

**TATIANA LIE FUJIKI**

**DESENVOLVIMENTO DE CONTROLADOR DIGITAL  
BASEADO EM UM MODELO NEURAL E UM OTIMIZADOR  
DE UMA PLANILHA ELETRÔNICA**

Dissertação de Mestrado apresentada à Faculdade de Engenharia Química da Universidade Estadual de Campinas, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Química.

Área de Concentração: Sistemas de Processos Químicos e Informática

Orientador: Ana Maria Frattini Fileti

**Campinas – SP**

**Março – 2009**

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE -  
UNICAMP

F95d Fujiki, Tatiana Lie  
Desenvolvimento de controlador digital baseado em um modelo neural e um otimizador de uma planilha eletrônica / Tatiana Lie Fujiki. --Campinas, SP: [s.n.], 2009.

Orientador: Ana Maria Frattini Fileti.  
Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Química.

1. Redes neurais (Computação). 2. Comunicação digital - Inovações tecnológicas. 3. Controle de processos químicos. 4. Planilhas eletrônicas. I. Fileti, Ana Maria Frattini. II. Universidade Estadual de Campinas. Faculdade de Engenharia Química. III. Título.

Título em Inglês: Development of a digital controller based on a neural model and an optimizer of an electronic worksheet

Palavras-chave em Inglês: Artificial neural networks, Fieldbus interface, Digital process control, Electronic worksheet

Área de concentração: Sistemas de Processos Químicos e Informática

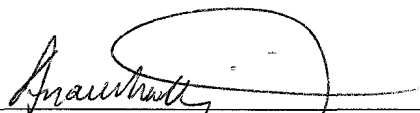
Titulação: Mestre em Engenharia Química

Banca examinadora: Antonio José Gonçalves da Cruz, Christianne Elisabete da Costa Rodrigues, Flavio Vasconcelos da Silva

Data da defesa: 20/03/2009

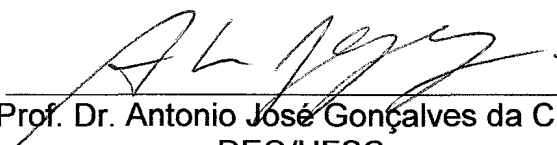
Programa de Pós Graduação: Engenharia Química

Dissertação de Mestrado defendida por Tatiana Lie Fujiki e aprovada em 20 de março de 2009 pela banca examinadora constituída pelos doutores:



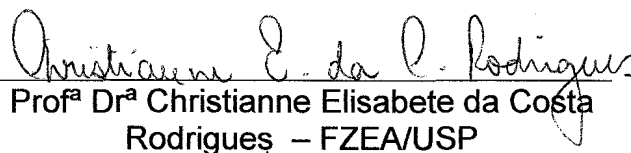
---

Prof<sup>a</sup> Dr<sup>a</sup> Ana Maria Frattini Fileti –  
FEQ/UNICAMP (Orientadora)



---

Prof. Dr. Antonio José Gonçalves da Cruz –  
DEQ/UFSCar



---

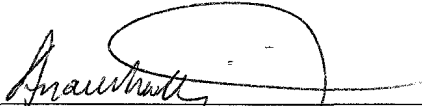
Prof<sup>a</sup> Dr<sup>a</sup> Christianne Elisabete da Costa  
Rodrigues – FZEA/USP



---

Prof. Dr. Flavio Vasconcelos da Silva –  
FEQ/UNICAMP

Este exemplar corresponde à versão final da Dissertação de Mestrado em Engenharia Química, defendida por Tatiana Lie Fujiki, em 20 de março de 2009.



---

Prof<sup>a</sup> Dr<sup>a</sup> Ana Maria Frattini Fileti –  
FEQ/UNICAMP (Orientadora)

Dedico esta dissertação aos meus pais Jorge e Elisabeth, que nunca pouparam esforços para a minha educação e à minha avó Teruko (*In memoriam*), que deixa saudades.

*“A melhor de todas as coisas é aprender. O dinheiro pode ser perdido ou roubado, a saúde e a força podem falhar, mas o que você dedicou à sua mente é seu para sempre.”*

*Louis L'Amour*

## **AGRADECIMENTOS**

Agradeço aos professores do Departamento de Engenharia de Sistemas Químicos Ana Maria Frattini Fileti e Flávio Vasconcelos da Silva pela orientação, apoio, ensinamentos, conselhos e amizade.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro.

Aos meus pais Jorge e Elisabeth e às minhas irmãs Adriana, Luciana e Fernanda pela formação do meu caráter e por todo o apoio com que sempre pude e sempre poderei contar. A família é essencial (inclusive para o fornecimento de cascas de abacaxi).

