computa os parâmetros das funções de pertinência que permitem o mapeamento das entradas com as saídas através do sistema de inferência nebuloso. Os ajustes nos valores dos parâmetros permitem ao sistema nebuloso aprender com os dados. Esta é a principal característica do ANFIS.

Os ajustes nos parâmetros das funções de pertinência são feitos através de algoritmos de propagação retrógrada do erro ou *backpropagation* (por exemplo, gradiente descendente), ou um algoritmo híbrido de *backpropagation* (associado aos parâmetros da função de pertinência de entrada) com um método de mínimos quadrados (associados aos parâmetros da função de pertinência de saída) (MATHWORKS, 2016).

A abordagem de modelagem do ANFIS é similar às técnicas de identificação de sistemas. Primeiro hipotetiza-se a estrutura de um modelo (relacionando entradas e saídas às funções de pertinência e construindo regras nebulosas). Em seguida treina-se o ANFIS com

dados de entrada e saída, para que possa gerar um FIS que emule os dados treinados. Em geral, essa técnica de modelagem funciona bem se o conjunto de treino é representativo das características dos dados que o FIS almeja modelar. Por fim, é necessária uma validação do modelo. Podem ser utilizados para este fim conjuntos de dados de teste ou de checagem. A utilização de conjunto de teste para validação do modelo é feita apresentando-se o vetor de entrada com dados de entrada e saída não utilizados para treinar o FIS ao modelo treinado, e assim verificar “o quão bem” o modelo FIS prevê o correspondente conjunto de valores de saída (capacidade de generalização). Já a utilização de conjunto de checagem prevê que conforme aumenta-se o número de treinos, diminui-se o erro. Até um ponto em que ocorre um sobreajuste (*overfitting*) do modelo aos dados, e o erro do modelo começa a aumentar.

A ferramenta *Neuro Fuzzy Designer* do ANFIS inclui 4 áreas distintas (conforme detalhado a seguir), que permitem ao usuário carregar, plotar ou limpar dados, gerar ou carregar uma estrutura FIS inicial, treinar um FIS e validar um FIS treinado (MATHWORKS, 2016).

1) Carregar, plotar ou limpar dados: para treinar um FIS é necessário carregar um conjunto de treinamento, que contém os dados de entrada e saída do sistema a ser modelado. Neste arquivo em formato de texto, cada linha é um par de dados de entrada e saída. Assim, tem-se um número de linhas igual ao número de pares de entrada e saída. Como o ANFIS possui apenas uma única saída, o número de colunas no arquivo é igual ao número de entradas acrescido de um. Assim, qualquer conjunto de dados a ser carregado deve estar em forma de vetor coluna, e os dados de saída devem estar na última coluna. Depois de carregar os dados eles são plotados num gráfico em que o eixo y representa a saída e o eixo x representa a posição dos dados (linha em que se localiza o determinado dado). O conjunto de treinamento é utilizado para treinar um sistema nebuloso através do ajuste de parâmetros das funções de pertinência que melhor modelem o conjunto de dados. Também é possível carregar conjuntos de dados de teste e de checagem (MATHWORKS, 2016).

2) Gerar ou carregar uma estrutura FIS inicial: após o carregamento dos dados o próximo passo consiste em determinar uma estrutura FIS. Ela pode ser gerada automaticamente pelo *Neuro Fuzzy Designer* pelos métodos *grid partition* ou *clustering* ou pode ser carregada pelo usuário. No primeiro caso, o *grid partition* é o método padrão. Ele permite ao usuário escolher a quantidade e o tipo das funções de pertinência. Observa-se que as funções de pertinência de saída devem ser todas lineares ou todas constantes. No segundo caso a estrutura

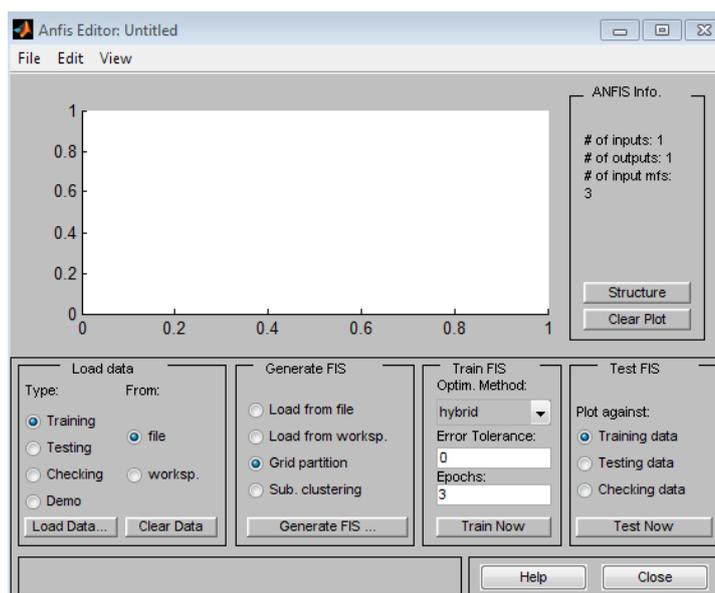
FIS é carregada já pré-estabelecida pelo usuário para otimização dos parâmetros pelo ANFIS (MATHWORKS, 2016).

3) Treinar um FIS: como descrito anteriormente, há duas opções de métodos de otimização paramétricas disponíveis no ANFIS: propagação retrógrada e híbrido (propagação retrógrada e mínimos quadrados). É possível determinar o critério de parada através da tolerância do erro, baseado no tamanho do mesmo. Também é possível escolher o número de iterações (*epochs*) a serem feitas pelo ANFIS.

4) Testar um FIS treinado: para se testar um FIS treinado tem-se três opções: utilizando um conjunto de checagem, um conjunto de teste ou um conjunto de treinamento (MATHWORKS, 2016).

A interface do *Neuro Fuzzy Designer* é mostrada na Figura 19.

Figura 19 - ANFIS editor



Fonte: MATHWORKS, 2016

2.6.3. Neural Network Time Series

As Redes Neurais Artificiais podem ser divididas em duas categorias: estáticas e dinâmicas. Redes neurais estáticas (tipo *feedforward*) não tem elementos que caracterizam uma retro-alimentação dos dados e não contém *delays* (atrasos). A saída é calculada a partir da

entrada diretamente através de conexões do tipo *feedforward*. Ao passo que as redes neurais dinâmicas a saída depende não só dos valores atuais de entrada, mas também de seus valores anteriores, assim como os valores de saída anteriores (BEALE; HAGAN; DEMUTH, 2017). Para o propósito desta dissertação foram utilizadas as redes neurais dinâmicas, haja vista o objetivo de modelagem dinâmica do processo.

O *Time Series app* é composto por 8 interfaces que, passo a passo, trabalham desde a concepção da rede neural desejada até a implementação da mesma na ferramenta *Simulink*.

2.6.3.1. Interface Inicial e escolha de dados

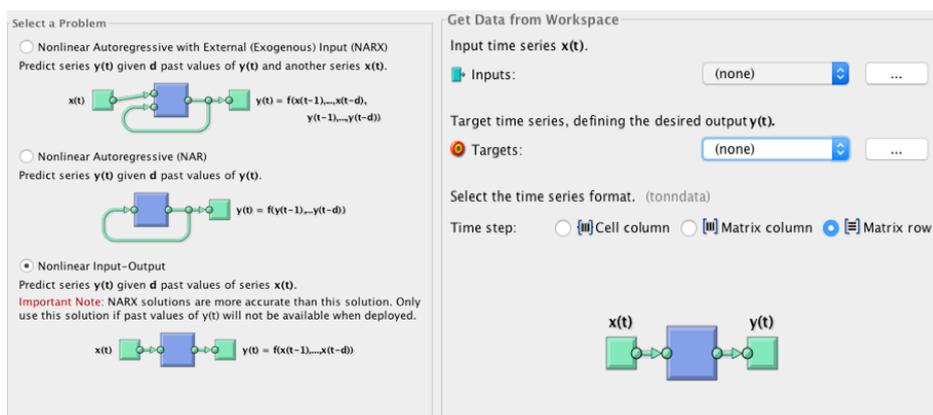
Na interface inicial é possível escolher dentre três opções de problemas (BEALE; HAGAN; DEMUTH, 2017):

- Problema não linear autorregressivo com entrada externa (*NARX*): Neste tipo de problema são preditos valores da série $y(t)$ dados “d” valores anteriores de $y(t)$ (atrasos) e uma outra série $x(t)$.
- Problema não linear Autoregressivo (*NAR*): Neste tipo de problema são preditos valores da série $y(t)$ dados “d” valores anteriores de $y(t)$ (atrasos).
- Problema não linear entrada-saída: Neste tipo de problema são preditos valores da série $y(t)$ dados “d” valores anteriores de uma série $x(y)$ (atrasos).

Na segunda interface são escolhidos os dados de entrada (*Inputs*) e os dados de saída (*Outputs*), que seriam os dados almejados.

As duas interfaces são mostradas na Figura 20.

Figura 20 - Interface inicial (esquerda) e escolha dos dados de entrada e saída do *Time series app* (direita)



Fonte: BEALE; HAGAN; DEMUTH, 2017

2.6.3.2. Separação dos dados e escolha da arquitetura da rede

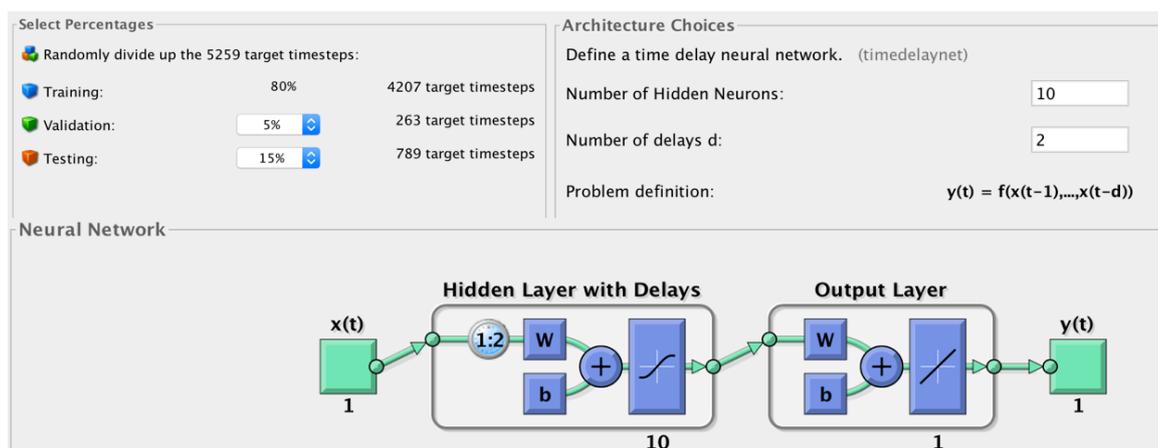
Após a escolha dos dados o aplicativo requer a escolha da porcentagem de dados a serem utilizadas nas três fases: treinamento, validação e testes. Na primeira fase os dados serão utilizados para treinamento da rede neural. Na segunda fase uma parte do conjunto de dados separada será utilizada para medir a capacidade de generalização da rede neural. Por fim, na última fase, a última parte do conjunto de dados será utilizada para mensurar a performance da rede, durante e após o treinamento. Embora o usuário possa definir a porcentagem de dados de cada fase, a escolha propriamente dos dados é feita de forma aleatória pelo programa. A teoria recomenda manter pelo menos cerca de 70% dos dados para treinamento e o restante dividido entre validação e teste.

A seguir é escolhida a arquitetura da rede. Porém só é possível escolher a quantidade de neurônios na camada escondida e o número de atrasos. Não é possível adicionar mais camadas escondidas (BEALE; HAGAN; DEMUTH, 2017).

Um resumo da arquitetura da rede é mostrado ao fim deste processo.

A Figura 21 mostra as interfaces previamente referidas.

Figura 21 - Da esquerda para a direita tem-se a interface de escolha da divisão dos dados; escolha dos parâmetros da arquitetura da rede e por fim um breve resumo da rede.



Fonte: BEALE; HAGAN; DEMUTH, 2017

2.6.3.3. Treinamento da rede neural

Na interface de treinamento da rede neural é possível escolher entre três algoritmos de treinamento: Levenberg-Marquardt, Regularização Bayesiana e Gradiente Escalar Conjugado. Em geral utiliza-se o algoritmo de Levenberg-Marquardt, em que o treinamento termina quando a generalização não mostra sinais de melhora.

Após o treinamento é possível checar os valores de Regressão (R), que mensura o valor da correlação entre o valor da saída da rede e o valor de saída desejado. Um valor de R igual a 1 indica uma correspondência próxima. Um valor zero indica uma relação aleatória (BEALE; HAGAN; DEMUTH, 2017).

Na interface de resultados do treinamento da rede é possível plotar uma série de gráficos que podem auxiliar na avaliação da rede treinada. Tem-se por exemplo:

- Regressão: plota-se os gráficos da saída da rede em função dos valores de saída desejados, bem como a equação linear da correlação.
- Resposta: são plotados os valores de saída (para as três fases) do modelo neural bem como os valores de saída desejados em função do número de passos fornecidos (*timesteps*).

A Figura 22 mostra as interfaces previamente referidas.

Figura 22 - Da esquerda para a direita tem-se a interface de escolha do método de treinamento da rede; resultados do treinamento da rede e por fim a escolha dos gráficos que podem ser plotados para auxiliar na avaliação da rede treinada.

Train Network

Choose a training algorithm:

- Levenberg-Marquardt
- Bayesian Regularization
- Scaled Conjugate Gradient

Training automatically stops improving, as indicated by an increase in the mean square error of the validation samples.

Train using Levenberg-Marquardt. (trainlm)

Results

	Target Values	MSE	R
Training:	4207	8.00158e-3	9.99997e-1
Validation:	263	5.31290e-4	9.99999e-1
Testing:	789	7.41913e-3	9.99998e-1

Plots

- (plotperform)
- (plottrainstate)
- (ploterrhist)
- (plotregression)
- (plotresponse)
- (ploterrcorr)
- (plotinerrcorr)

Fonte: BEALE; HAGAN; DEMUTH, 2017

2.6.3.4. Avaliação da rede neural

Nesta interface é possível tentar melhorar o desempenho da rede neural se o mesmo está ainda aquém do planejado. É possível re-treinar a rede, mudar a arquitetura da rede, importar um conjunto maior de treinamento e realizar um teste com outros dados de entrada e de saída para a mesma rede, tal como mostra a Figura 23 (BEALE; HAGAN; DEMUTH, 2017).

Figura 23 - Interface de ajuste da rede neural treinada.

Iterate for improved performance

Try training again if a first try did not generate good results or you require marginal improvement.

Increase network size if retraining did not help.

Not working? You may need to use a larger data set.

Optionally perform additional tests

Inputs: (none) ...

Targets: (none) ...

Time step: Cell column Matrix column Matrix row

No inputs selected.

No targets selected.

MSE

R

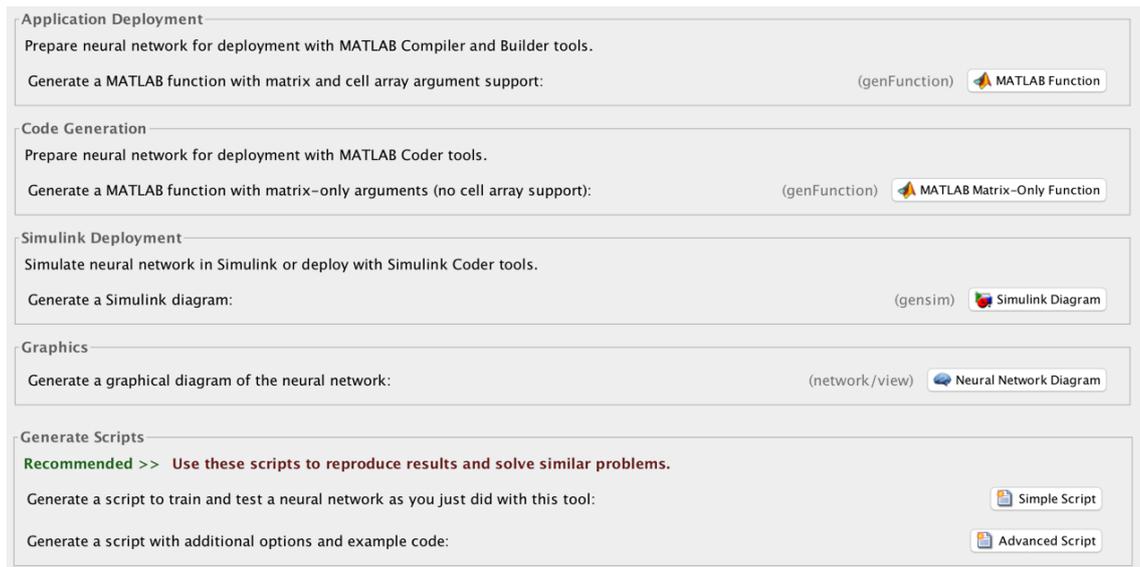
Fonte: BEALE; HAGAN; DEMUTH, 2017

2.6.3.5. Implementação da rede neural

Nas duas últimas interfaces é possível implementar a rede neural treinada através de uma função no MATLAB, através de um diagrama no SIMULINK ou estabelecer um roteiro (*Script*) para treinar e testar futuras redes neurais.

Por fim é possível salvar os dados relativos às fases de treinamento, validação e testes. A Figura 24 mostra as interfaces descritas previamente.

Figura 24 - Interface para exportar e implementar a rede neural treinada



Fonte: BEALE; HAGAN; DEMUTH, 2017

3. METODOLOGIA

Neste capítulo são detalhadas as condições em que foram obtidos os dados de cultivos de *E. coli* que serviram de base para o desenvolvimento deste trabalho. Também são apresentadas as metodologias utilizadas para o desenvolvimento do controlador nebuloso e implementação da rede neural.

3.1. Dados experimentais

Os dados de controle de cultivo de *E.coli* modificada para expressão de proteínas recombinantes utilizados para o desenvolvimento do controlador nebuloso foram gentilmente cedidos pelo Prof. Dr. Antonio Carlos Luperni Horta (HORTA, 2011). O cultivo A16 foi utilizado para a *fuzzyficação* da árvore de decisão inicialmente proposta para manipulação das vazões de ar e oxigênio e, posteriormente, para o treinamento das redes neurais que compõem o simulador do processo. Já o cultivo A12 foi utilizado para retreinar as redes neurais e, assim, permitir a realização de teste final de robustez do sistema de controle nebuloso (isto é, para verificar se o sistema nebuloso obtido pela *fuzzyficação* de dados de controle do cultivo A16 seria capaz de controlar um processo símile, representado pelo modelo neural referente ao cultivo A12).

3.1.1. Microrganismo

Dois dos 22 cultivos realizados utilizando o SUPERSYS_HCDC foram os cultivos A16 e A12. Para ambos utilizou-se como microrganismo *E. coli* recombinante BL21(DE3) modificada com o plasmídeo PspA3/pET37b+. A clonagem de um fragmento do gene PspA3 foi realizada pela Dra. Eliane Miyaji do centro de Biotecnologia do Instituto Butatan, no vetor pET37b+. As cepas expressavam um fragmento da proteína da membrana do *S. Pneumoniae* (HORTA 2011).

3.1.2. Cultivo

Os cultivos foram conduzidos em batelada alimentada em biorreator de 5 l construído no próprio laboratório e supervisionados pelo SUPERSYS_HCDC. O pH foi controlado em 6,7

(pHmetro GLI PRO), na fase batelada e a 6,9 na fase de alimentação, pela adição de hidróxido de amônio (30%). A temperatura foi controlada em 35°C. A concentração de oxigênio dissolvido foi monitorada utilizando um sensor de oxigênio dissolvido (Mettler Toledo Inpro 6800) e mantida a 30% do valor de saturação com ar a 35°C por um controlador híbrido (PID + árvore de decisão) que automaticamente variava tanto a frequência de agitação (de 200 a 900 rpm), através de um inversor de frequência, quanto a composição da corrente de gás que alimentava o biorreator (mistura de oxigênio puro e ar). A vazão total de gás foi mantida numa faixa de 3 a 6 l/min por dois fluxômetros de massa (GFC AALBORG). A permissividade e a condutividade foram monitoradas por um sensor de biomassa (FOGALE Nanotech). A composição do gás de exaustão foi medida por um analisador de oxigênio e gás carbônico (Sick/Maihak) (HORTA 2011).

Tanto a aquisição de dados *on line* quanto o monitoramento e controle de todos os instrumentos foram feitos utilizando o *software* SUPERSYS_HCDC implementado em LABVIEW e MATLAB. Foi utilizado o *Compact Field Point (National Instruments)* para conectar os instrumentos de aferição ao computador. Além disso foram utilizados também uma bomba de alimentação (ISMATEC), um banho térmico e bombas de ácido e base (HORTA 2011).

Para o cultivo A16, foi possível obter uma concentração celular máxima de 155 g/l em 20,5 horas de cultivo. Já para o cultivo A12 foi possível obter uma concentração celular máxima de 123 g/l em 20,75 horas de cultivo.

Os resultados experimentais (controlados utilizando a árvore de decisão) dos cultivos utilizados para o desenvolvimento do controlador nebuloso e treinamento das redes neurais (A16) e para teste de robustez do controle nebuloso (A12) podem ser encontrados nos anexos A e B (respectivamente).

3.2. Desenvolvimento do controlador nebuloso

Como elucidado na revisão bibliográfica, o controlador nebuloso desenvolvido por Sousa e Sousa (2015) foi ajustado de forma heurística, utilizando dados de cultivo disponíveis. Como contribuição desta dissertação foi utilizada a ferramenta ANFIS (*Adaptive Neuro Fuzzy Inference System*), para obter os parâmetros ótimos das funções de pertinência. Porém foram necessárias modificações acerca do sistema de inferência nebuloso utilizado por Sousa e Sousa (2015).

3.2.1. Adaptações do sistema de inferência nebuloso

Observando-se a arquitetura do ANFIS nota-se a necessidade de algumas modificações no sistema de inferência nebuloso desenvolvido por Sousa e Sousa (2015). Primeiramente a arquitetura do ANFIS só permite a utilização de sistemas de inferência nebulosos do tipo Sugeno (saída do sistema de inferência é uma função). Porém os sistemas nebulosos utilizados por Sousa e Sousa (2015) eram do tipo Mandani (saída do sistema de inferência é um conjunto nebuloso). Assim as variáveis de saída das sub-rotinas “Delta” e “Princ” deixaram de ser conjuntos nebulosos (perdendo também seus respectivos valores linguísticos) e passaram a ser funções lineares, cujos parâmetros foram determinados pelo próprio ANFIS.

Outra modificação necessária foi o desmembramento da sub-rotina “Princ” que possuía três variáveis de saída (QarEscolha, QO₂Escolha e Qmáx). Esta mudança se deveu à arquitetura do ANFIS, que limita o sistema de inferência nebuloso a apenas uma variável de saída. A princípio a sub-rotina “Princ” seria desmembrada em 3 sub-rotinas, porém após observado que a vazão máxima do sistema (Qmáx) permanecia praticamente constante, optou-se por suprimi-la como variável de saída, mantendo-a como um valor constante de 7 l/min. Assim a sub-rotina “Princ” foi desmembrada em duas sub-rotinas, uma sub-rotina “Princ1”, cuja variável de saída era QarEscolha e uma sub-rotina “Princ2” cuja variável de saída era QO₂Escolha.

Por fim, após uma revisão da literatura, optou-se pela inserção da velocidade específica de crescimento (μ) como variável de entrada do sistema de inferência nebuloso. A velocidade específica de crescimento influencia a velocidade de consumo de oxigênio pelo microrganismo, uma vez que a velocidade específica de consumo de oxigênio é função da velocidade específica de crescimento. Assim as sub-rotinas “Princ1” e “Princ2” foram atualizadas com uma nova variável de entrada: μ , com 2 valores linguísticos, “BP” (Baixo e Positivo) e “MP” (Médio e Positivo).

O Quadro 9 mostra as novas sub-rotinas “Delta”, “Princ1” e “Princ2” com suas variáveis de entrada (juntamente com seus valores linguísticos) e de saída.

Quadro 9 - Variáveis de entrada e saída das novas sub-rotinas e seus respectivos valores linguísticos.

Sub-rotinas	Variáveis de entrada	Valores linguísticos	Variáveis de saída
“Delta”	QO ₂ ant: vazão inicial de oxigênio (l/min)	BP: baixa e positiva MP: média e positiva AP: alta e positiva	Delta
“Princ1”	CX: concentração de microrganismos (g/l)	BP: baixa e positiva AP: alta e positiva	QarEscolha
	AgitMed: agitação (rpm)	BP: baixa e positiva AP: alta e positiva MAP: Muito alta e positiva	
	OD: Oxigênio dissolvido (%)	BP: baixo e positivo MP: médio e positivo AP: alto e positivo	
	QarSe: variável auxiliar	NÃO SIM	
	Mi: velocidade específica de crescimento (1/h)	BP: baixo e positivo MP: médio e positivo	
“Princ2”	CX: concentração de microrganismos (g/l)	BP: baixa e positiva AP: alta e positiva	QO ₂ Escolha
	AgitMed: agitação (rpm)	BP: baixa e positiva AP: alta e positiva MAP: Muito alta e positiva	
	OD: Oxigênio dissolvido (%)	BP: baixo e positivo MP: médio e positivo AP: alto e positivo	
	QarSe: variável auxiliar	NÃO SIM	
	Mi: velocidade específica de crescimento (1/h)	BP: baixo e positivo MP: médio e positivo	

Fonte: Acervo Pessoal

3.2.2. Inserção dos dados na ferramenta ANFIS

Na ferramenta ANFIS o mapeamento das entradas com as saídas através do sistema de inferência nebuloso se dá por meio da utilização de dados de entrada e de saída do sistema. Porém, observou-se que na rotina criada por Sousa e Sousa (2015), a saída do sistema de inferência nebuloso não era a vazão de ar ou a vazão de oxigênio diretamente. Tratava-se de uma variável auxiliar (QarEscolha ou QO₂Escolha, respectivamente). Elas são faixas de valores

que mimetizam as opções fornecidas pela árvore de decisão (o trecho da rotina em que se encontram as faixas de valores está ilustrado no Quadro 10). Observa-se que no caso da vazão de ar, tem-se que se QarEscolha for maior ou igual, por exemplo, a 0,595, soma-se um valor DeltaP à vazão de ar calculada no passo anterior (QarantP). Caso contrário, diminui-se um valor delta da vazão de ar calculada no passo anterior. O mesmo raciocínio aplica-se para QO₂Escolha, porém com três faixas de valores.

Logo não seria possível utilizar diretamente os dados das vazões de ar e oxigênio fornecidos pela árvore de decisão do SUPERSYS_HCDC. Isto fez com que fosse necessária a aplicação de um raciocínio reverso detalhado nas subseções subsequentes.

Quadro 10 - Faixas de QarEscolha e QO₂Escolha na rotina em MATLAB.

```

b = readfis('princ1S_red5');
QarEscolha = evalfis([CXP AgitMedP ODP QarSeP mi],b);

if QarEscolha >=0.595;
    QarP = QarantP+1*DeltaP;
end

if QarEscolha < 0.595;
    QarP=QarantP-1.*DeltaP;
end

c = readfis('Princ2S_red13');
QO2Escolha = evalfis([CXP AgitMedP ODP QarSeP mi],c);

if QO2Escolha >=0.63;
    QO2P = QO2antP+1.*DeltaP;
end

if QO2Escolha <=0.28;
    QO2P = QO2antP-1.*DeltaP;
end

if QO2Escolha > 0.28;
    if QO2Escolha <0.63;
        QO2P = QO2antP;
    End
end

```

Fonte: Acervo Pessoal

3.2.2.1. Ajuste do sistema de inferência nebuloso Delta

Para o ajuste de Delta foram necessários apenas dois conjuntos de dados, as vazões de oxigênio e os valores de Delta. O primeiro foi adquirido de forma direta, utilizando os dados de vazão de oxigênio fornecidos pela árvore de decisão. Já para o segundo conjunto, como Delta é uma variação da vazão de ar e/ou oxigênio, foi feito o seguinte procedimento:

- I. De posse dos valores das vazões de ar e/ou oxigênio que seriam os valores fornecidos pela árvore de decisão do SUPERSYS_HCDC, criou-se duas colunas de dados. A primeira coluna foi formada pelos dados de vazão de ar ou oxigênio, com exceção do último valor, denominados QarantP/QO₂antP, representando as vazões anteriores. A segunda coluna foi formada deslocando-se a primeira em um passo, ou seja, englobando do segundo valor de vazão até o último. Esta foi chamada de QarP/QO₂P, representando as vazões correntes.
- II. Delta é o valor absoluto da subtração de uma das colunas pela outra coluna. Destaca-se que esta operação só é possível pois Delta é um número pertencente ao conjunto dos naturais.
- III. De posse dos valores de entrada e saída criou-se um arquivo de dados no formato requerido pelo ANFIS.
- IV. Por fim ajustou-se o número de valores linguísticos das variáveis linguísticas de entrada (tal como disposto no Quadro 9) para criação do sistema de inferência nebuloso. Foi utilizada uma função de pertinência da forma Gaussiana 2 (gauss2mf), cuja equação é dada pela Equação 18:

$$f(x, \sigma, c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \quad (18)$$

Esta função é uma combinação de 2 parâmetros. A primeira função, especificada por σ_1 e c_1 , determina a forma da parte esquerda da dependência funcional. Já a segunda função, especificada por σ_2 e c_2 , determina a forma da parte direita da dependência funcional.

- V. Procedeu-se a fase de treinamento do sistema de inferência nebuloso.

3.2.2.2. Preparação do conjunto de variáveis de entrada e saída dos sistemas de inferência nebulosos Princ1 e Princ2

Para o ajuste dos sistemas de inferência nebulosos Princ1 e Princ2 foi necessária uma fase de preparação dos dados a serem utilizados. De posse de dados experimentais de concentração celular (CX), velocidade média de agitação (AgitMed) e porcentagem de oxigênio

dissolvido (OD) ainda restavam os valores de velocidade específica de crescimento (μ) e da variável auxiliar QarSe.

Para o primeiro conjunto de dados (A16) observou-se que os valores de velocidade específica de crescimento (*on line*) disponibilizados por Horta (2011) continham alguns valores negativos. Assim foi aplicada uma suavização tal como descrito pela Equação 19.

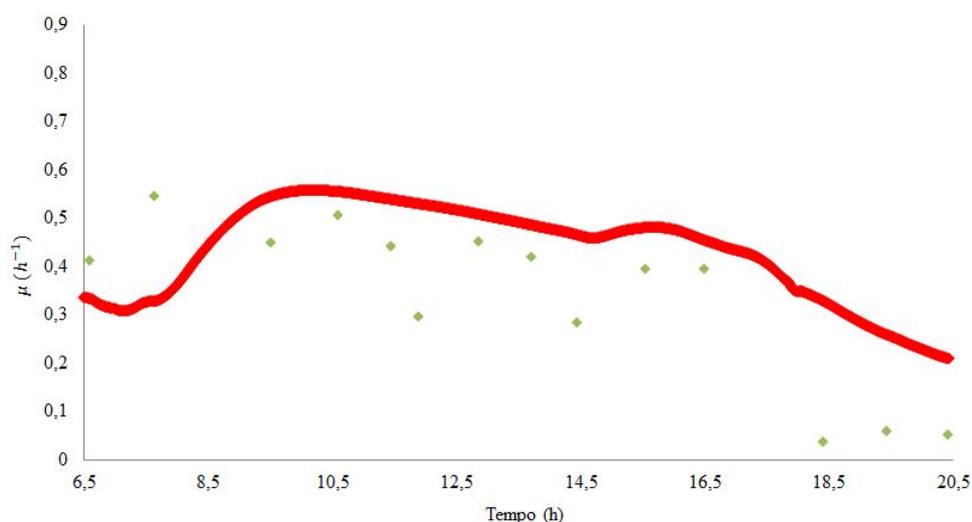
$$S_t = \alpha X_t + (1 - \alpha) * S_{t-1} \quad (19)$$

Em que S é o valor suavizado da variável, α é o grau de suavização, X são os valores de entrada da sequência e t é referente ao tempo.

Para realizar a suavização exponencial foram suprimidos os primeiros 1700 pontos, ou seja, diminui-se o conjunto de dados disponíveis de 6959 para 5259 pontos. O valor de α utilizado foi de 0,00075.

Na Figura 25 tem-se o gráfico da velocidade específica de crescimento experimental (pontos em verde, por medidas de CX em massa seca) e a suavizada (pontos em vermelho, por suavização das medidas por permissividade) em função do tempo.

Figura 25 - Velocidade específica de crescimento experimental e a suavizada em função do tempo.



Fonte: Acervo Pessoal

A fase de indução foi iniciada a partir de 16,6 h, quando se tem uma redução de μ .

Por fim, utilizou-se um excerto da rotina já implementada no MATLAB por Sousa e Sousa (2015) para o cálculo de QarSe com base nos valores de QO₂antP, QarantP e QmáxP tal como mostra o Quadro 11.

Quadro 11 - Excerto da rotina em MATLAB para o cálculo de QarSe.

```
QarConta = QmaxP - QO2antP;  
  
if QarantP >= QarConta;  
QarSeP = 0.9;  
end  
  
if QarantP < QarConta;  
QarSeP = 0.1;  
end
```

Fonte: SOUSA; SOUSA (2015)

3.2.2.3. Ajuste dos sistemas de inferência nebulosos Princ1 e Princ2

Procedendo-se para a última fase, o ajuste dos sistemas de inferência nebulosos Princ1 e Princ2 se deu através das seguintes etapas.