Aos meus colegas de laboratório pelas contribuições, por compartilhar momentos de descontração e de dificuldades e que de uma forma ou de outra sempre tinham um palpite para ajudar: Camila Schulz, Ivan MATLAB Franco, Luiz Ferrarezi, Manuela Leite, Marcelo Piá, Ricardo Almeida, Thiago Araxá Costa, Thiago Pinelli e Wagner Beicin de Pulga Campos. Considero vocês profissionais extremamente competentes e batalhadores. Com cada um de vocês aprendi coisas que levarei para toda a vida. Em especial, agradeço às amigas Cá e Manu e aos meninos Luiz e Thiago Araxá, que contribuíram demais, tendo sido colegas de projeto. Com certeza as discussões a respeito dos nossos trabalhos foram bastante enriquecedoras, interessantes e mostraram que o trabalho em equipe existe e funciona. Muito obrigada!

Não poderia deixar de agradecer aos alunos de Iniciação Científica: Atahualpa, Márcia Sanzovo e Rodrigo Paiva, pela ajuda no desenvolvimento do projeto e pela oportunidade de compartilhar o meu conhecimento, e assim me ajudando a

compreender minha própria linha de pensamento e à Francine, pela ajuda com o sistema de destilação.

Ao professor Roger Josef Zemp, pelos palpites e orientações sobre o Solver.

Ao professor Luiz Carlos Bertevello pelas sugestões e pela amizade.

Aos técnicos da oficina da Faculdade de Engenharia Química Alexandre, Daniel, Edgar, ao analista de sistemas do DESQ Juliano, à secretária do DESQ Valquíria e aos funcionários da Secretaria de Pós-graduação Evandro, Gabriel, Márcia e Marcos.

Ao Daniel, meu querido amor, por toda a paciência, pelo apoio, revisões de textos, conselhos e ajuda em todos os momentos.

À minha segunda família: Glória, José, Márcia, Milena e Rafael pelo apoio, carinho, cuidado e pelo fornecimento de cascas de abacaxi!

Aos amigos Andrea, Danilo, Paula, Renata, Thaís e Vivian, que estiveram sempre na torcida!

*“Quero, um dia, dizer às pessoas que nada foi em vão... Que o amor existe, que vale a pena se doar às amizades e às pessoas, que a vida é bela sim e que eu sempre dei o melhor de mim... E que valeu a pena.”*

*Mario Quintana*



## RESUMO

A demanda de mercado com relação à qualidade dos produtos, à produtividade e lucratividade dos processos e à questão ambiental tem forçado as indústrias a buscar melhorias nos processos. Com esta finalidade, os engenheiros têm buscado no Controle de Processos soluções para sistemas não lineares e/ou transientes. Sistemas de controle avançados, baseados em Redes Neurais Artificiais (RNA), por serem adequados para modelar relações não lineares, podem ser indicados como soluções para estes casos. Neste trabalho, as equações algébricas de um modelo neural foram implementadas numa planilha do Microsoft Excel. O modelo neural foi treinado com o algoritmo de Levenberg-Marquardt, utilizando o *toolbox* de redes neurais do MATLAB e os pesos e *bias* encontrados foram inseridos na planilha. O monitoramento do sistema foi feito através de uma rede digital *Fieldbus*, configurada através do *software* Syscon. O sistema supervisor utilizado foi o Indusoft Web Studio, sendo a comunicação entre estes dois programas feita através do protocolo *Ole for Process Control* (OPC). A atualização das variáveis de entrada do modelo neural foi realizada através do sistema *Open Data Base Connectivity* (ODBC), responsável pela troca de informações entre o Indusoft e o Excel. A variável controlada um instante de amostragem à frente é automaticamente calculada a cada instante de amostragem e usada como função-objetivo no *Solver* do Excel (minimizar o erro quadrático). Restrições foram adicionadas ao método do gradiente descendente generalizado para que uma solução adequada fosse encontrada dentro de uma faixa de valores da variável manipulada. Como caso-estudo, este sistema de controle alternativo foi utilizado para controlar a temperatura interna de um tanque de precipitação de bromelina com etanol a frio, para evitar desnaturação da enzima. O processo de precipitação a partir de caldo de abacaxi foi realizado num tanque encamisado operando em regime de batelada alimentada, com agitação constante. O controle da temperatura do meio precipitante é feito através da manipulação da rotação da bomba de fluido refrigerante (propilenoglicol). Dados experimentais (75% do banco de dados de ensaios em malha aberta) foram utilizados para treinar a rede *feed forward* multicamadas. As variáveis de entrada da RNA, atualizadas a cada 4 segundos, são: temperatura do etanol, temperatura do meio precipitante (variável controlada), temperaturas de entrada e de saída do propilenoglicol, nível de líquido, rotação da bomba de propilenoglicol (variável manipulada) e degrau na bomba de propilenoglicol. Na camada intermediária, a função de ativação tangente hiperbólica foi aplicada a 14 neurônios. A variável de saída da rede é a temperatura do meio um instante de amostragem à frente, com a função de ativação linear. Testes *offline* com dados não vistos no treinamento mostraram que a RNA foi capaz de prever a temperatura do tanque, já que os coeficientes angular e linear dos gráficos de dispersão (saída calculada *versus* saída real) aproximavam-se de um e zero, respectivamente. O controlador alternativo foi implementado e os resultados experimentais mostraram sua efetividade em manter a temperatura do meio em torno do *set-point* (5°C).

Palavras-chave: redes neurais artificiais, rede digital *Fieldbus*, controle de processos digital, planilha eletrônica.

## ABSTRACT

Intelligent systems, based on Artificial Neural Networks (ANN) have been studied as alternative solutions for ill-defined plants or nonlinear and transient systems, due to the worldwide market requirements for high standard products and safe and environmentally friendly processes. In this work, the algebraic equations of a neural model were implemented in a Microsoft Excel worksheet. This ANN was previously trained with Levenberg-Marquardt algorithm, using the ANN toolbox of MATLAB and the optimized weights and biases were inserted in the worksheet as well. By means of a digital Fieldbus network architecture, configured by Syscon Software, the data acquisition and process control were performed. The Indusoft Web Studio was employed as supervisory software and the communication between these two softwares was achieved through Ole for Process Control protocol. Since the data acquisition system is supposed to update the ANN input variables in the worksheet, information is exported from Indusoft to Excel through Open Data Base Connectivity system, which was properly configured. The one-step-ahead controlled variable is automatically predicted every sample time and then used in the objective function of the Excel Solver (quadratic error minimization), implemented via Visual Basic programming. Restrictions were added to the quasi-Newton optimization method so that a smooth solution could be found inside a suitable range of the manipulated variable. As a case-study, this alternative system was implemented to control the temperature of a bromelain precipitation tank. The bromelain recovery from pineapple juice was carried out in a fed-batch stirred tank, using precipitation with ethanol at low temperature to avoid protein denaturation. The coolant (propylene glycol) flow rate was manipulated through a variable speed pump. Experimental data (75% of the database from open-loop tests) was used to train a multi-layered feed forward ANN. The ANN input layer variables, updated every 4 seconds, were chosen as follows: ethanol temperature, bulk temperature (controlled variable), coolant inlet temperature and coolant outlet temperature, liquid volume, pump flow rate (manipulated variable) and pump flow rate variation. In the hidden layer a hyperbolic tangent function was applied to 14 neurons. The output variable was the one-step-ahead bulk temperature, using the linear activation function. Offline tests with unseen data proved that the ANN successfully predicted the tank temperature, as the linear fitting slope coefficient of the dispersion plots (network output *versus* target vector) was close to the unity and the interception coefficient was around zero. The alternative controller was implemented and experimental results showed its effectiveness in maintaining the bulk temperature around the set-point (5°C). From the present results it was clearly shown that this methodology is a promising new way to face complex process control problems and the tendency is to increase their range of applicability in industrial processes.

Keywords: artificial neural networks, Fieldbus interface, digital process control, electronic worksheet.

## ÍNDICE DE TABELAS

Tabela 2.1: Comparação entre controle <i>feedback</i> e <i>feed forward</i> .....	6
Tabela 4.1: Comparação entre os modelos testados.....	56
Tabela 4.2: Comparação entre o Modelo Neural 4b e o Modelo Neural 6.....	84
Tabela 4.3: Comparação entre os parâmetros do controlador PID na etapa de sintonia fina .....	93
Tabela 4.4: Comparação entre os parâmetros dos controladores PID e alternativo e do ensaio em malha aberta .....	95

## ÍNDICE DE FIGURAS

Figura 2.1: Processo de aprendizagem da rede neural artificial .....	10
Figura 3.1: Etapas envolvidas no desenvolvimento do modelo neural e os <i>softwares</i> utilizados....	27
Figura 3.2: Comunicação entre os elementos do processo .....	28
Figura 3.3: Estrutura de rede estudada .....	30
Figura 3.4: Etapas para predição da temperatura do meio precipitante utilizando o modelo neural na planilha eletrônica.....	32
Figura 3.5: Modelo neural desenvolvido em planilha eletrônica .....	32
Figura 3.6: Interface para configuração do Microsoft Excel <i>Solver</i> .....	35
Figura 3.7: Fluxo de informações no acoplamento do modelo neural com o otimizador.....	36
Figura 3.8: Fluxograma de instrumentação do protótipo de precipitação de bromelina.....	37
Figura 3.9: Visão geral do sistema experimental .....	38
Figura 3.10: Painel elétrico utilizado no sistema experimental .....	39
Figura 3.11: Topologia da rede <i>Fieldbus</i> utilizada (barramento com <i>spurs</i> ).....	44
Figura 3.12: Estratégia de controle configurada no Syscon.....	45
Figura 3.13: Diagrama da montagem utilizada nos ensaios de sintonia do controlador PID. ....	47
Figura 3.14: Curva de reação para determinação dos parâmetros do processo para sintonia através do método de Ziegler-Nichols.....	48
Figura 4.1: Comportamento da variável controlada com a variável manipulada fixa em 40% de rotação .....	49
Figura 4.2: Perturbações na variável manipulada (ensaios em malha aberta) com (a) degraus uniformes e (b) degraus variados .....	51
Figura 4.3: Ensaio em malha aberta com a bomba desligada.....	51
Figura 4.4: Ensaio em malha aberta com desligamentos e religamentos da bomba .....	52
Figura 4.5: Tela inicial do sistema supervisório.....	53
Figura 4.6: Tela do controlador PID do sistema supervisório .....	53
Figura 4.7: Tela do controlador neural do sistema supervisório .....	54
Figura 4.8: Tela de gráficos do sistema supervisório .....	54