- I. De posse dos valores das vazões de ar e/ou oxigênio novamente criou-se duas colunas de dados (a primeira dispondo os valores de QarantP/QO₂antP e a segunda os valores de QarP e QO₂P), seguindo o mesmo procedimento descrito na seção 3.2.2.1 para o ajuste do sistema de inferência nebuloso Delta.
- II. Para QarEscolha e QO₂Escolha, foi observado que não seria possível dispor de seus valores diretamente. Então, comparou-se o valor de QarP/QO₂P com os valores de QarantP/QO₂antP para definir qual seria o valor de QarEscolha/QO₂Escolha. Foram utilizados valores extremos para QarEscolha e QO₂Escolha, para que o ANFIS pudesse distinguir as diferentes condições que foram selecionadas pelo controle baseado na árvore de decisão. Para QarEscolha foram utilizados valores de 0,1 e 1,2 e para QO₂Escolha foram utilizados valores de 0,1, 0,55 e 1.
- III. De posse dos valores de entrada e saída criou-se um arquivo de dados no formato requerido pelo ANFIS.
- IV. Por fim ajustou-se o número de valores linguísticos das variáveis linguísticas de entrada (tal como disposto no Quadro 1) para criação do sistema de inferência nebuloso. Foi utilizada a função de pertinência do tipo Gauss (gaussmf) cuja equação é dada pela Equação 13.
- V. Procedeu-se para a fase de treinamento do sistema de inferência nebuloso.

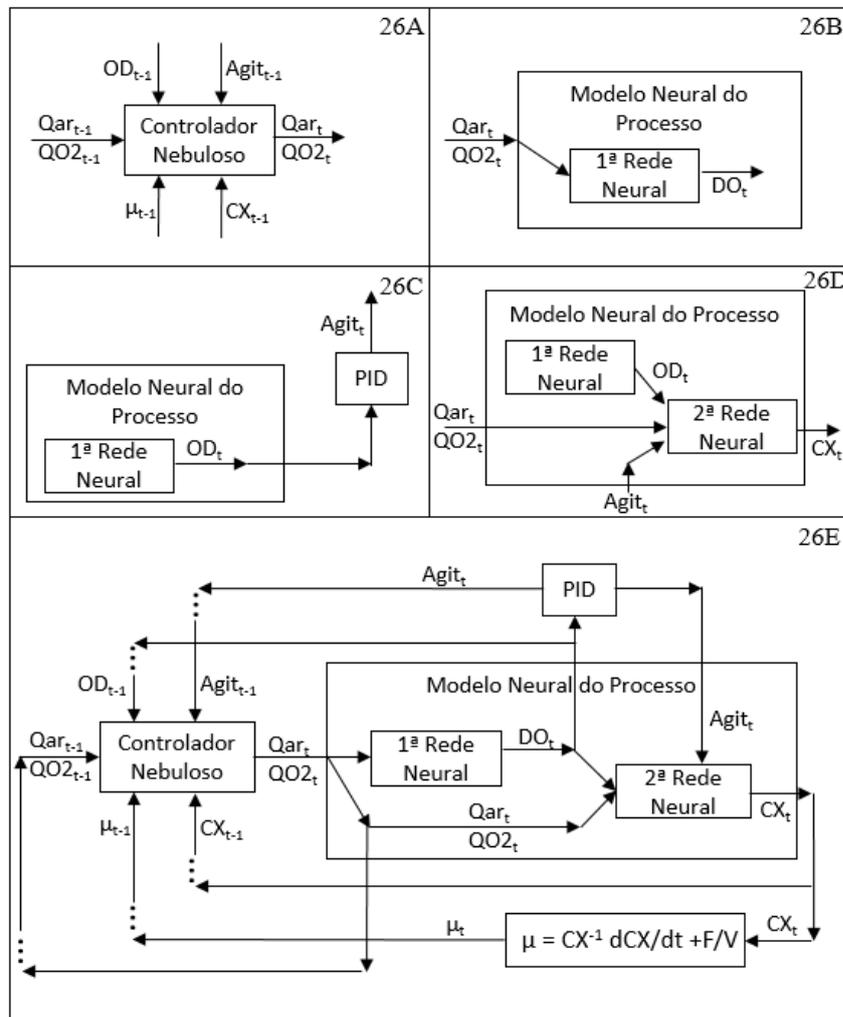
3.3. Simulador

Para testar o novo controlador nebuloso foi criado um simulador para a resposta dinâmica do processo. Para isso foi utilizada a ferramenta de redes neurais do MATLAB.

3.3.1. Planejamento do simulador

Inicialmente, foi realizado um planejamento para acoplar o controlador nebuloso com a rede neural tal como detalhado na Figura 26.

Figura 26 - Diagrama do simulador pretendido.



Fonte: Acervo Pessoal

Acoplou-se o controlador nebuloso ao modelo dinâmico neural do processo e ao controle PID da agitação. O controle nebuloso é o “motor” do simulador, ou seja, a partir de

valores anteriores de CX, OD, AgitMed, Qarant, QO₂ant e μ , o controle fornece os novos valores das vazões de ar e oxigênio (Figura 26A). Estes são repassados ao modelo neural do processo (formado por duas redes neurais acopladas em série). A primeira rede neural (Rede 1) fornece a porcentagem de oxigênio dissolvido a partir dos valores das vazões de ar e oxigênio advindas do controlador nebuloso (Figura 26B). Em seguida, este valor é encaminhado ao PID para atualização do valor da velocidade de agitação (Figura 26C). Por fim, esses resultados são utilizados pela segunda rede neural (Rede 2) para estimar o valor da concentração celular (Figura 26D). Este, por sua vez, é utilizada para estimar, com base na derivada da concentração celular em função do tempo, o valor de velocidade específica de crescimento. Por fim, todos os valores são realimentados no controlador nebuloso (Figura 26E).

3.3.2. Treinamento das redes neurais

Para o propósito desta dissertação foram utilizadas as redes neurais dinâmicas. Para o treinamento das mesmas, foi utilizado o *toolbox* de redes neurais do MATLAB. Na interface do toolbox escolheu-se, para ambas as redes, a rede para um problema não linear com um vetor de entrada e uma saída.

Para o primeiro conjunto de dados experimentais (A16), iniciou-se o processo de treinamento da Rede 1 utilizando-se como dados de entrada os valores de vazão de ar e oxigênio e para saída, dados (controlados via árvore de decisão) de porcentagem de oxigênio dissolvido. Determinou-se as porcentagem de dados utilizados para cada fase, treinamento, validação e teste (70, 15 e 15%, respectivamente). Por fim, determinou-se a arquitetura da rede. Após alguns testes preliminares, optou-se, neste caso, por utilizar apenas uma parte do conjunto de dados que fosse representativa do todo. A partir desta delimitação, foram realizados diversos testes variando o número de neurônios na camada escondida, alguns deles mostrados na Tabela 1.

Tabela 1: Valores de R para diferentes arquiteturas de rede para treinamento da Rede 1 (A16)

Neurônios na camada escondida	Regressão (R)		
	Treinamento	Validação	Testes
5	0,55331	0,58729	0,28676
10	0,59655	0,56567	0,50336
15	0,66768	0,59360	0,65731
20	0,67542	0,61935	0,56252

Fonte: Acervo Pessoal

Com base nos valores de R apresentados no treinamento e nos resultados fornecidos pela rede neural após a implementação no simulador, selecionou-se uma camada escondida com 15 neurônios, 4 atrasos e função de saída do tipo sigmoidal. Na camada de saída foi utilizado apenas um neurônio com função de saída do tipo linear.

Para o treinamento da segunda rede neural utilizou-se como variáveis de entrada as vazões de ar e oxigênio, os valores de oxigênio dissolvido e velocidade de agitação. Para saída utilizou-se os dados experimentais de concentração celular. Após alguns testes preliminares, optou-se por utilizar todo o conjunto de dados disponíveis. A divisão do mesmo se deu de forma idêntica ao procedimento previamente descrito.

Na Tabela 2 tem-se os resultados de alguns testes para seleção da arquitetura da Rede 2.

Tabela 2: Valores de R para diferentes arquiteturas de rede para treinamento da Rede 2 (A16)

Neurônios na camada escondida	Regressão (R)		
	Treinamento	Validação	Testes
5	0,99923	0,99920	0,99804
10	0,99974	0,99970	0,99971
15	0,99985	0,99983	0,99986
20	0,99988	0,99985	0,99976

Fonte: Acervo Pessoal

Neste caso, o critério para a escolha da rede foram os resultados fornecidos após a implementação no simulador. Assim, adotou-se uma arquitetura de rede similar à utilizada para o treinamento da primeira rede neural.

O algoritmo de treinamento escolhido, para ambas as redes neurais, foi o de Levenberg-Marquardt.

Para o segundo conjunto de dados experimentais (A12, para teste de robustez do controlador nebuloso), foi adotado o mesmo procedimento descrito anteriormente para o treinamento da Rede 1 do conjunto de dados A16. Novamente, após testes preliminares, optou-se por utilizar uma parte do conjunto de dados representativa do todo. Em seguida foram realizados diversos testes variando-se o número de neurônios na camada escondida, alguns deles mostrados na Tabela 3.

Tabela 3: Valores de R para diferentes arquiteturas de rede para treinamento da Rede 1 (A12)

Neurônios na camada escondida	Regressão (R)		
	Treinamento	Validação	Testes
5	0,70308	0,48218	0,38300
10	0,70491	0,55269	0,31402
15	0,75918	0,55850	0,36556
20	0,77393	0,72218	0,62811

Fonte: Acervo Pessoal

Com base dos valores de R apresentados no treinamento e nos resultados fornecidos pela rede neural após a implementação no simulador, selecionou-se uma camada escondida com 20 neurônios, 4 atrasos e função de saída do tipo sigmoidal. Na camada de saída foi utilizado apenas um neurônio com função de saída do tipo linear.

Para o treinamento da segunda rede neural, novamente, adotou-se o mesmo procedimento descrito anteriormente para o treinamento da Rede 2. Na Tabela 4 tem-se os resultados de alguns testes para a seleção da arquitetura da Rede 2.

Tabela 4: Valores de R para diferentes arquiteturas de rede para treinamento da Rede 2 (A12)

Neurônios na camada escondida	Regressão (R)		
	Treinamento	Validação	Testes
4	0,98335	0,98122	0,97978
5	0,98614	0,97959	0,98291
10	0,98191	0,98194	0,98249
15	0,98934	0,98194	0,98249
20	0,98959	0,98490	0,98754

Fonte: Acervo Pessoal

Neste caso, dada a proximidade dos valores de R, o critério utilizado para a escolha da arquitetura da rede foram os resultados fornecidos após a implementação no simulador. Assim adotou-se uma arquitetura da rede com uma camada escondida com 4 neurônios, 4 atrasos e função de saída do tipo sigmoïdal.

3.3.3. Adaptação da rotina em MATLAB

Após o treinamento das redes neurais, foram gerados os respectivos códigos em MATLAB na forma de duas funções distintas. Para que elas pudessem ser incorporadas à rotina implementada para o controlador nebuloso (SOUSA e SOUSA, 2015), algumas modificações no código foram necessárias.

Primeiramente, a chamada à função correspondente à primeira rede neural foi inserida na rotina, tal como mostra o Quadro 12.

Quadro 12 – Chamada à função correspondente à rede 1.

$X = \{[\text{res}(i,3); \text{res}(i,4)]\};$ $[Y, Xf] = \text{Rede1_OD}(X, Xi);$ $\text{ODPnovo} = Y\{1,1\}(1,1);$
--

Fonte: Acervo Pessoal

Em que Y é vetor de saída (variável OD), X é o vetor de entrada (vazões de ar e de oxigênio fornecidas pelo controlador nebuloso - res(i,3) e res(i,4) - respectivamente), Xi e Xf

correspondem ao primeiro e ao último vetor de atrasos (no caso, como foi escolhida uma rede com 4 atrasos, o vetor foi formado pelos 4 valores anteriores das variáveis de entrada), e Af é o vetor final da camada de atraso.

De posse dos valores de porcentagem de oxigênio dissolvido (ODPnovo), introduziu-se na rotina a equação do PID (ilustrada pela Equação 3), tal como mostra a Equação 20.

$$AgitMedP_{novo} = AgitMedP + kc * ((SetP - ODP_{novo}) - (SetP - res(i - 1,7))) + k_i * (SetP - ODP_{novo}) + kd * ((SetP - ODP_{novo}) - 2 * (SetP - res(i - 1,7)) + (setP - res(i - 2,7)))) \quad (20)$$

Em que $AgitMedP_{novo}$ corresponde à velocidade de agitação calculada no passo atual, $AgitMedP$ é a velocidade de agitação no passo anterior (rpm), kc , k_i e kd são constantes do controlador (em que $kc = Kc$, $k_i = \frac{\Delta t}{T_i}$ e $kd = \frac{T_d}{\Delta t}$), $SetP$ é o *set point* do oxigênio dissolvido (%), ODP_{novo} é a porcentagem de oxigênio dissolvido no passo atual (%) e $res(i-1,7)$ e $res(i-2,7)$ são as porcentagens de oxigênio dissolvido um passo e dois passos anteriores, respectivamente. Os valores de kc , k_i e kd foram: 5; 0,432; 0,00485 respectivamente.

Em seguida, a chamada à função correspondente à segunda rede neural foi inserida na rotina, tal como mostra o Quadro 13.

Quadro 13 - Chamada à função correspondente à rede 2.

$$\begin{aligned} Xc &= \{[ODP_{novo}; AgitMedP_{novo}; res(i,3); res(i,4)]\}; \\ [Yc, Xcf, Acf] &= Rede2_CX(Xc, Xci); \\ CXP_{novo} &= Yc\{1,1\}(1,1); \end{aligned}$$

Fonte: Acervo Pessoal

Em que Yc corresponde ao vetor de saída (variável CXP_{novo}), Xc corresponde ao vetor de entrada (vazões de ar e oxigênio advindas do controlador nebuloso - $res(i,3)$ e $res(i,4)$ - respectivamente, e os novos valores de porcentagem de oxigênio dissolvido - ODP_{novo} - e velocidade de agitação - $AgitMedP_{novo}$), Xci e Xcf correspondem ao primeiro e ao último vetor de atrasos, respectivamente e Acf corresponde ao vetor final da camada de atraso.

A última alteração na rotina foi a inserção do cálculo da velocidade específica de crescimento. Esta é calculada utilizando a Equação 21.

$$\mu = \frac{1}{CX} \frac{dCx}{dt} + \frac{F}{V} \quad (21)$$

Em que μ é a velocidade específica de crescimento, CX é a concentração celular, F é a vazão de alimentação e V é o volume de líquido no biorreator (HORTA, 2011). O excerto da rotina para o cálculo da velocidade específica de crescimento é mostrado no Quadro 14.

Quadro 14 - Excerto da programação para o cálculo da velocidade específica de crescimento

```

A = [res(i-749); CXPnovo];
M = mean(A);
minovo = (CXPnovo - res(i-749,9))/(749*DeltaT)*1/M + F/V
```

Fonte: Acervo Pessoal

Em que A é um vetor contendo os valores de concentração celular no passo atual e seu valor há 749 passos atrás (`res(i-749,9)`). M é a média calculada desses valores. DeltaT é o intervalo de tempo para aferição dos dados *on line* (foi utilizado um valor médio de 0,003 h ou 10,8 s). F é a vazão de alimentação (l/h) e V é o volume do biorreator (l). Foi aplicado um filtro de média móvel para reduzir possíveis surtos no valor da concentração celular. O intervalo de 749 pontos foi necessário para evitar valores de μ negativos erroneamente (dado pela proximidade dos valores de concentração celular aferidos pelo sensor de biomassa em curtos intervalos tempo).

A versão final da rotina em MATLAB pode ser encontrada no Apêndice A.

4. RESULTADOS E DISCUSSÕES

Nesta seção serão analisados os principais resultados obtidos primeiramente com o ajuste do sistema de inferência nebuloso, otimizado via ANFIS, os principais resultados da implementação do simulador e por fim os resultados obtidos para o teste de robustez do controlador nebuloso.

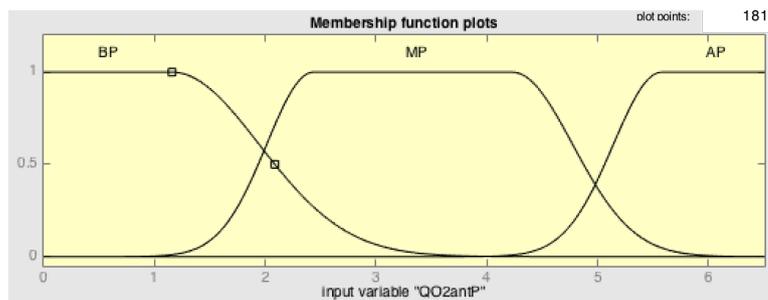
4.1. Resultados com controle nebuloso otimizado via ANFIS

Nesta seção serão descritos os principais resultados obtidos com o controlador nebuloso otimizado via ANFIS utilizando o primeiro conjunto de dados (A16).

4.1.1. Otimização dos parâmetros do sistema de inferência nebuloso “Delta”

A Figura 27 mostra o resultado do ajuste das funções de pertinência de entrada do sistema nebuloso Delta realizado pelo ANFIS.

Figura 27 - Resultado do ajuste das funções de pertinência de entrada do sistema nebuloso Delta realizado pelo ANFIS.



Fonte: Acervo Pessoal

Na Tabela 5 tem-se os valores dos parâmetros das funções de pertinência de entrada do sistema de inferência nebuloso Delta ajustados pelo ANFIS, de acordo com a Equação 18.

Tabela 5 - Valores dos parâmetros das funções de pertinência de entrada do sistema de inferência nebuloso Delta ajustados pelo ANFIS.