Figura 4.9: Estrutura da RNA para o Modelo Neural 1. ....	57
Figura 4.10: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 1.....	57
Figura 4.11: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 1. ....	58
Figura 4.12: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 1.....	58
Figura 4.13: Estrutura da RNA para o Modelo Neural 2. ....	59
Figura 4.14: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 2.....	60
Figura 4.15: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 2. ....	60
Figura 4.16: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 2.....	61
Figura 4.17: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 3.....	62
Figura 4.18: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 3. ....	62
Figura 4.19: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 3.....	63
Figura 4.20: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha aberta, no teste do Modelo Neural 3.....	64
Figura 4.21: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha fechada, no teste do Modelo Neural 3.....	65
Figura 4.22: Estrutura da RNA para o Modelo Neural 4a.....	66
Figura 4.23: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 4a.....	66
Figura 4.24: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 4a. ....	67
Figura 4.25: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 4a. ....	67

Figura 4.26: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha aberta, no teste do Modelo Neural 4a.....	68
Figura 4.27: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha fechada, no teste do Modelo Neural 4a.....	69
Figura 4.28: Comparação entre a temperatura real e a predita pela RNA na validação <i>online</i> do Modelo Neural 4a. ....	70
Figura 4.29: Análise da validação <i>online</i> do Modelo Neural 4a. ....	71
Figura 4.30: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 4b.....	72
Figura 4.31: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 4b. ....	72
Figura 4.32: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 4b. ....	73
Figura 4.33: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha aberta, no teste do Modelo Neural 4b.....	74
Figura 4.34: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha fechada, no teste do Modelo Neural 4b. ....	75
Figura 4.35: Comparação entre a temperatura real e a predita pela RNA na validação <i>online</i> do Modelo Neural 4b. ....	75
Figura 4.36: Análise da validação <i>online</i> do Modelo Neural 4b.....	76
Figura 4.37: Estrutura da RNA para o Modelo Neural 5. ....	78
Figura 4.38: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 5. ....	79
Figura 4.39: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 5. ....	79
Figura 4.40: Comparação entre a saída real e a calculada pela RNA para um ensaio em malha aberta (a) e para um ensaio em malha fechada (b), no teste do Modelo Neural 5. ....	80
Figura 4.41: Estrutura da RNA para o Modelo Neural 6. ....	81
Figura 4.42: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 6. ....	81
Figura 4.43: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 6. ....	81

Figura 4.44: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha aberta, no teste do Modelo Neural 6.....	82
Figura 4.45: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha fechada, no teste do Modelo Neural 6.....	83
Figura 4.46: Ensaio de controle utilizando a restrição: $MV_{k+1} = [0;100]\%$ ..	85
Figura 4.47: Ensaio de controle utilizando as restrições: $MV_{k+1} = [0;100]\%$ e $\Delta MV_k = \pm 20\%$ ..	86
Figura 4.48: Ensaio de controle com as restrições: $MV_{k+1} = [0;100]\%$ ; $\Delta MV_k = \pm 20\%$ e $T_{k+1} = 4,0^\circ C \Rightarrow MV_{k+1} = 0$ ..	87
Figura 4.49: Ensaio de controle com as restrições: $MV_{k+1} = [0;100]\%$ ; $\Delta MV_k = \pm 20\%$ e $T_{k+1} = 4,9^\circ C \Rightarrow MV_{k+1} = 0$ ..	87
Figura 4.50: Ensaio de controle com as restrições: $MV_{k+1} = [0;100]\%$ ; $\Delta MV_k = \pm 35\%$ e $T_{k+1} = 4,9^\circ C \Rightarrow MV_{k+1} = 0$ ..	88
Figura 4.51: Curva de resposta da variável controlada frente a uma perturbação em degrau na variável manipulada. ....	89
Figura 4.52: Análise de estabilidade - método do <i>Root Locus</i> . ....	90
Figura 4.53: Ensaio de controle – sintonia fina do PID, utilizando os parâmetros: $K_c = 7\%/^\circ C$ , $\tau_I = 6s$ e $\tau_D = 1,5s$ ..	91
Figura 4.54: Ensaio de controle – sintonia fina do PID, utilizando os parâmetros: $K_c = 35\%/^\circ C$ , $\tau_I = 18s$ e $\tau_D = 1,5s$ ..	91
Figura 4.55: Ensaio de controle – sintonia fina do PID, utilizando os parâmetros: $K_c = 35\%/^\circ C$ , $\tau_I = 28s$ e $\tau_D = 1,5s$ ..	92
Figura 4.56: Ensaio de controle – sintonia fina do PID, utilizando os parâmetros: $K_c = 35\%/^\circ C$ , $\tau_I = 28s$ e $\tau_D = 7s$ ..	93
Figura 4.57: Comparação entre os controladores PID e alternativo e um ensaio em malha aberta. ....	94
Figura 4.58: Comparação entre os controladores PID e alternativo e o ensaio em malha aberta...	96
Figura A.1: Primeiro passo para configuração do <i>driver</i> ODBC.....	109
Figura A.2: Segundo passo para configuração do <i>driver</i> ODBC.....	110