Variável Linguística	σ_1	c_1	σ_2	c_2
BP	0,5527	-0,976	0,7889	1,158
QO₂antP				
MP	0,4222	2,434	0,546	4,23
AP	0,4371	5,578	0,5527	7,483

Fonte: Acervo Pessoal

Em que “BP”, “MP” e “AP” são, respectivamente, os valores linguísticos “Baixo e Positivo”, “Médio e Positivo” e “Alto e Positivo”.

Foram ajustadas 3 regras nebulosas pelo ANFIS:

1. Se QO₂antP é BP então Delta é F1
2. Se QO₂antP é MP então Delta é F2
3. Se QO₂antP é AP então Delta é F3

Em que F(i) são as funções lineares de saída, ajustadas pelo ANFIS, tal como mostra a Equação 22 (generalizada):

$$F_i = \alpha * QO_{2antP_i} + \beta \quad (22)$$

Em que α é o coeficiente da variável QO₂antP, e β é o coeficiente linear.

Na Tabela 6 tem-se os valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Delta ajustados pelo ANFIS, de acordo com a Equação 17:

Tabela 6 - Valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Delta ajustados pelo ANFIS.

Funções	α	β
F1	0,00030	0,00454
F2	0,00012	0,01711
F3	0,0002	0,00331

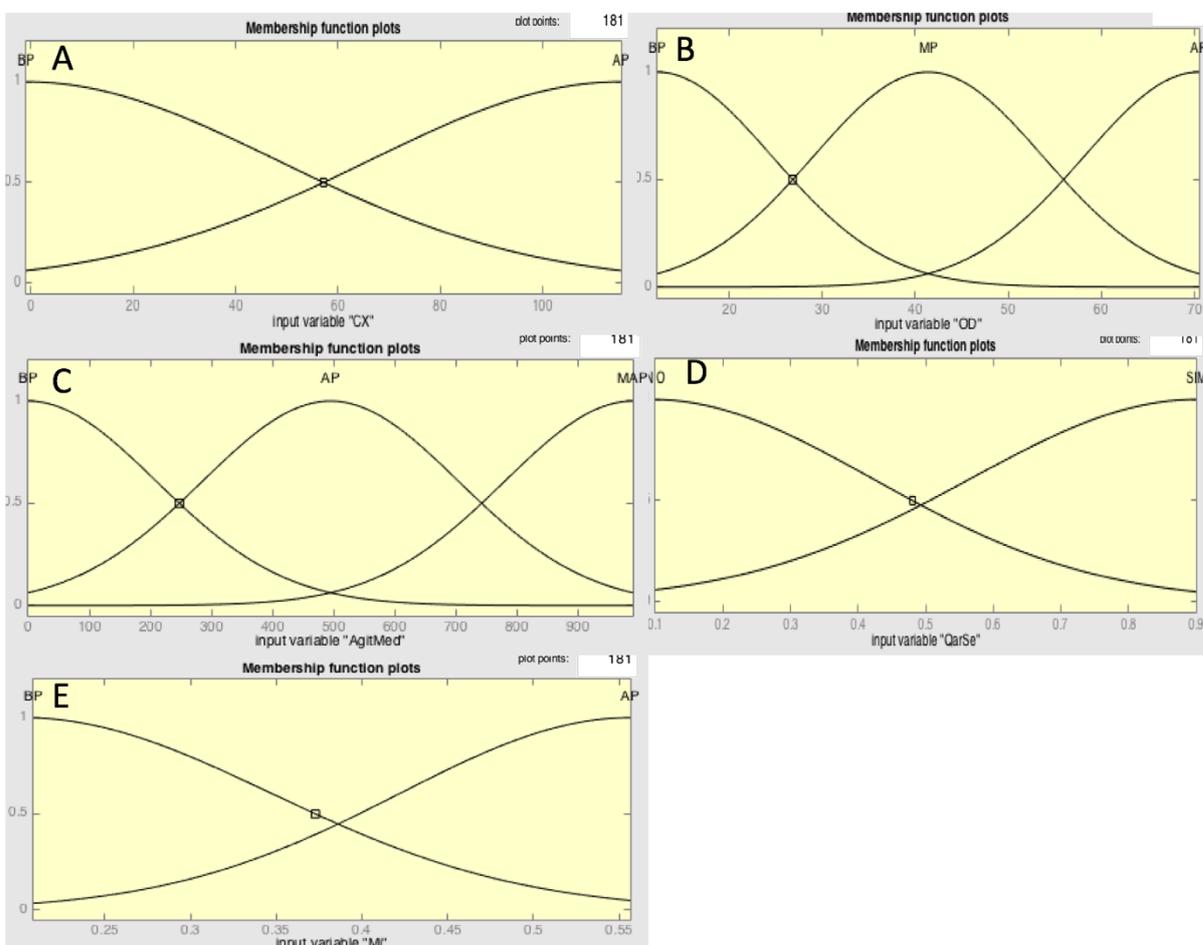
Fonte: Acervo Pessoal

Após o ajuste do sistema de inferência nebuloso Delta foi renomeado para DeltaS.

4.1.2. Otimização dos parâmetros dos sistemas de inferência nebulosos “Princ1” e “Princ2”

A Figura 28 mostra uma composição dos resultados dos ajustes das funções de pertinência de entrada do sistema nebuloso Princ1 realizado pelo ANFIS. As Figuras 28A, B, C, D e E correspondem aos ajustes das variáveis linguísticas CX, OD, AgitMed, QarSe e Mi, respectivamente.

Figura 28 - Ajuste das funções de pertinência de entrada do sistema de inferência nebuloso “Princ1” realizada pelo ANFIS.



Fonte: Acervo Pessoal

Na Tabela 7 tem-se os valores dos parâmetros das funções de pertinência de entrada do sistema de inferência nebuloso Princ1 ajustados pelo ANFIS, de acordo com a Equação 18.

Tabela 7 - Valores dos parâmetros das funções de pertinência de entrada do sistema de inferência nebuloso Princ1 ajustados pelo ANFIS.

Variável Linguística		σ	c
CX	BP	49,34	-0,917
	AP	49,34	115,3
AgitMed	BP	210,2	1,22e-05
	AP	210,2	495
	MAP	210,2	990
OD	BP	12,36	12,28
	MP	12,3	41,4
QarSe	AP	12,36	70,47
	NÃO	0,3275	0,09489
	SIM	0,3362	0,9015
Mi	BP	0,1445	0,2028
	AP	0,1344	0,5566

Fonte: Acervo Pessoal

Foram ajustadas 72 regras nebulosas definidas pelo ANFIS:

1. SE CX É BP E AGITMED É BP E OD É BP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₁
2. SE CX É BP E AGITMED É BP E OD É BP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₂
3. SE CX É BP E AGITMED É BP E OD É BP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₃
4. SE CX É BP E AGITMED É BP E OD É BP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₄
5. SE CX É BP E AGITMED É BP E OD É MP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₅
6. SE CX É BP E AGITMED É BP E OD É MP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₆
7. SE CX É BP E AGITMED É BP E OD É MP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₇
8. SE CX É BP E AGITMED É BP E OD É MP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₈
9. SE CX É BP E AGITMED É BP E OD É AP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₉
10. SE CX É BP E AGITMED É BP E OD É AP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₁₀
11. SE CX É BP E AGITMED É BP E OD É AP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₁₁
12. SE CX É BP E AGITMED É BP E OD É AP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₁₂
13. SE CX É BP E AGITMED É AP E OD É BP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₁₃
14. SE CX É BP E AGITMED É AP E OD É BP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₁₄
15. SE CX É BP E AGITMED É AP E OD É BP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₁₅
16. SE CX É BP E AGITMED É AP E OD É BP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₁₆
17. SE CX É BP E AGITMED É AP E OD É MP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₁₇
18. SE CX É BP E AGITMED É AP E OD É MP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₁₈
19. SE CX É BP E AGITMED É AP E OD É MP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₁₉
20. SE CX É BP E AGITMED É AP E OD É MP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₂₀
21. SE CX É BP E AGITMED É AP E OD É AP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₂₁
22. SE CX É BP E AGITMED É AP E OD É AP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₂₂
23. SE CX É BP E AGITMED É AP E OD É AP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₂₃
24. SE CX É BP E AGITMED É AP E OD É AP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₂₄
25. SE CX É BP E AGITMED É MAP E OD É BP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₂₅
26. SE CX É BP E AGITMED É MAP E OD É BP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₂₆
27. SE CX É BP E AGITMED É MAP E OD É BP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₂₇
28. SE CX É BP E AGITMED É MAP E OD É BP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₂₈
29. SE CX É BP E AGITMED É MAP E OD É MP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₂₉
30. SE CX É BP E AGITMED É MAP E OD É MP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₃₀

31. SE CX É BP E AGITMED É MAP E OD É MP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₃₁
32. SE CX É BP E AGITMED É MAP E OD É MP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₃₂
33. SE CX É BP E AGITMED É MAP E OD É AP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₃₃
34. SE CX É BP E AGITMED É MAP E OD É AP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₃₄
35. SE CX É BP E AGITMED É MAP E OD É AP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₃₅
36. SE CX É BP E AGITMED É MAP E OD É AP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₃₆
37. SE CX É AP E AGITMED É BP E OD É BP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₃₇
38. SE CX É AP E AGITMED É BP E OD É BP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₃₈
39. SE CX É AP E AGITMED É BP E OD É BP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₃₉
40. SE CX É AP E AGITMED É BP E OD É BP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₄₀
41. SE CX É AP E AGITMED É BP E OD É MP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₄₁
42. SE CX É AP E AGITMED É BP E OD É MP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₄₂
43. SE CX É AP E AGITMED É BP E OD É MP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₄₃
44. SE CX É AP E AGITMED É BP E OD É MP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₄₄
45. SE CX É AP E AGITMED É BP E OD É AP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₄₅
46. SE CX É AP E AGITMED É BP E OD É AP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₄₆
47. SE CX É AP E AGITMED É BP E OD É AP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₄₇
48. SE CX É AP E AGITMED É BP E OD É AP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₄₈
49. SE CX É AP E AGITMED É AP E OD É BP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₄₉
50. SE CX É AP E AGITMED É AP E OD É BP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₅₀
51. SE CX É AP E AGITMED É AP E OD É BP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₅₁
52. SE CX É AP E AGITMED É AP E OD É BP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₅₂
53. SE CX É AP E AGITMED É AP E OD É MP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₅₃
54. SE CX É AP E AGITMED É AP E OD É MP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₅₄
55. SE CX É AP E AGITMED É AP E OD É MP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₅₅
56. SE CX É AP E AGITMED É AP E OD É MP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₅₆
57. SE CX É AP E AGITMED É AP E OD É AP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₅₇
58. SE CX É AP E AGITMED É AP E OD É AP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₅₈
59. SE CX É AP E AGITMED É AP E OD É AP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₅₉
60. SE CX É AP E AGITMED É AP E OD É AP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₆₀
61. SE CX É AP E AGITMED É MAP E OD É BP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₆₁
62. SE CX É AP E AGITMED É MAP E OD É BP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₆₂
63. SE CX É AP E AGITMED É MAP E OD É BP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₆₃
64. SE CX É AP E AGITMED É MAP E OD É BP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₆₄
65. SE CX É AP E AGITMED É MAP E OD É MP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₆₅
66. SE CX É AP E AGITMED É MAP E OD É MP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₆₆
67. SE CX É AP E AGITMED É MAP E OD É MP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₆₇
68. SE CX É AP E AGITMED É MAP E OD É MP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₆₈
69. SE CX É AP E AGITMED É MAP E OD É AP E QARSE É NÃO E MI É BP ENTÃO SAÍDA É F'₆₉
70. SE CX É AP E AGITMED É MAP E OD É AP E QARSE É NÃO E MI É AP ENTÃO SAÍDA É F'₇₀
71. SE CX É AP E AGITMED É MAP E OD É AP E QARSE É SIM E MI É BP ENTÃO SAÍDA É F'₇₁
72. SE CX É AP E AGITMED É MAP E OD É AP E QARSE É SIM E MI É AP ENTÃO SAÍDA É F'₇₂

Em que $F'(i)$ são as funções lineares de saída, ajustadas pelo ANFIS, tal como mostra a Equação 23 (generalizada):

$$F'_i = a * CX_i + b * AgitMed_i + c * OD_i + d * QarSe_i + e * Mi_i + f \quad (23)$$

Em que “a”, “b”, “c”, “d”, e “e” são os coeficientes das variáveis CX, AgitMed, OD, QarSe e Mi, respectivamente, e f é o coeficiente linear.

Este grande número de regras é decorrente da estrutura do ANFIS que utiliza todas as combinações possíveis entre os valores linguísticos das variáveis linguísticas de entrada.

No Apêndice B são mostrados os valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Princ1 ajustados pelo ANFIS, de acordo com a Equação 23.

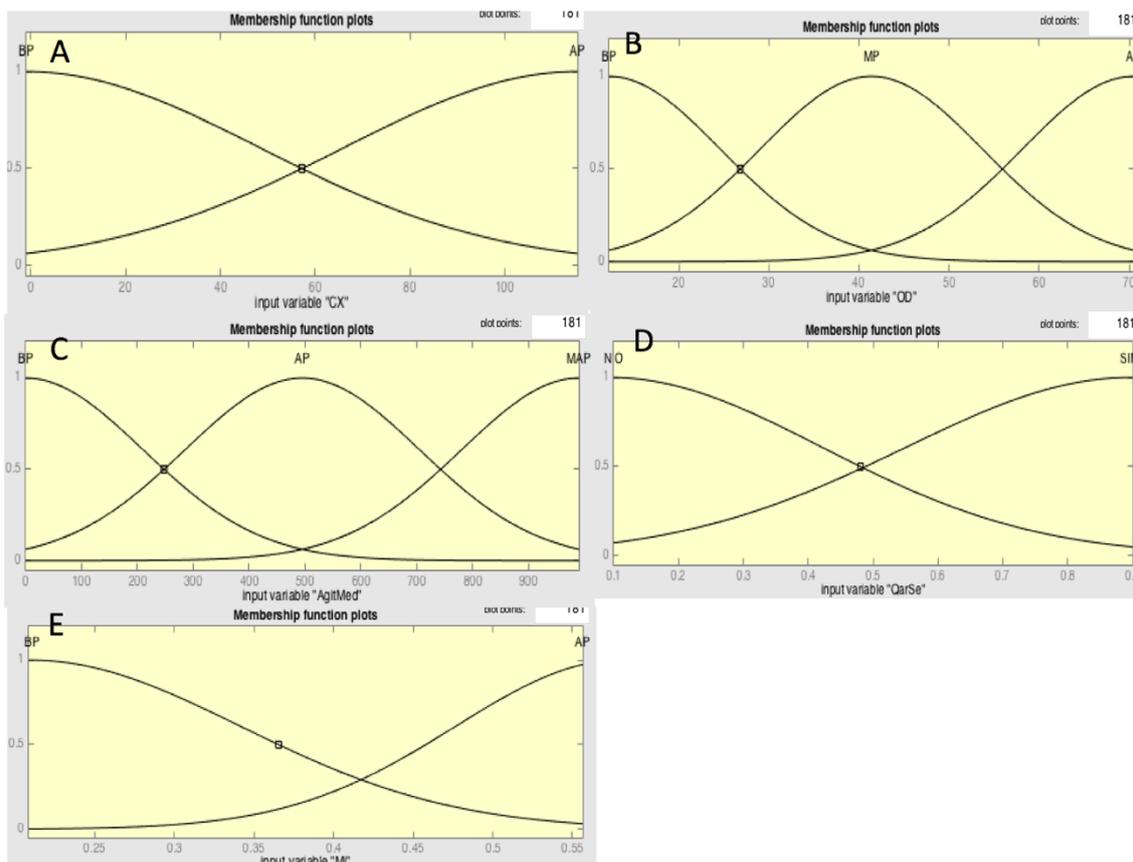
A saída final do sistema é a média ponderada de todas as regras, dada pela Equação 24.

$$\text{Saída final} = \frac{\sum_{i=1}^N (w_i * z_i)}{\sum_{i=1}^N w_i} \quad (24)$$

Em que N é o número de regras nebulosas, w_i é o grau de ativação de cada regra e z_i é o nível de saída de cada regra (MATHWORKS, 2016).

A Figura 29 mostra uma composição dos resultados dos ajustes das funções de pertinência de entrada do sistema nebuloso Princ2 realizado pelo ANFIS. As Figuras 29A, B, C, D e E correspondem aos ajustes de CX, OD, AgitMed, QarSe e Mi, respectivamente.

Figura 29 - Ajuste das funções de pertinência de entrada do sistema de inferência nebuloso Princ2 realizada pelo ANFIS.



Fonte: Acervo Pessoal

Na Tabela 8 tem-se os valores dos parâmetros das funções de pertinência de entrada do sistema de inferência nebuloso Princ2 ajustados pelo ANFIS, de acordo com a Equação 18.

Tabela 8 - Valores dos parâmetros das funções de pertinência de entrada do sistema de inferência nebuloso Princ2 ajustados pelo ANFIS.

Variável Linguística		σ	c
CX	BP	49,34	-0,916
	AP	49,34	115,3
AgitMed	BP	210,2	2,429e-05
	AP	210,2	495
	MAP	210,2	990
OD	BP	12,36	12,28
	MP	12,36	41,38
	AP	12,36	70,47
QarSe	NÃO	0,3276	0,09486
	SIM	0,3473	0,897
Mi	BP	0,1324	0,2096
	AP	0,1041	0,5804

Fonte: Acervo Pessoal

Novamente, foram ajustadas 72 regras nebulosas pelo ANFIS, sendo que a principal diferença para as 72 regras ajustadas para o sistema de inferência nebuloso “Princ1” reside nos valores dos parâmetros das funções lineares de saída. Estes, por sua vez, são apresentados no Apêndice C.

Observa-se que os parâmetros das funções de pertinência de entrada, ajustados para as subrotinas “Princ1” e “Princ2”, foram muito próximos. Isto se deve à utilização do mesmo conjunto de dados (de entrada) no treinamento no ANFIS, para ambos os casos.

Após o ajuste dos sistemas de inferência nebulosos, Princ1 foi renomeado para Princ1S_red5 e Princ2 para Princ2S_red13.

4.1.3. Resultados utilizando o controle nebuloso otimizado via ANFIS

De posse da rotina em MATLAB criada por Sousa e Sousa (2015), dos novos sistemas de inferência nebulosos otimizados e de dados de controle de cultivo de *E. coli* modificada para expressão de proteínas recombinantes, foi realizada uma busca, de forma heurística, pelas faixas

ótimas de QarEscolha e QO₂Escolha. Esta busca foi efetuada ajustando o sistema, de modo que o mesmo apresentasse resultados de vazões de ar e oxigênio coerentes com os obtidos por Horta (2011) no cultivo A16 utilizando o controle híbrido (PID + Árvore de decisão) (HORTA, 2011).

O Quadro 15 ilustra as faixas para QarEscolha e QO₂Escolha, conforme obtidos por Sousa e Sousa (2015), utilizados como valores iniciais para realização da busca heurística.

Quadro 15 - Faixas de valores ótimos para QarEscolha e QO₂Escolha utilizados no simulador simples

Faixa de valores	QarP / QO₂P
QarEscolha \geq 0,6	QarP = QarantP + DeltaP
QarEscolha $<$ 0,6	QarP = QarantP - DeltaP
QO₂Escolha \geq 0,65	QO ₂ P = QO ₂ antP + DeltaP
QO₂Escolha \leq 0,45	QO ₂ P = QO ₂ antP - DeltaP
0,45 $<$ QO₂Escolha $<$ 0,65	QO ₂ P = QO ₂ antP

Fonte: SOUSA, SOUSA (2015)

Após algumas simulações, determinou-se as faixas ótimas tal como mostra o Quadro 16.

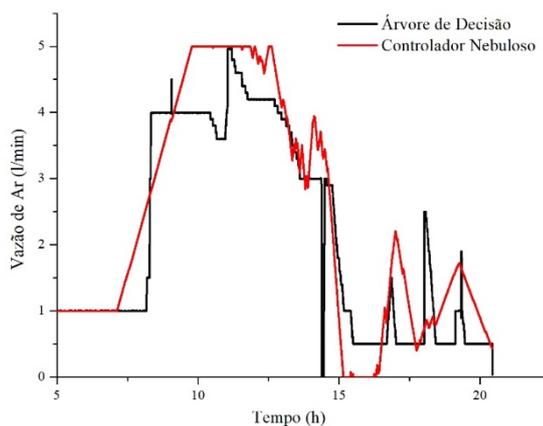
Quadro 16 - Faixas de valores ótimos para QarEscolha e QO₂Escolha utilizados no simulador simples

Faixa de valores	QarP / QO₂P
QarEscolha \geq 0,595	QarP = QarantP + DeltaP
QarEscolha $<$ 0,595	QarP = QarantP - DeltaP
QO₂Escolha \geq 0,63	QO ₂ P = QO ₂ antP + DeltaP
QO₂Escolha \leq 0,28	QO ₂ P = QO ₂ antP - DeltaP
0,28 $<$ QO₂Escolha $<$ 0,63	QO ₂ P = QO ₂ antP

Fonte: Acervo Pessoal

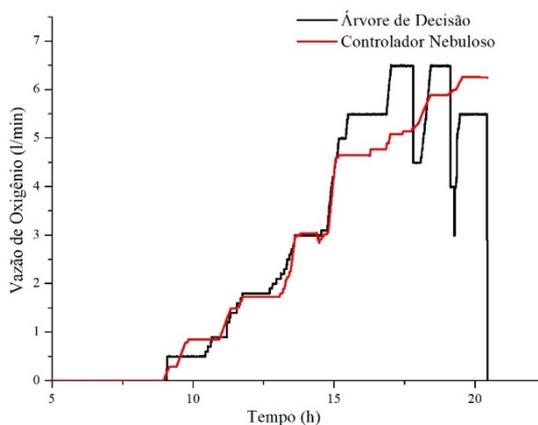
As Figuras 30 e 31 ilustram os resultados finais das curvas das vazões de ar e oxigênio, respectivamente, em função do tempo. A linha preta representa os dados de cultivo controlado utilizando a árvore de decisão, enquanto que a linha vermelha representa os dados obtidos com o controlador nebuloso otimizado.

Figura 30 - Vazão de ar (l/min) em função do tempo (h)



Fonte: Acervo Pessoal

Figura 31 - Vazão de oxigênio (l/min) em função do tempo (h)



Fonte: Acervo Pessoal

Observou-se que nos dois casos, as tendências requerida pelo cultivo de *Escherichia coli* para expressão de proteínas recombinantes foram seguidas com sucesso. Além disso, os degraus das vazões de ar e oxigênio foram suavizados. Este resultado reforça as conclusões de Sousa e Sousa (2015), que afirmaram que o sistema de lógica nebulosa é uma opção viável para o controle das vazões de ar e oxigênio no cultivo de *E.coli* para expressão de proteínas recombinantes.

4.2. Resultados com o simulador

A execução do simulador ocorreu em duas etapas. A primeira consistiu na implementação de um simulador simples, composto pela primeira rede neural (Rede 1 – Figura 26B), pelo controlador nebuloso (Figura 26A), pelo controlador PID (Figura 26C) e por dados experimentais de concentração celular. Já a segunda etapa consistiu da implementação de um simulador avançado, composto pelo simulador simples, pela segunda rede neural (Rede 2 – Figura 26D) e pela Equação da velocidade específica de crescimento, tal como mostra a Figura 26E.

4.2.1. Resultados utilizando o simulador simples

Após testes utilizando diversas redes neurais observou-se que embora os valores de R fossem próximos a 1 (para as fases de treinamento, validação e testes das redes), os resultados globais (do simulador) não foram satisfatórios (rede supertreinada). Optou-se então por flexibilizar os valores de R, a fim de treinar uma rede com uma melhor capacidade de generalização. O resultado do treinamento da Rede 1 escolhida é mostrado na Tabela 9.

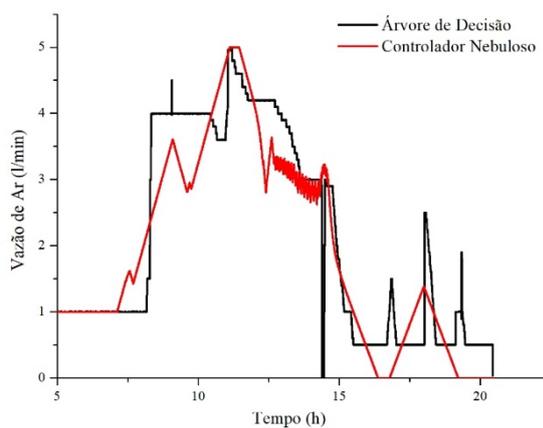
Tabela 9 - Resultados do treinamento da rede neural (Rede 1) para o primeiro conjunto de dados

	Amostras	R
Treinamento	216	0,66768
Validação	47	0,59360
Teste	47	0,65734

Fonte: Acervo Pessoal

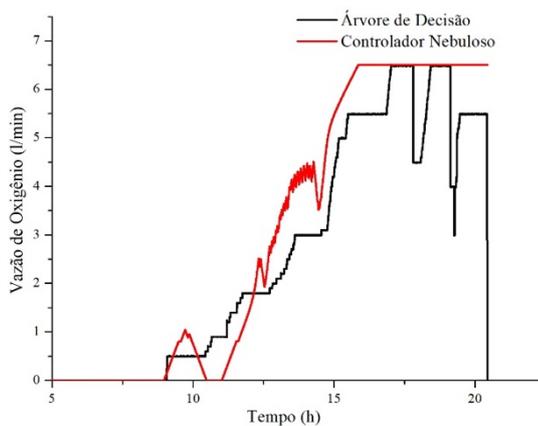
Os melhores resultados para as vazões de ar e oxigênio, utilizando o simulador simples, são mostrados nas Figuras 32 e 33, respectivamente.

Figura 32 - Vazão de ar (l/min) em função do tempo (h) utilizando o simulador simples (primeiro conjunto de dados).



Fonte: Acervo Pessoal

Figura 33 - Vazão de oxigênio (l/min) em função do tempo (h) utilizando o simulador simples (primeiro conjunto de dados).



Fonte: Acervo Pessoal

O Quadro 17 mostra as novas faixas de valores ótimos para QarEscolha e QO₂Escolha utilizados no simulador simples.

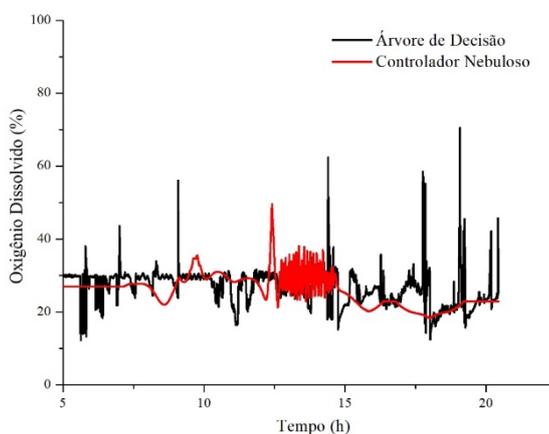
Quadro 17 - Faixas de valores ótimos para QarEscolha e QO₂Escolha utilizados no simulador simples

Faixa de valores	QarP / QO ₂ P
QarEscolha \geq 0,675	QarP = QarantP + DeltaP
QarEscolha $<$ 0,675	QarP = QarantP - DeltaP
QO ₂ Escolha \geq 0,35	QO ₂ P = QO ₂ antP + DeltaP
QO ₂ Escolha \leq 0,05	QO ₂ P = QO ₂ antP - DeltaP
0,05 < QO ₂ Escolha < 0,35	QO ₂ P = QO ₂ antP

Fonte: Acervo Pessoal

Embora a resposta simulada tenha se mostrado um pouco oscilatória no trecho entre 12 e 14 horas para ambas as curvas, novamente, as tendências das necessidades das vazões de ar e oxigênio requeridas pelo cultivo de *Escherichia coli* para expressão de proteínas recombinantes foram seguidas. Na Figura 34, plotou-se o gráfico da porcentagem de oxigênio dissolvido controlada utilizando a árvore de decisão (linha preta), juntamente com os dados de oxigênio dissolvido simulado (linha vermelha), obtidos utilizando o controlador nebuloso, em função do tempo (h).

Figura 34 - Comparação entre as porcentagens de oxigênio dissolvido simulado e experimental utilizando o simulador simples (primeiro conjunto de dados).



Fonte: Acervo Pessoal

Utilizando o controle nebuloso foi possível manter a porcentagem de oxigênio dissolvido em valores muito próximos ao valor ótimo de 30% durante todo o cultivo. De um modo geral, os picos de oxigênio dissolvido foram abrandados, ou seja, a resposta ao controle nebuloso foi mais suave em relação ao controle por árvore de decisão.

4.2.2. Resultados utilizando o simulador avançado

O resultado do treinamento da rede neural escolhida (Rede 2) é mostrado na Tabela 10.

Tabela 10 - Resultados do treinamento da rede neural (Rede 2) para o primeiro conjunto de dados

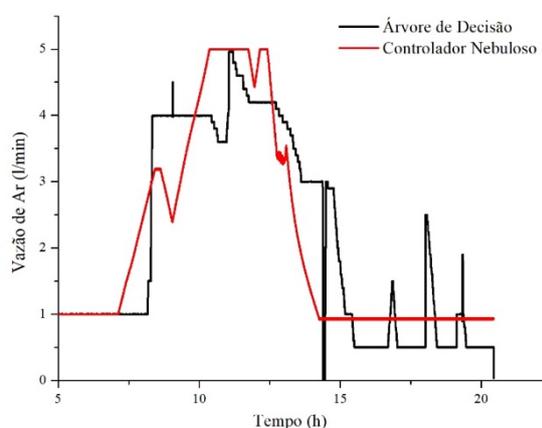
	Amostras	R
Treinamento	3681	0,99985
Validação	789	0,99983
Teste	789	0,99986

Fonte: Acervo Pessoal

Após testes utilizando diversas redes neurais observou-se que, diferentemente dos resultados obtidos para o treinamento da Rede 1 (mostrados na Tabela 7), foi possível manter os valores de R próximos a 1 (para as fases de treinamento, validação e testes). Estes resultados são decorrentes de um comportamento “menos oscilatório” dos dados de saída utilizados para o treinamento da Rede 2.

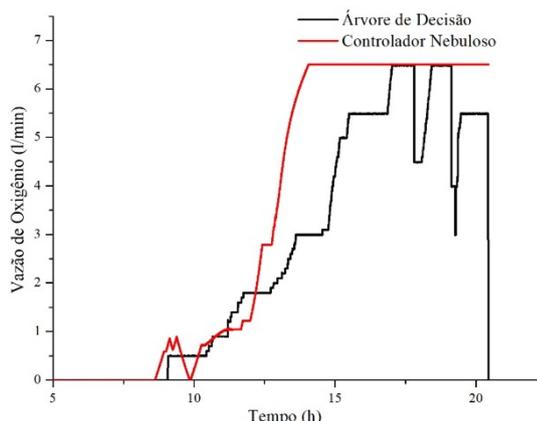
Os melhores resultados para as vazões de ar e oxigênio, utilizando o simulador avançado, são mostrados nas Figuras 35 e 36 respectivamente.

Figura 35 - Vazão de ar (l/min) em função do tempo (h) utilizando o simulador avançado (primeiro conjunto de dados).



Fonte: Acervo Pessoal

Figura 36 - Vazão de oxigênio (l/min) em função do tempo (h) utilizando o simulador avançado (primeiro conjunto de dados).



Fonte: Acervo Pessoal

O Quadro 18 mostra as faixas de valores ótimos para QarEscolha e QO₂Escolha utilizando o simulador avançado.

Quadro 18 - Faixas de valores ótimos para QarEscolha e QO₂Escolha utilizando o simulador avançado.

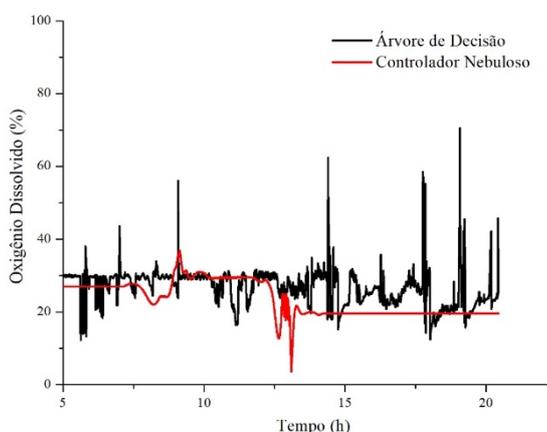
Faixas de valores	QarP / QO ₂ P
QarEscolha $\geq 0,56$	QarP = QarantP + DeltaP
QarEscolha $< 0,56$	QarP = QarantP - DeltaP
QO₂Escolha $\geq 0,565$	QO ₂ P = QO ₂ antP + DeltaP
QO₂Escolha $\leq 0,05$	QO ₂ P = QO ₂ antP - DeltaP
$0,05 < \text{QO}_2\text{Escolha} < 0,65$	QO ₂ P = QO ₂ antP

Fonte: Acervo Pessoal

Ressalta-se que, conforme explicado na seção de Metodologia, a implementação do simulador simples foi uma etapa para a implementação do simulador avançado. O primeiro compreendendo a Rede 1, o controlador PID e o controlador nebuloso otimizado, com o segundo compreendendo o simulador simples, a Rede 2 e a equação para cálculo da velocidade específica de crescimento. Ao comparar as curvas das vazões de ar e oxigênio utilizando o simulador simples (Figuras 32 e 33, respectivamente), com as obtidas utilizando o simulador avançado (Figuras 35 e 36, respectivamente) observou-se que estas se mostraram mais estáveis em relação ao simulador simples. Novamente, as tendências das necessidades das vazões de ar e oxigênio requerida pelo cultivo de *Escherichia coli* para expressão de proteínas recombinantes foram seguidas. Este resultado comprova as conclusões de Sousa e Sousa

(2015). Na Figura 37, plotou-se o gráfico da porcentagem de oxigênio dissolvido experimental (linha preta) e simulado (linha vermelha) em função do tempo.

Figura 37 - Comparação entre as porcentagens de oxigênio dissolvido simulado e experimental utilizando o simulador avançado (primeiro conjunto de dados).