Figura A.3: Terceiro passo para configuração do <i>driver</i> ODBC .....	110
Figura A.4: Quarto passo para configuração do <i>driver</i> ODBC .....	111
Figura A.5: Quinto passo para configuração do <i>driver</i> ODBC .....	111
Figura A.6: Sexto passo para configuração do <i>driver</i> ODBC.....	112
Figura A.7: Sétimo passo para configuração do <i>driver</i> ODBC .....	112
Figura A.8: Oitavo passo para configuração do <i>driver</i> ODBC.....	113
Figura A.9: Nono passo para configuração do <i>driver</i> ODBC.....	113
Figura A.10: Décimo passo para configuração do <i>driver</i> ODBC .....	114
Figura A.11: Configuração da planilha ODBC no Indusoft, para escrever valores do Indusoft no Excel .....	115
Figura A.12: Configuração da planilha ODBC no Indusoft, para ler valores do Excel, atribuindo-os a <i>tags</i> no Indusoft .....	116
Figura C.1: Curva de calibração da bomba de propilenoglicol.....	123
Figura C.2: Curva de calibração da bomba peristáltica .....	124
Figura C.3: Curva de calibração do sensor de temperatura do álcool .....	124
Figura C.4: Curva de calibração do sensor de temperatura do meio precipitante.....	125
Figura C.5: Curva de calibração do sensor de temperatura de entrada do propilenoglicol .....	125
Figura C.6: Curva de calibração do sensor de temperatura de saída do propilenoglicol .....	126
Figura D.1: Curva de calibração do refratômetro para análise das amostras de resíduo da precipitação de bromelina .....	127



## SUMÁRIO

CAPÍTULO 1: INTRODUÇÃO .....	1
1.1. Motivação.....	1
1.2. Objetivo .....	3
1.3. Organização do texto .....	3
CAPÍTULO 2: FUNDAMENTAÇÃO TEÓRICA E REVISÃO BIBLIOGRÁFICA.....	5
2.1. Controle de Processos.....	5
2.2. Redes Neurais Artificiais (RNA) .....	8
2.2.1. Definição .....	8
2.2.2. Arquiteturas da RNA.....	9
2.2.3. Treinamento da RNA.....	9
2.2.4. Algoritmo de retropropagação .....	9
2.2.5. Função de ativação.....	10
2.2.6. Aplicações.....	11
2.2.7. Vantagens e limitações .....	12
2.3. Bromelina .....	12
2.3.1. Definição .....	12
2.3.2. Aplicações.....	13
2.3.3. O Processo de Precipitação .....	13
2.4. Estado da arte .....	14
2.4.1. Redes Neurais Artificiais.....	14
2.4.2. Bromelina .....	18
2.5. Otimização.....	19
2.6. Comunicação Digital <i>Fieldbus</i> .....	22
2.7. Conclusões do capítulo .....	24
CAPÍTULO 3: MATERIAIS E MÉTODOS.....	27