Fonte: Acervo Pessoal

Utilizando o controle nebuloso foi possível manter a porcentagem de oxigênio dissolvido em uma faixa de 20% a 31%, que compreende o valor ótimo de 30%. Pode-se verificar a presença de dois picos mais baixos de porcentagem de oxigênio dissolvido, entre 13 e 13,5h. Mas, de um modo geral, a resposta ao controle nebuloso foi novamente mais suave em relação ao controle por árvore de decisão. É possível concluir que neste caso haveria uma diminuição no *stress* metabólico, e conseqüentemente um aumento da produtividade do cultivo celular. Comprovou-se, portanto, que o sistema de lógica nebulosa é uma opção viável para o controle do oxigênio dissolvido no cultivo de *E.coli* para expressão de proteínas recombinantes.

4.3. Resultados de teste de robustez do simulador

Nesta seção serão apresentados os resultados de teste de robustez do controlador nebuloso. Foi utilizado um segundo conjunto de dados (A12) de cultivo de *E. coli* para expressão de proteínas recombinantes, também gentilmente cedidos por Horta (HORTA, 2011).

4.3.1. Resultados utilizando o simulador simples

Após testes utilizando diversas redes neurais, o resultado do treinamento da Rede 1 escolhida (para o segundo conjunto de dados, A12) é mostrado na Tabela 11.

Tabela 11 - Resultados do treinamento da rede neural (Rede 1) para o segundo conjunto de dados

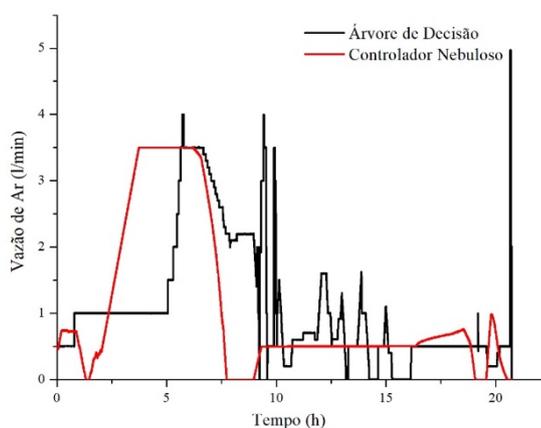
	Amostras	R
Treinamento	290	0,77393
Validação	62	0,72218
Teste	62	0,62811

Fonte: Acervo Pessoal

Observa-se que ao comparar os valores de R dispostos na Tabela 9 com os resultados da Tabela 11, nota-se que foi possível obter um melhor R ao aumentar o número de neurônios da camada escondida de 15 (treinamento de R1 para o primeiro conjunto de dados) para 20 (treinamento de R1 para o segundo conjunto de dados).

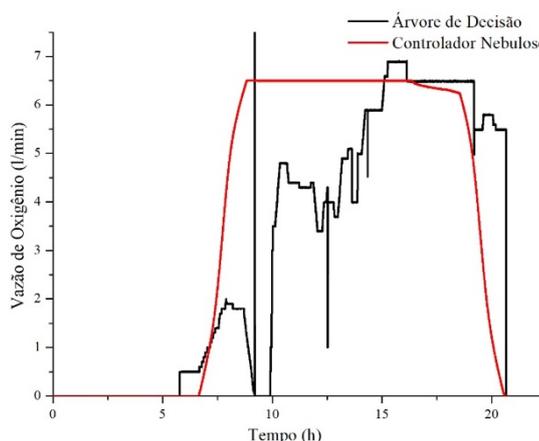
Os melhores resultados para as vazões de ar e oxigênio, utilizando o simulador simples, são mostrados nas Figuras 38 e 39, respectivamente.

Figura 38 - Vazão de ar (l/min) em função do tempo (h) utilizando o simulador simples (segundo conjunto de dados).



Fonte: Acervo Pessoal

Figura 39 - Vazão de oxigênio (l/min) em função do tempo (h) utilizando o simulador simples (segundo conjunto de dados).



Fonte: Acervo pessoal

O Quadro 19 mostra as novas faixas de valores ótimos para QarEscolha e QO₂Escolha utilizados no simulador simples.

Quadro 19 - Faixas de valores ótimos para QarEscolha e QO₂Escolha utilizados no simulador simples

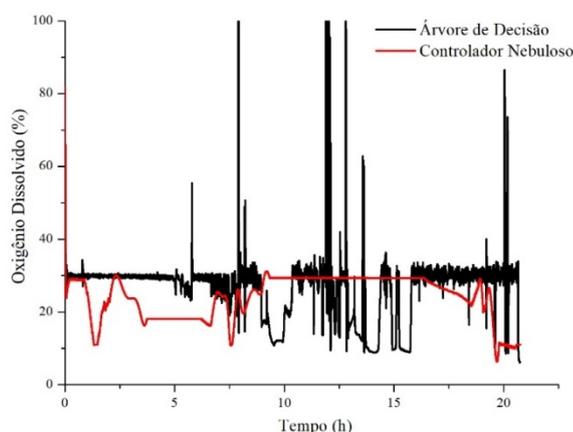
Faixa de valores	QarP / QO ₂ P
QarEscolha \geq 16	QarP = QarantP + DeltaP
QarEscolha $<$ 16	QarP = QarantP - DeltaP
QO₂Escolha \geq 6	QO ₂ P = QO ₂ antP + DeltaP
QO₂Escolha \leq 4	QO ₂ P = QO ₂ antP - DeltaP
4 < QO₂Escolha < 6	QO ₂ P = QO ₂ antP

Fonte: Acervo Pessoal

Como o conjunto utilizado para o retreinamento da rede neural difere do conjunto utilizado no treinamento dos sistemas de inferência nebulosos pelo ANFIS, novas faixas de valores ótimos para QarEscolha e QO₂Escolha foram heurísticamente determinadas. Isso possibilitou que não fosse necessário reajuste nas funções definidas pelo ANFIS para o controlador nebuloso. Novamente, as tendências das necessidades das vazões de ar e oxigênio requeridas pelo cultivo de *Escherichia coli* para expressão de proteínas recombinantes foram seguidas. Este resultado demonstra a robustez do controlador nebuloso proposto. Na Figura 40, plotou-se o gráfico da porcentagem de oxigênio dissolvido experimental (linha preta), obtido

utilizando a árvore de decisão, juntamente com os dados de oxigênio dissolvido simulados (linha vermelha), obtidos utilizando o controlador nebuloso, em função do tempo.

Figura 40 - Comparação entre as porcentagens de oxigênio dissolvido simulado e experimental utilizando o simulador simples (segundo conjunto de dados).



Fonte: Acervo pessoal

Utilizando o controle nebuloso foi possível manter a porcentagem de oxigênio dissolvido em valores muito próximos ao valor ótimo de 30% durante todo o cultivo. De um modo geral, os picos de oxigênio dissolvido foram abrandados, ou seja, a resposta ao controle nebuloso foi mais suave em relação ao controle por árvore de decisão.

4.3.2. Resultados utilizando o simulador avançado

O resultado do treinamento da rede neural escolhida (Rede 2) é mostrado na Tabela 12.

Tabela 12 - Resultados do treinamento da rede neural (Rede 2) para o segundo conjunto de dados

	Amostras	R
Treinamento	5160	0,98335
Validação	1106	0,98122
Teste	1106	0,97978

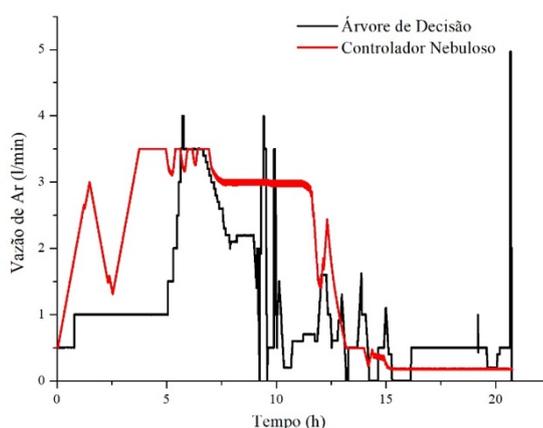
Fonte: Acervo Pessoal

Observa-se que, ao comparar os valores de R dispostos na Tabela 10 com os resultados da Tabela 12, nota-se uma diminuição nos valores de R para as três fases (treinamento, validação e teste). Após diversos testes constatou-se a necessidade de diminuição do número

de neurônios da camada escondida da rede neural, uma vez que a mesma estava “super-treinada”. Uma vez reduzido o número de neurônios da camada escondida de 15 (treinamento de R2 para o primeiro conjunto de dados) para 4 (treinamento de R2 para o segundo conjunto de dados) houve uma pequena redução no valor de R.

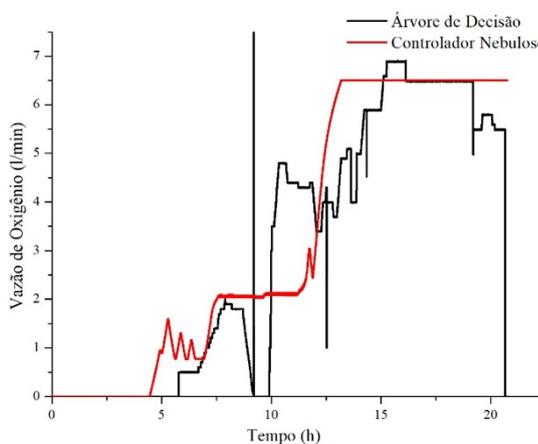
Os melhores resultados para as vazões de ar e oxigênio, utilizando o simulador avançado são mostrados nas Figuras 41 e 42 respectivamente.

Figura 41 - Vazão de ar (l/min) em função do tempo (h) utilizando o simulador avançado (segundo conjunto de dados).



Fonte: Acervo Pessoal

Figura 42 - Vazão de oxigênio (l/min) em função do tempo (h) utilizando o simulador avançado (segundo conjunto de dados).



Fonte: Acervo Pessoal

O Quadro 20 mostra as novas faixas de valores ótimos para QarEscolha e QO₂Escolha utilizados no simulador avançado.

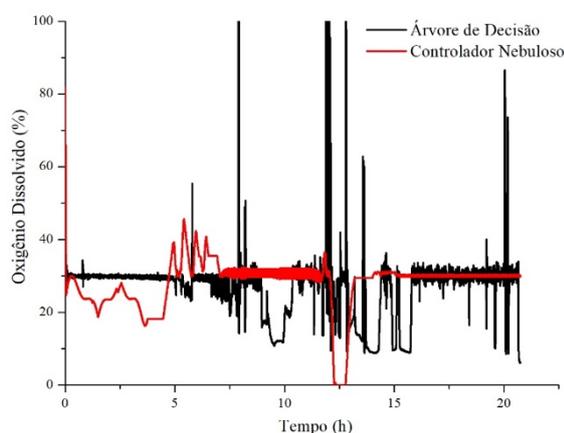
Quadro 20 - Faixas de valores ótimos para QarEscolha e QO₂Escolha utilizados no simulador avançado.

Faixa de valores	QarP / QO₂P
QarEscolha ≥ 7	QarP = QarantP + DeltaP
QarEscolha < 7	QarP = QarantP - DeltaP
QO₂Escolha ≥ -108	QO ₂ P = QO ₂ antP + DeltaP
QO₂Escolha ≤ -195	QO ₂ P = QO ₂ antP - DeltaP
$-195 < \text{QO}_2\text{Escolha} < -108$	QO ₂ P = QO ₂ antP

Fonte: Acervo Pessoal

Novamente, como o conjunto utilizado para o retreinamento da rede neural difere do conjunto utilizado no treinamento dos sistemas de inferência nebulosos pelo ANFIS, novas faixas de valores ótimos para QarEscolha e QO₂Escolha foram heurísticamente determinadas. Há de se notar que houve uma mudança significativa para os valores de QO₂Escolha (comparando os Quadros 18 e 20). Isso poderia ser de se esperar, já que existe uma ampla faixa de valores para os coeficientes “a”, “b”, “c”, “d”, “e” e “f”, conforme observado na Tabela 14 apresentada no Apêndice C. Novamente, as tendências das necessidades das vazões de ar e oxigênio requeridas pelo cultivo de *Escherichia coli* para expressão de proteínas recombinantes foram seguidas. Este resultado demonstra a robustez do controlador nebuloso proposto. Na Figura 43, plotou-se o gráfico da porcentagem de oxigênio dissolvido experimental (linha preta), obtido utilizando a árvore de decisão, juntamente com os dados de oxigênio dissolvido simulados (linha vermelha), obtidos utilizando o controlador nebuloso, em função do tempo.

Figura 43 - Comparação entre as porcentagens de oxigênio dissolvido simulado e experimental utilizando o simulador avançado (segundo conjunto de dados).



Fonte: Acervo Pessoal

Utilizando o controle nebuloso foi possível manter a porcentagem de oxigênio dissolvido em uma faixa de 20% a 40%, que compreende o valor ótimo de 30%. De um modo geral, a resposta ao controle nebuloso foi novamente mais suave em relação ao controle por árvore de decisão, é possível concluir que neste caso também haveria uma diminuição no *stress* metabólico, e conseqüentemente um aumento da produtividade do cultivo celular. Comprovou-se portanto que o sistema de lógica nebulosa é uma opção viável para o controle do oxigênio dissolvido no cultivo de *E.coli* para expressão de proteínas recombinantes, com boa robustez em relação aos parâmetros do controlador otimizado via ANFIS. Os principais resultados do trabalho desenvolvido neste mestrado foram compilados em um artigo, aceito para apresentação e publicação no *European Symposium on Computer-Aided Process Engineering* (ESCAPE-28).

5. CONCLUSÕES

Tal como elucidado na seção de Resultados e Discussões, concluiu-se que é possível *fuzzyficar* a árvore de decisão/pseudocódigo existente para controle de oxigênio dissolvido.

Foi possível modelar o processo com redes neurais artificiais, considerando suas principais variáveis de estado.

A integração entre o controlador nebuloso e o modelo neural do processo, a fim de criar um simulador, foi feita em duas etapas, sendo que na primeira foi implementado um simulador simples e na segunda o simulador avançado. O simulador final forneceu bons resultados em relação ao controle da concentração de oxigênio dissolvido. De um modo geral, as tendências das necessidades de variação das vazões de ar e oxigênio, requeridas pelo cultivo de *E. coli* para expressão de proteínas recombinantes, foram satisfeitas pelo sistema. Ou seja, a lógica nebulosa foi capaz de indicar as necessidades do processo em estudo. Além disso, utilizando o controlador nebuloso foi possível obter comportamentos mais suaves nas variáveis manipuladas (variação das vazões de ar e de oxigênio), em relação às obtidas pela árvore de decisão. Isto acarretaria em um aumento da vida útil dos fluxômetros de massa, responsáveis pela injeção de ar e de oxigênio no sistema.

Pode-se observar que os picos de oxigênio dissolvido foram abrandados, ou seja, a resposta ao controle nebuloso foi mais suave em relação ao controle por árvore de decisão. Isto possivelmente acarretaria em uma diminuição do *stress* metabólico e, conseqüentemente, um aumento da produtividade do cultivo celular, haja vista a sensibilidade em relação a variações bruscas na porcentagem de oxigênio dissolvido durante o cultivo.

6. SUGESTÕES PARA TRABALHOS FUTUROS

A fim de dar continuidade ao processo de desenvolvimento do controlador nebuloso para controle do oxigênio dissolvido sugere-se:

- i) Implementação do controlador nebuloso na ferramenta SUPERSYS_HCDC e realização de cultivos de *E. coli* para mais testes de robustez do mesmo.
- ii) Implementação de um sistema para correção automática das faixas de QarEscolha e QO₂Escolha, baseado em redes neurais artificiais.
- iii) Aperfeiçoamento do controlador nebuloso para que o mesmo possa ser utilizado em outros tipos de cultivos celulares, haja vista o grande número de cultivos celulares que podem ser conduzidos pelo SUPERSYS_HCDC.
- iv) Aperfeiçoamento do simulador criado incorporando mais variáveis às redes neurais, a fim de melhorar seu desempenho.

7. REFERÊNCIAS BIBLIOGRÁFICAS

AHMAD, Z; CROWLEY, D; MARINA, N; JHA, S.K. Estimation of biosurfactant yield produced by *Klebsiella sp.* FKOD36 bacteria using artificial neural network approach. **Measurements**. v. 81, p. 163 – 173, 2015.

APPLIKON. **Biorreatores: Da Descoberta à Produção**. Delft (Países Baixos), 2016. 56 p.

BABAEIPOUR, V; SHOJAOSADATI, S.A; ROBATJAZI, S.M; KHALILZADEH, R; MAGHSOUDI, N. Over-production of human interferon- γ by HCDC of recombinant *Escherichia coli*. **Process Biochemistry**, v. 42, n. 1, p. 112–117, 2006.

BAUGHMAN, D.R; LIU, Y.A. Introduction to Neural Networks. In: _____. **Neural Networks in Bioprocessing and Chemical Engineering**. 1. ed. San Diego: Academic Press, INC, 1995.

BEALE, M.H; HAGAN, M.T; DEMUTH, H.B. **Neural Network Toolbox User's Guide**. Natick: MathWorks, 2017

BELOHLAVEK, R; DAUBEN, J. W; KLIR, G.J. **FUZZY LOGIC AND MATHEMATICS: A Historical Perspective**. 1. ed. Oxford Press, 2017.

CARNEIRO, S; FERREIRA, E.C; ROCHA, I. Metabolic responses to recombinant bioprocesses in *Escherichia coli*. **Journal of Biotechnology**, v. 164, n.3, p. 396 – 408, 2013.

CASTRO, L.N; ZUBEN, F.J. **Redes Neurais Artificiais**. Campinas: UNICAMP/DCA/FEEC, 2003. 48p. Apostila

CDC: Centers for Disease Control and Prevention. E. coli Homepage: general information. Atlanta, 2015. Disponível em: <<https://www.cdc.gov/ecoli/general/index.html>>. Acesso em: 11 ago. 2016.

CDC: Centers for Disease Control and Prevention. Pneumococcal disease Homepage: general information. Atlanta, 2017. Disponível em: <<https://www.cdc.gov/pneumococcal/>>. Acesso em: 15 set. 2017.

EPPENDORF. **Equipamentos para bioprocessos**. Hamburgo, 2017. Disponível em: <<https://online-shop.ependorf.com.br/BR-pt/Equipamentos-para-Bioprocesso-44559.html#goto-Equipamentos-para-Bioprocesso-WebPMain-44559>>. Acesso em: 28 dez. 2017.

FAULKNER, E; BARRET, M; KIERAN, P; CASEY, E; PARADISI, F; ENGEL, P; GLENNON, B. Use of Fed-Batch Cultivation for Achieving High Cell Densities for the Pilot-Scale Production of a Recombinant Protein (Phenylalanine Dehydrogenase) in *Escherichia*

coli. **Biotechnology**. v. 22, p. 889 – 897, 2006.

FINOCCHIO, M.A.F. **Noções de Redes Neurais Artificiais**. Cornélio Procópio: UTFPR/Departamento Acadêmico de Elétrica, 2014. 29p. Apostila

GENENTECH. **Moment 3 of 40: Cloning Gene**. São Francisco, 2016. Disponível em: <<https://www.gene.com/40th/cloning-insulin>>. Acesso em 01 nov. 2016

GLASSEY, J; MONTAGUE, G.A; WARD, A.C. KARA, B.V. Artificial Neural Network Based Experimental Design Procedures for Enhancing Fermentation Development. **Biotechnology and Bioengineering**. v. 44, p. 397 – 405, 1994.

GOMIDE, A.A.C; GUDWIN, R. R. Modelagem , Controle , Sistemas e Lógica Fuzzy. **SBA Controle & Automação**, v. 4, n. 1, p. 97 – 115, 1994.

GONÇALVES, V.M. **Cultivo de *Streptococcus pneumoniae* sorotipo 23F: produção e putificação de seu polissacarídeo capsular para formulação de vacinas**. Tese (Doutorado) – Faculdade de Ciências Farmacêuticas, Universidade de São Paulo, São Paulo, 2002.

HORTA, A. C.L; **Técnicas computacionais inteligentes para a inferência de estado e a otimização de cultivos de *Streptococcus pneumoniae***. 2008. 101 f. Dissertação (Mestrado em Engenharia Química) - Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, 2008

HORTA, A. C.L; **Sistema automático de supervisão e controle de cultivos de alta densidade celular de *E. coli* recombinante**. 2011. 170 f. Tese (Doutorado em Engenharia Química) - Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, 2011.

JANG, J.S.R; SUN, C.T; MIZUTANI, E. Neuro Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. 1 ed. Prentice Hall, 1997. 614 p.

KOPOT, A. **Regulation of Citric Acid Cycle**. Disponível em: <<http://www.aklectures.com/subject/biochemistry#184-Carbohydrate%20Metabolism>> Acesso em: 02 dez. 2017

KONSTANTINOV, K. B; YOSHIDA, T. Mini review: Knowledge-Based Control of Fermentation Processes. **Biotechnology and bioengineering**, v.39, n. 5 p. 479 – 486, 1992.

LULI, G. W; STROHL, W.R. Comparison of Growth, Acetate Production, and Acetate Inhibition of *Escherichia coli* Strains in Batch and Fed-Batch Fermentations. *Applied and Environmental Microbiology*. v. 56, n. 4, p. 1004 – 1011, 1990.

LEE, C.C. Fuzzy Logic in Control System: Fuzzy Logic Controller, Part II. **Transactions on Systems, Man, and Cybernetics**. v. 20, n.2, p. 419 – 435, 1990.

LEE, S. Y; High cell-density culture of *Escherichia coli*. **Trends in biotechnology**, v. 14, n. 3, p. 98–105, 1996.

MÄRKEL, H; ZENNECK, C; DUBACH, A.CH; OGBONNA, J. C. Cultivation of *Escherichia coli* to high cell densities in a dialysis reactor. **Applied Microbiology Biotechnology**, v. 39, n.1, p. 48 – 52, 1992.

MORI, H; YANO, T; KOBAYASHI, T; SHIMIZU, S. High Density Cultivation of Biomass in Fed-Batch System with DO-STAT. **Journal of Chemical Engineering of Japan**, v.12, n.4, p. 313 – 319, 1979.

NUCCI, E.R; SILVA, R.G; GOMES, T.C; GIORDANO, R.C, CRUZ, A.J.G. A Fuzzy logic algorithm for identification of the harvesting threshold during PGA production by *Bacillus megaterium*. **Brazilian Journal of Chemical Engineering**. v.22, n.4, p. 521 – 527, 2005.

NELLES, O. Fuzzy and Neuro Fuzzy Models. In: _____. Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models. 1° ed. Springer, 2001, p. 299 – 340.

MATHWORKS. **Fuzzy Logic Toolbox User's Guide**. Natick, 2016, 367 p.

PALMA NETO, Luiz Garcia; NICOLETTI, Maria do Carmo. Considerações iniciais sobre redes neurais construtivas. In: _____. **Introdução às Redes Neurais Construtivas**. 1. ed. São Carlos: EdUFSCar, 2005. p. 17-29.

PORTAL DA SAÚDE. Ministério da Saúde. **Meningites**. Disponível em: <<http://portalsaude.saude.gov.br/index.php/o-ministerio/principal/secretarias/svs/meningites>> Acesso em: set. 2017.

REIS, G. B; HORTA, A.C.L; ZANGIROLAMI, T.C; GIORDANO, R.C; CRUZ, A.J.G. Control of Fed-Batch Yeast Cultivation Using a Capacitance Sensor. **Computer Aided Chemical Engineering**. v.27 p. 1323-1328, 2009

RIESENBERG, D; High-cell-density cultivation of *Escherichia coli*. **Current Opinion in Biotechnology**, v.2, n.3 p.380-384, 1991.

RIESENBERG, D.; GUTHKE, R. High-cell-density cultivation of microorganisms. **Applied Microbiology and Biotechnology**, v. 51, n. 4, p. 422–430, 1999.

RIGGS, A. D; GOEDDEL, D. V; KLEID, D. G; BOLIVAR, F; HEYNEKER, H. L; YANSURA, D.G; CREA, R; HIROSE, T; KRASZEWSKI A; ITAKURA, K. Expression in *Escherichia coli* of chemically synthesized genes for human insulin. **Biochemistry**, v. 76, v.1, p. 106 – 110, 1979.

SEBORG, Dale E; EDGAR, Thomas F; MELLICHAMP, Duncan A. Feedback controllers: Digital Versions of PID Controllers. In: _____. **Process Dynamics and Control**.

2. Ed. John Wiley & Sons, Inc, 2004. p. 200 – 205.

SILVA, R.G; PINOTTI, L.M; CRUZ, A.J.G; GIORDANO, R.L.C; GIORDANO, R.C.G. Artificial Neural Network to infer biomass and product concentration during the production of penicillin G acylase from *Bacillus megaterium*. **Journal of Chemical Technology and Biotechnology**. v. 83, p. 739 – 749, 2008.

SHILOACH, J.; FASS, R. Growing E. coli to high cell density - A historical perspective on method development. **Biotechnology Advances**, v. 23, n. 5, p. 345–357, 2005.

SOUSA, R; **Aplicação da lógica nebulosa no controle de reatores bioquímicos para a produção de Cefalosporina C em regime de batelada alimentada**. 1999. 143 f Dissertação (Mestrado em Engenharia Química) - Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, 1999

SOUSA, R; SOUSA, V.M. **Desenvolvimento de controlador nebuloso (“fuzzy”) para controle das vazões de ar e oxigênio no cultivo de Escherichia coli para expressão de proteínas recombinantes**. 2015. 31 f. Monografia (Trabalho de Conclusão de Curso em Engenharia Química) - Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, 2015.

TAKEMURA, R. Y; **Controle inteligente: lógica difusa**. Universidade Estadual de Maringá – UEM. Disponível em: <<http://www.din.uem.br/ia/control/index.html>>. Acesso em: 02 ago. 2016.

TORTORA, G. J; FUNKE, B.R; CASE, C.L; Biotecnologia e DNA recombinante. In _____. **Microbiologia 10**. Ed. Porto Alegre: Artmed, 2012. p 246 – 272.

ZADEH, L. A. Fuzzy Sets. **Information and Control**, v. 8, n. 3, p. 338–353, 1965.

ZADEH, L. A. The Concept of a Linguistic Variable and its Application to Approximate Reasoning - I. **Information science**, v. 8, n. 3, p. 199–249, 1975.

ZADEH, L. A. Fuzzy Logic, Neural Networks, and Soft Computing. **Communications of the ACM**, v. 37, n. 3, p. 77–84, 1994.

APÊNDICE

A. Versão final da rotina em Matlab – Simulador

```

clear all;
clc;
load arquivo2_1.txt;

dados = arquivo2_1;

%Constantes da programa??o
QmaxP = 7;
SetP = 30;
kc = 5;
ki = 0.432;
kd = 0.00485;
Fase1 = 1;
Fase2 = 5;
Fase3 = 10;

for z = 1:4 %35

    Xi(:,z) = {[dados(z,3); dados(z,6)]};
    xil(:,z) = [dados(z,4); dados(z,2); dados(z,3); dados(z,6)];

end

for i= 5:7372 %5258 %36

    if i == 5 %36

        CXP = dados(4,5); %35
        AgitMedP = dados(4,2);
        mi = dados(4,7);
        ODP = dados (4,4) ;
        QarantP = dados(4,3);
        QO2antP = dados(4,6);
        % FaseC = dados(4,8);
        DeltaT = 0.003;

        end

        if i > 5 %36

            CXP = CXPnovo;
            AgitMedP = AgitMedPnovo; %dados(i-1,2);
            mi = minovo;
            ODP = ODPnovo;
            QarantP = res(i-1,3);
            QO2antP = res(i-1,4);
            F = dados(i,9);
            V = dados(i,10);

            end

        a=readfis('DeltaS1'); %lendo a rotina fuzzy (toolbox)

```

```

DeltaP = evalfis([QO2antP],a); %fornece o valor de Delta, que pode ser
variavel

if DeltaP > 0.5;
    DeltaP = 0.5;
end

QarConta = QmaxP - QO2antP;

if QarantP >= QarConta;
    QarSeP = 0.9;
end

if QarantP < QarConta;
    QarSeP = 0.1;
end

end

b = readfis('princ1S_red5');
QarEscolha = evalfis([CXP AgitMedP ODP QarSeP mi],b);

if QarEscolha >=7;
    QarP = QarantP+1*DeltaP;
end

if QarEscolha <7; %12 %11 %13
    QarP=QarantP-1.*DeltaP;
end

end

c = readfis('Princ2S_red13');
QO2Escolha = evalfis([CXP AgitMedP ODP QarSeP mi],c); %avaliar para as
atuais entradas CXP, AgitMedP, ODP, QarSeP

if QO2Escolha >=-108; %135 %133 %145
    QO2P = QO2antP+1.*DeltaP;
end

if QO2Escolha <= -195; %190 %195 %195
    QO2P = QO2antP-1.*DeltaP;
end

if QO2Escolha > -195;
    if QO2Escolha < -108;
        QO2P = QO2antP;
    end
end

if QO2P < 0;
    QO2P = 0;
end

if QO2P>6.5;
    QO2P=6.5;
end

QO2antP = QO2P;

if QarP<0;
    QarP=0;
end

```

```

if QarP>3.5;
    QarP=3.5;
end

QarantP = QarP;

if i<=1600;
    QO2antP = 0.002;

end

res(i,3) = QarantP;
res(i,4) = QO2antP;
res(i,5) = QarEscolha;
res(i,6) = QO2Escolha;
res(i,7) = ODP;
res(i,8) = AgitMedP;
res(i,9) = CXP;
res(i,10) = mi;

X = {[res(i,3); res(i,4)]};

[Y,Xf,Af] = RedeOD1(X,Xi);

ODPnovo = Y{1,1}(1,1);

if ODPnovo < 0;
    ODPnovo =0;
end

res(3,7) = dados(3,4);
res(4,7) = dados(4,4);

AgitMedPnovo = AgitMedP + kc*((SetP-ODPnovo)-(SetP-res(i-
1,7)))+ki*(SetP-ODPnovo)+kd*((SetP-ODPnovo)-2*(SetP-res(i-1,7))+(SetP-res(i-
2,7))));

if AgitMedPnovo >= 990;
    AgitMedPnovo = 990;

end

if AgitMedPnovo <= 200;
    AgitMedPnovo = 200;
end

x1 = [ODPnovo; AgitMedPnovo; QarantP; QO2antP];

[y1,xf1] = RNeural(x1,xil);

CXPM = y1(1,1);

xil = xf1;

res(1,9) = 0.043849;
res(2,9) = 0.043849;
res(3,9) = 0.043849;
res(4,9) = 0.043849;

```

```

B = [CXPm; res(i-1,9);res(i-2,9); res(i-3,9); res(i-4,9)];
CXPnovo = mean(B);

if i < 750;
minovo = dados(i,7);

else

A = [res(i-749,9);CXPnovo];
M = mean(A);
minovo = (CXPnovo-res(i-749,9))/(i*DeltaT - (i-749)*DeltaT)*(1/M)+F/V;

if minovo < 0;
    minovo = 0;
end

if minovo > 1;
    minovo = 1;
end

end

for z = 1:3 %34

    Xi(:,z) = Xi(:,z+1);
    Xi(:,4) = X(:,1); %35

end

end

m=figure(1);
plot(dados(:,1), res(:,4), '--r');
xlabel('Tempo (h)');
ylabel('Vazao de O2 (L/min)');
hold on

plot(dados(:,1),dados(:,6), 'k');
hold on

n=figure(2);
plot(dados(:,1),res(:,3), '--r');
xlabel('Tempo (h)');
ylabel('Vazao de ar (L/min)');
hold on

plot(dados(:,1), dados(:,3), 'k');
hold on

o=figure(3);
plot(dados(:,1),res(:,7), '--r');
xlabel('Tempo (h)');
ylabel('Oxigenio Dissolvido (%)');
hold on

plot(dados(:,1), dados(:,4), 'k');
hold on

```

```
p=figure(4);
plot(dados(:,1),res(:,9),'--r');
xlabel('Tempo (h)');
ylabel('Concentracao Celular (g/L)');
hold on

plot(dados(:,1), dados(:,5), 'k');
hold on

q=figure(5);
plot(dados(:,1),res(:,8),'--r');
xlabel('Tempo (h)');
ylabel('AgitMed (rpm)');
hold on

plot(dados(:,1), dados(:,2), 'k');
hold on

r=figure(6);
plot(dados(:,1),res(:,10),'--r');
xlabel('Tempo (h)');
ylabel('mi (1/h)');
hold on

plot(dados(:,1), dados(:,7), 'k');
hold on
```

B. Valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Princ1 ajustados pelo ANFIS.

Tabela 13 - Valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Princ1 ajustados pelo ANFIS.