3.1. Desenvolvimento do Modelo Neural .....	27
3.1.1. Configuração de softwares.....	28
3.1.2. Ensaios em Malha Aberta.....	29
3.1.3. Treinamento e Testes da RNA.....	30
3.1.4. Validação online do modelo neural .....	31
3.2. Configuração do otimizador.....	34
3.3. Caso-estudo: precipitação de bromelina .....	36
3.3.1. Protótipo da planta de precipitação de bromelina .....	36
3.3.2. Preparo das amostras.....	39
3.3.3. Procedimento para os ensaios em malha aberta.....	40
3.3.4. Procedimento para os ensaios em malha fechada .....	41
3.3.5. Tratamento de resíduos .....	42
3.4. Calibração de bombas e sensores.....	43
3.4.1. Bombas.....	43
3.4.2. Sensores de temperatura.....	43
3.5. Rede <i>Fieldbus</i> .....	44
3.6. Desenvolvimento e Sintonia do Controlador Convencional PID.....	45
3.7. Comparação entre os controladores.....	48
CAPÍTULO 4: RESULTADOS E DISCUSSÕES .....	49
4.1. Construção do banco de dados – ensaios em malha aberta .....	49
4.2. Desenvolvimento do Sistema Supervisório.....	52
4.3. Modelagem do processo por Redes Neurais Artificiais .....	55
4.3.1. Modelo Neural 1 .....	55
4.3.2. Modelo Neural 2 .....	59
4.3.3. Modelo Neural 3 .....	61
4.3.4. Modelo Neural 4a.....	65

4.3.5. Modelo Neural 4b .....	70
4.3.6. Modelo Neural 5 e Modelo Neural 6 .....	78
4.4. Ensaio de Controle.....	84
4.5. Sintonia do Controlador PID.....	89
4.6. Comparação entre os Controladores .....	94
CAPÍTULO 5: CONCLUSÕES.....	97
5.1. Redes Neurais Artificiais.....	97
5.2. Uso da planilha Excel.....	99
5.3. Comparação entre os Controladores .....	100
5.4. Bromelina .....	100
CAPÍTULO 6: SUGESTÕES PARA TRABALHOS FUTUROS.....	101
CAPÍTULO 7: REFERÊNCIAS BIBLIOGRÁFICAS .....	103
APÊNDICE A: CONFIGURAÇÃO DO DRIVER ODBC .....	109
APÊNDICE B: <i>Scripts</i> de Programação Utilizados no Desenvolvimento do Projeto .....	119
APÊNDICE C: Curvas de Calibração de Bombas e Sensores .....	123
C.1. Calibração da bomba de deslocamento positivo.....	123
C.2. Calibração da bomba peristáltica.....	123
C.3. Calibração das termorresistências .....	124
APÊNDICE D: Tratamento de Resíduos .....	127

# CAPÍTULO 1: INTRODUÇÃO

## 1.1. Motivação

Alternativas ao controle convencional utilizado comumente nas indústrias vêm sendo pesquisadas para trazer melhorias às operações industriais, devido às crescentes exigências do mercado em termos de qualidade, segurança (do trabalho e ambiental) e ao aumento da complexidade dos processos. Controladores avançados que utilizam redes neurais artificiais e lógica *fuzzy* podem ser indicados como soluções alternativas para estes problemas que apresentam não-linearidades e/ou transitoriedade.

A aplicação de sistemas inteligentes nas indústrias será cada vez mais necessária para uma melhoria contínua, muito embora haja ainda grande resistência à sua aceitação. Isto se deve ao fato de os controladores convencionais atenderem muito bem às atuais necessidades dos processos industriais (MARTINS e COELHO, 2000). Outro fator que contribui grandemente para esta resistência é a facilidade de implementação dos controladores convencionais nas indústrias por mão-de-obra não especializada. Assim, a aplicação de controladores avançados na indústria é marginal, visando geralmente à melhoria dos controladores existentes.

Estudos bem-sucedidos de redes neurais aplicadas à modelagem e ao controle de bioprocessos (podendo ser citado o processo de fermentação alcoólica) mostram a importância desta ferramenta como modelo não-linear.

A grande vantagem da aplicação de Redes Neurais Artificiais reside na capacidade destas de modelar relações não lineares entre variáveis de entrada e de saída dos processos, utilizando cálculos algébricos simples. Segundo Lennox et al. (2001), o potencial de aplicação é tão grande que o Departamento de Comércio e Indústria do Reino Unido (*UK Department of Trade and Industry*) iniciou uma campanha de £5,7 milhões, destinada ao incentivo para divulgação desta tecnologia nas indústrias.

Porém, para a aplicação desta ferramenta, muitas vezes as indústrias se vêem reféns de códigos computacionais complexos e *softwares* comerciais caros.

Assim, neste trabalho procurou-se mostrar uma alternativa de fácil implementação desta promissora tecnologia, utilizando um *software* amplamente divulgado e disponível, como o Microsoft Excel.

Como estudo de caso para aplicação do controlador desenvolvido, escolheu-se trabalhar com o processo de precipitação de bromelina devido ao alto valor agregado desta enzima e sua importante atividade terapêutica, aliado ao fato de se utilizar resíduos industriais (casca e talo do abacaxi). A bromelina é uma enzima proteolítica encontrada em diversas partes do abacaxi, cuja produção brasileira encontra-se na primeira colocação no mercado mundial (CUNHA, 2007). Vale destacar ainda, que existe uma perspectiva de crescimento de mercado de produtos processados, elaborados a partir de ingredientes naturais, como as frutas, devido a mudanças no modo de vida da população (GRANADA, ZAMBIAZI e MENDONÇA, 2004), o que conseqüentemente aumenta as perspectivas de crescimento do mercado de subprodutos.

O processo em estudo foi realizado em batelada e por esta razão, tem-se um estado transiente, com as condições do processo variando com o tempo. Uma outra característica, verificada por Leite (2007), é a não linearidade, pois não foi observada nenhuma simetria nas curvas de reação para perturbações positivas e negativas de mesma amplitude na variável manipulada. A conseqüência observada para estas duas características é a mesma: os parâmetros do processo (ganho ( $K_p$ ), constante de tempo ( $\tau_p$ ) e tempo morto ( $t_d$ )) mudam com o progresso da batelada, acarretando em mudanças também nos parâmetros de um controlador PID, já que estes são calculados a partir daqueles. Portanto, um controlador PID de parâmetros fixos não seria o mais indicado para controlar este processo, sendo indicados como soluções controladores adaptativos (que possuam parâmetros variando com o avanço da batelada) ou controladores baseados em modelo não linear, daí a proposta deste trabalho de se modelá-lo utilizando Redes Neurais Artificiais.

## 1.2. Objetivo

Desenvolver e implementar um controlador utilizando modelagem por Redes Neurais Artificiais e um otimizador de uma planilha eletrônica.

Comparar o desempenho do controlador desenvolvido com o de um controlador convencional PID. Para tanto, critérios como parâmetros das curvas de comportamento das variáveis controlada e manipulada devem ser utilizados, assim como o critério de desempenho ITAE (integral do erro absoluto vezes tempo).

## 1.3. Organização do texto

Esta dissertação contém 7 capítulos, assim divididos:

No Capítulo 1, apresenta-se a justificativa da escolha do tema do trabalho, com foco na escolha da técnica de controle aplicada (redes neurais artificiais) e uma breve explicação sobre o caso-estudo (bromelina). Os objetivos principais e secundários também são enunciados neste capítulo.

No Capítulo 2, realizou-se uma revisão bibliográfica, na qual são mostrados os avanços recentes nesta área, buscando-se relacionar o tema estudado com trabalhos anteriores. A base teórica para o desenvolvimento da dissertação é brevemente recapitulada, sendo apresentados alguns conceitos fundamentais envolvidos com a teoria de redes neurais artificiais, controle de processos e otimização.

O Capítulo 3 contém a descrição e o detalhamento de todas as etapas do projeto, assim como o sistema experimental, os materiais e *softwares* utilizados.

No Capítulo 4 são mostrados os resultados obtidos em cada fase do projeto e seus respectivos comentários, discussões e análises.

As conclusões compõem o Capítulo 5 e o Capítulo 6 traz algumas sugestões para trabalhos futuros.

Por fim, o Capítulo 7 traz as referências bibliográficas utilizadas.

Os Apêndices encontram-se nas páginas finais.

# CAPÍTULO 2: FUNDAMENTAÇÃO TEÓRICA E REVISÃO BIBLIOGRÁFICA

Neste capítulo são abordados conceitos fundamentais a respeito de controle de processos, de redes neurais artificiais, de otimização, da enzima bromelina e da tecnologia digital *Fieldbus*. Também foi realizada uma revisão da literatura sobre aplicações de redes neurais artificiais em controle de processos e uma breve revisão sobre a enzima bromelina.

## 2.1. Controle de Processos

No mundo atual, a motivação para melhoria dos processos industriais sempre está voltada para o lado financeiro. Assim, com o aumento no volume de produção das indústrias e com a necessidade de aumento da competitividade, os engenheiros têm recorrido ao controle de processos para manter a operabilidade dos mesmos, aumentando a produtividade com perdas reduzidas.

O controle pode ser aplicado num processo com diversos objetivos: manutenção das condições operacionais, impedindo que determinadas variáveis fiquem fora da faixa de operação, mantendo a qualidade do produto e atendendo a especificações da legislação ambiental.

Dois estratégias de controle merecem destaque: *feedback* (controle por realimentação) e *feed forward* (controle por antecipação). A diferença fundamental entre as duas estratégias é que a ação de controle é tomada depois que o erro acontece (*feedback*) ou antes que o erro aconteça (*feed forward*). No controle *feedback*, a variável controlada é medida e a partir da comparação deste valor com um valor de referência (*set-point*), implementa-se a ação de controle (variável manipulada). Já no controle *feed forward*, a medida é feita no distúrbio do processo, inferindo-se um erro que irá ocorrer e a partir daí calcula-se a ação de controle (SEBORG, EDGAR e MELLICHAMP, 1989). Desta forma, é possível

detectar as vantagens e desvantagens de cada tipo de estratégia, como mostrado na Tabela 2.1.