Função	a	b	c	d	e	f
F1	9,088	3,244	-11,740	-6,548	-66,794	-62,653
F2	35,716	4,240	-43,578	6,939	33,436	72,531
F3	76,896	-15,456	80,641	-6,892	-8,294	-10,885
F4	22,175	-7,857	40,848	-6,825	-0,903	-3,806
F5	-32,903	-4,544	-10,561	38,605	-2,711	392,744
F6	-12,778	-0,105	-2,927	6,767	46,873	71,820
F7	-50,679	2,067	55,249	-13,152	0,862	5,774
F8	-18,951	-1,434	102,541	-9,157	-1,196	-6,427
F9	42,845	21,089	101,281	-0,642	-2,107	-6,408
F10	35,400	-140,886	52,453	-0,100	-0,249	-1,171
F11	134,702	97,052	-201,207	-0,059	1,461	-0,398
F12	100,811	-109,294	-106,440	0,382	-0,099	0,362
F13	7,643	-0,164	-0,124	14,301	29,552	113,155
F14	-8,546	1,069	4,533	-33,877	-364,480	-410,246
F15	0,240	0,298	-14,441	69,545	121,699	82,941
F16	0,909	-0,325	2,185	162,509	-174,834	158,852
F17	-14,009	-0,236	-1,614	7,610	-150,543	53,309
F18	8,947	-0,507	11,860	-6,498	-274,427	-13,877
F19	3,219	0,233	-10,846	52,882	38,722	61,372
F20	-0,895	0,120	1,788	-117,980	158,165	-131,476
F21	232,514	-4,483	47,657	-0,044	-0,440	0,584
F22	38,419	2,380	-19,637	-2,136	-7,025	-10,506
F23	-22,188	15,584	-59,511	-2,369	14,067	-2,595
F24	-43,469	-1,514	24,279	-25,100	-10,999	-28,363
F25	2,374	0,004	1,356	0,994	-264,789	44,221
F26	1,467	0,244	-7,137	-4,026	52,441	-46,983
F27	0,002	0,78	5,836	-78,840	-45,347	-85,087
F28	-0,029	-0,031	-0,290	15,244	10,017	14,461
F29	-13,512	-0,396	-5,585	1,382	67,899	120,050
F30	-2,938	-0,0323	-0,757	7,239	158,059	66,855
F31	0,676	0,045	10,221	-244,367	319,653	-264,612
F32	-0,006	-0,004	-1,640	13,158	26,433	18,047
F33	7,800	7,364	-65,116	0,851	7,262	7,758

F34	-73,783	-5,787	69,467	2,919	-0,737	9,370
F35	12,060	-2,180	-38,923	1,773	18,182	2,367
F36	20,383	1,529	-21,791	45,771	14,807	51,214
F37	76,709	-45,237	221,045	0,227	-4,464	-2,781
F38	23,641	-6,928	29,946	1,082	2,766	5,701
F39	0,724	2,428	-56,448	11,626	-15,487	12,741
F40	-13,055	0,303	48,907	11,846	12,369	13,424
F41	-154,193	62,579	-82,410	4,198	4,291	25,672
F42	-58,875	1,995	128,116	2,399	2,789	6,147
F43	23,454	-4,503	-22,383	37,765	89,574	43,185
F44	-15,712	2,391	16,090	41,191	2,862	45,971
F45	7,222	61,592	9,418	0,380	0,907	0,159
F46	9,313	117,763	-3,075	0,275	0,033	0,491
F47	102,631	-46,947	18,647	8,409	20,409	9,328
F48	-110,247	38,304	-49,417	5,219	-1,629	5,810
F49	6,976	-1,490	5,777	-0,563	12,734	16,111
F50	34,047	-2,170	-8,686	-3,196	-36,478	-29,557
F51	-0,007	0,0676	2,977	-50,067	-8,295	-54,654
F52	0,0393	-0,0384	-3,378	-5,726	-66,618	-7,877
F53	-27,341	-0,0384	-3,378	-5,726	-66,618	-7,877
F54	-14,894	1,281	-20,657	0,963	-4,935	10,205
F55	1,300	-0,355	2,162	-11,953	15,792	-13,005
F56	0,242	0,082	-2,718	-1,262	279,961	-0,869
F57	-38,501	5,914	3,988	0,578	9,449	0,742
F58	-60,085	5,992	-0,030	-0,661	-2,183	-1,850
F59	12,733	-3,631	6,379	11,631	193,972	12,929
F60	5,790	0,688	-4,460	-10,994	-28,973	-12,280
F61	-1,229	0,341	-10,106	1,363	-14,428	-6,311
F62	21,214	-1,549	12,992	0,705	0,899	-3,387
F63	-0,190	-0,0173	-0,147	16,913	6,452	19,071
F64	-0,962	0,025	-0,602	24,051	-2,701	26,480
F65	-47,028	4,764	37,887	1,102	1,841	10,386
F66	-7,583	-0,073	-0,300	-0,123	30,591	8,793
F67	2,746	-0,246	-2,548	1,527	-20,555	2,230
F68	0,115	0,068	-0,020	-23,072	8,008	-25,115
F69	9,859	-35,493	-18,874	2,677	3,202	-2,338
F70	86,116	13,071	-50,494	-0,866	-1,709	0,446
F71	6,988	0,870	8,129	-56,376	54,370	-62,603
F72	11,797	-1,955	-10,114	-21,009	-45,657	-23,261

C. Valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Princ2 ajustados pelo ANFIS.

Tabela 14 - Valores dos coeficientes das funções lineares de saída do sistema de inferência nebuloso Princ2 ajustados pelo ANFIS.

Funções	a	b	c	d	e	f
F1	-29,890	-0,109	-4,644	0,764	42,032	6,135
F2	70,555	1,407	-10,534	0,123	-1,516	2,266
F3	114,530	-27,629	49,588	3,517	-0,974	4,288
F4	1,178	-5,675	142,078	-2,371	0,340	-2,482
F5	29,313	0,358	-13,911	-3,410	14,177	-35,121
F6	-16,471	-1,368	15,669	-0,1589	-1,557	-0,640
F7	-145,736	22,745	223,668	2,095	-11,824	0,075
F8	-2,754	5,834	-150,581	-2,193	1,959	-2,470
F9	-1,703	6,787	18,463	0,091	0,108	0,508
F10	-2,072	-17,439	-14,031	0,017	0,001	-0,012
F11	126,386	-90,087	-15,288	0,936	0,059	1,069
F12	45,470	143,122	15,174	0,430	0,203	0,476
F13	-8,114	-0,381	-0,514	2,818	-43,587	23,615
F14	-11,284	0,075	0,472	-2,734	9,376	-40,973
F15	3,906	0,291	-4,368	10,684	7,882	13,344
F16	1,191	-0,065	0,194	31,134	4,272	31,865
F17	5,887	0,274	0,165	-2,758	25,647	-12,312
F18	2,603	-0,088	1,235	-5,514	37,331	-16,787
F19	-1,455	0,377	-10,939	-35,283	-14,013	-39,868
F20	0,067	0,133	2,535	-88,544	24,226	-99,149
F21	-5,685	0,401	-0,649	-0,251	-0,761	-2,781
F22	222,075	-0,284	34,189	0,309	1,884	1,289
F23	22,066	10,878	-42,081	0,599	0,0225	0,485
F24	-63,657	-2,159	4,836	4,159	6,658	4,689
F25	-10,883	-0,312	-1,410	-4,391	64,929	17,387
F26	10,509	-0,06	2,324	1,485	-50,802	5,477
F27	1,292	0,229	0,798	-141,193	-11,888	-155,279
F28	-0,697	-0,003	-0,352	21,654	-54,842	24,335
F29	3,598	0,224	-5,329	7,589	-43,107	57,547
F30	-3,551	-0,001	1,522	-0,943	4,794	-1,394
F31	-0,990	-0,071	-2,059	42,708	138,875	50,986
F32	0,251	0,025	-0,662	-18,528	49,191	-20,611
F33	-54,775	0,089	-6,631	0,327	-0,783	3,524

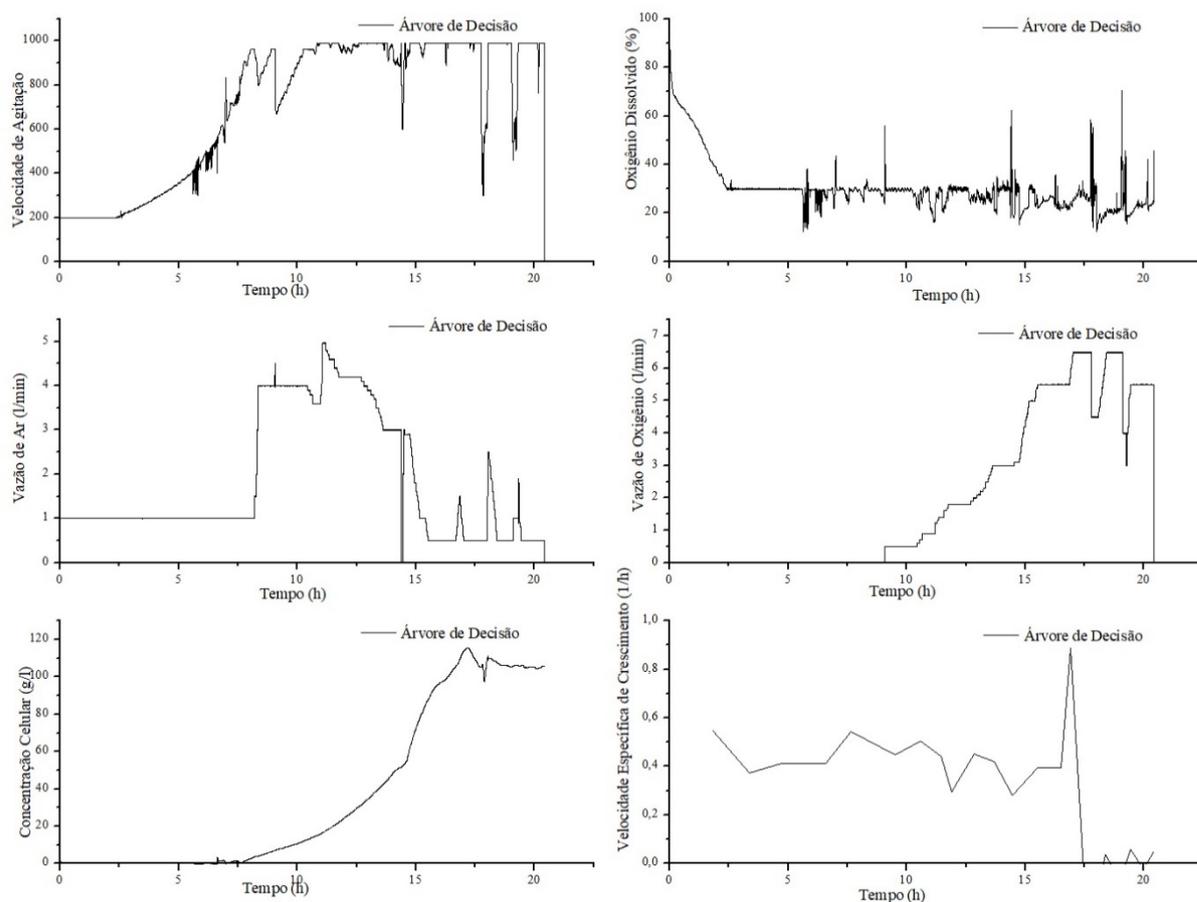
F34	121,900	1,470	-27,695	-0,117	2,693	-0,264
F35	38,972	4,486	-131,191	-0,559	27,669	-0,394
F36	-3,276	-0,411	20,603	-2,100	23,142	-2,343
F37	-27,976	31,067	12,054	-0,548	-0,313	-1,163
F38	-7,912	-17,113	76,042	-0,430	-0,079	-0,336
F39	-1,147	1,288	-46,783	-9,966	-53,513	-11,113
F40	-32,697	1,567	132,998	-9,141	0,974	-10,146
F41	42,645	-28,794	-24,086	-1,389	-6,971	-3,605
F42	-6,592	11,119	-89,326	-0,472	0,234	-0,656
F43	14,760	0,620	-38,958	-23,728	-154,905	-26,513
F44	0,169	-6,781	73,443	-9,425	6,771	-10,482
F45	3,042	-25,795	-5,383	0,202	-0,158	0,203
F46	-22,897	-54,361	-14,285	-0,054	-0,061	-0,166
F47	85,747	-13,919	-32,798	4,190	-2,725	4,654
F48	-297,917	70,734	14,855	-0,854	-0,268	-0,956
F49	-30,593	5,244	13,505	-6,411	-20,406	-18,339
F50	4,430	0,413	1,113	-1,969	-4,747	-6,280
F51	2,741	-0,238	0,448	-105,646	7,432	-118,191
F52	1,814	-0,147	-2,027	-30,959	-85,673	-34,693
F53	21,02	-4,369	9,134	-0,400	-14,620	-16,496
F54	0,192	-0,001	-0,517	4,526	3,525	4,098
F55	-1,699	0,214	0,801	28,697	-1,765	30,731
F56	-0,485	-0,166	-0,180	94,935	-39,774	105,416
F57	46,624	-11,400	-26,080	0,534	-1,444	0,086
F58	-130,506	-10,238	-38,497	0,200	1,427	-0,211
F59	3,210	-1,746	5,380	12,249	-16,364	13,573
F60	-9,746	3,702	2,467	5,092	24,725	5,627
F61	-36,735	4,450	4,437	0,305	3,471	3,061
F62	-15,047	-0,108	-14,390	1,246	0,632	2,336
F63	1,900	-0,228	-0,362	0,008	-1,347	0,204
F64	0,607	-0,030	0,854	23,350	6,561	26,013
F65	10,638	-3,339	58,805	0,090	-4,660	2,060
F66	16,635	0,314	-23,032	1,350	5,751	2,748
F67	-0,587	0,180	-2,608	-2,664	-12,166	-2,819
F68	-0,818	-0,062	1,125	24,799	-0,629	27,644
F69	50,911	9,664	16,946	-0,355	10,581	0,084
F70	31,482	-14,684	31,881	-1,325	2,083	-0,680
F71	-4,705	-1,144	12,779	-8,384	218,915	-9,281
F72	3,664	0,305	-11,201	-28,990	30,490	-32,154

ANEXO

A. Dados experimentais (controlados utilizando a árvore de decisão) do cultivo A16.

Na Figura 44 tem-se os dados experimentais de velocidade de agitação, oxigênio dissolvido, vazões de ar e oxigênio, concentração celular e velocidade específica de crescimento, todos em função do tempo, controlados utilizando a árvore de decisão.

Figura 44 - Dados experimentais do cultivo A16 (controlado pela árvore de decisão)

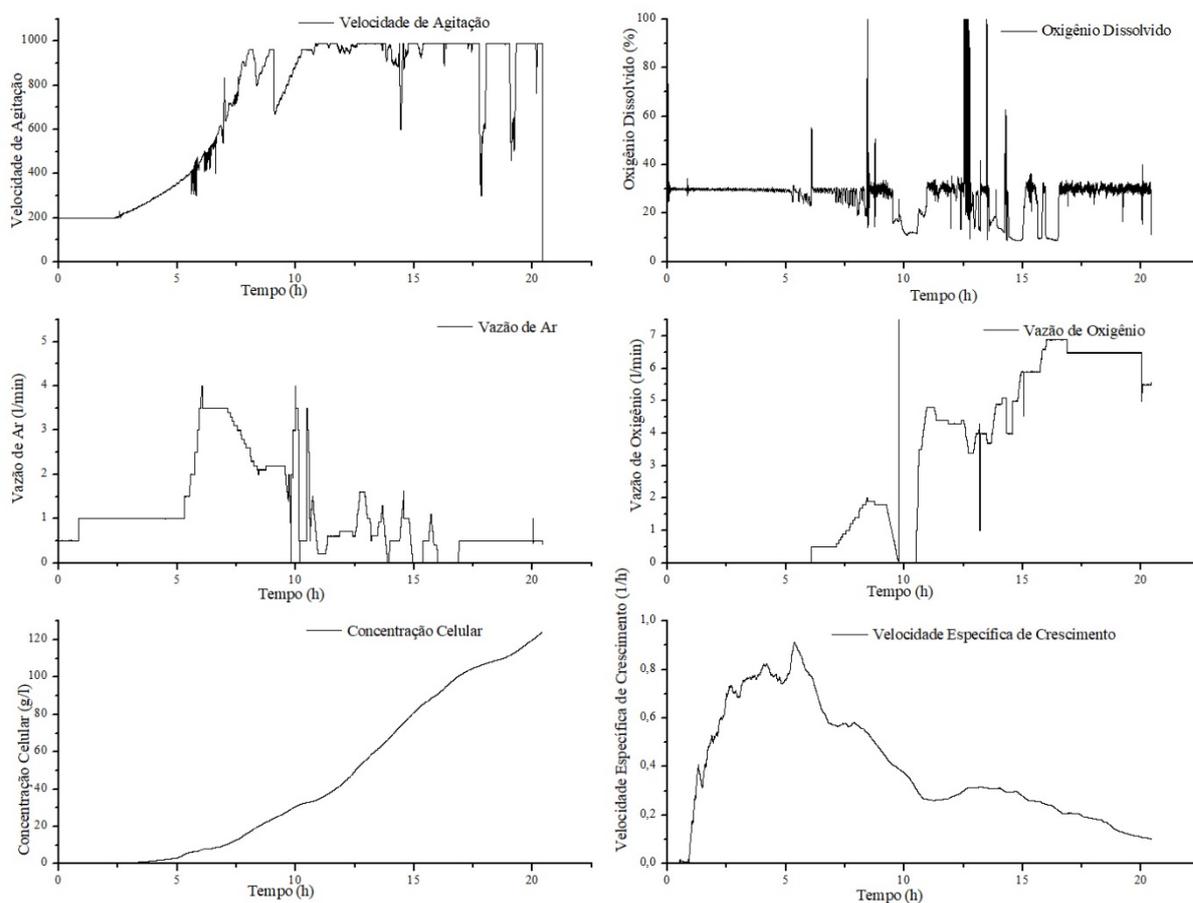


Fonte: HORTA, 2011

B. Dados experimentais (controlados utilizando a árvore de decisão) do cultivo A12.

Na Figura 45 tem-se os dados experimentais de velocidade de agitação, oxigênio dissolvido, vazões de ar e oxigênio, concentração celular e velocidade específica de crescimento, todos em função do tempo, controlados utilizando a árvore de decisão.

Figura 45 - Dados experimentais do cultivo A12 (controlado pela árvore de decisão)



Fonte: HORTA, 2011