Tabela 2.1: Comparação entre controle *feedback* e *feed forward*

	<i>Feedback</i>	<i>Feed forward</i>
VANTAGEM	<ul style="list-style-type: none"> <li>- a ação corretiva ocorre a partir de um desvio do <i>set-point</i>, independentemente do distúrbio (não requer medição do distúrbio)</li> <li>- não requer modelo matemático do processo</li> <li>- o controlador PID, amplamente utilizado, pode ser facilmente re-sintonizado em caso de alterações no processo</li> </ul>	<ul style="list-style-type: none"> <li>- a ação corretiva ocorre a partir da medição do distúrbio e assim, o erro nem chega a ocorrer</li> <li>- não introduz instabilidade no sistema em malha fechada</li> </ul>
DESVANTAGEM	<ul style="list-style-type: none"> <li>- o erro necessariamente ocorre para que ocorra a ação corretiva</li> <li>- pode não ter bom desempenho para processos com grandes valores de constante de tempo ou tempo morto</li> <li>- o controlador pode introduzir instabilidade no sistema</li> </ul>	<ul style="list-style-type: none"> <li>- necessidade de se identificar e monitorar os distúrbios do processo</li> <li>- requer modelo do processo (comportamento da variável controlada frente a perturbações na variável manipulada)</li> </ul>

Os controladores *feedback* mais empregados são:

- proporcional (P): a saída do controlador é proporcional ao erro, conforme a Equação 2.1.

$$c(t) = \bar{c} + K_c e(t)$$

Equação 2.1



- proporcional integral (PI): acrescenta-se um termo integral do erro (Equação 2.2).

$$c(t) = \bar{c} + K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(t) dt \quad \text{Equação 2.2}$$

- proporcional integral derivativo (PID): existe um termo derivativo do erro (Equação 2.3).

$$c(t) = \bar{c} + K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(t) dt + K_c \tau_D \frac{de(t)}{dt} \quad \text{Equação 2.3}$$

Onde:  $c(t)$  é a ação de controle,  $\bar{c}$  é o estado estacionário da variável manipulada e  $K_c$  é o ganho do controlador,  $\tau_I$  é a constante de tempo integral e  $\tau_D$  é a constante de tempo derivativa.

Os controladores *feed forward* se baseiam na predição do comportamento futuro da planta a ser controlada, e esta predição é feita através do modelo do processo, portanto, este modelo é um elemento importante para a eficácia do controlador. Tais modelos podem ser estáticos ou dinâmicos, sendo que os primeiros não levam em consideração a dinâmica do processo, levando a prejudicar o desempenho do controlador em condições transientes. Uma configuração comum para controladores *feed forward* é o seu acoplamento com controladores *feedback*, com uma ação de controle conjunta.

## 2.2. Redes Neurais Artificiais (RNA)

### 2.2.1. Definição

O cérebro humano é constituído de células nervosas chamadas neurônios. Os neurônios comunicam-se através de processos químicos desencadeados nas sinapses, sendo os sinais recebidos pelos dendritos, processados no núcleo e retransmitidos pelo axônio. Durante este processo, o sinal pode ser modificado pelo transmissor químico.

As redes neurais artificiais (RNA) são sistemas de processamento de informações que tentam imitar essa estrutura do sistema nervoso biológico. Da mesma forma, a RNA é constituída por unidades (os neurônios) interconectadas de modo que cada conexão possui um peso associado (onde o conhecimento é armazenado) que multiplicará o sinal transmitido pelo neurônio. Cada unidade aplica uma função de ativação ao sinal de entrada para determinar a saída, conforme mostra a Equação 2.4. A forma como os neurônios estão organizados na RNA constitui sua arquitetura.

$$a = f(wp + b) \quad \text{Equação 2.4}$$

Onde:  $a$  : saída do neurônio;  $f(wp + b)$ : argumento da função de ativação;  $w$  : peso;  $p$  : entrada do neurônio;  $b$  : *bias*.

Uma RNA pode ser classificada como monocamada (possui apenas uma camada de entrada e uma de saída) ou multicamada (possui uma camada intermediária, além das camadas de entrada e de saída), sendo esta última capaz de resolver uma gama maior de problemas. (FAUSETT, 1994)

A RNA pode processar relações lineares e não-lineares, sendo que o interesse está justamente nas não-linearidades.

### 2.2.2. Arquiteturas da RNA

As redes neurais podem ser classificadas de acordo com as conexões entre os seus neurônios: numa rede *feedforward* não há realimentação na entrada da rede, ou seja, o sinal se propaga da entrada, camada a camada, até a saída; já uma rede recorrente envolve elementos de atraso unitário e o sinal da saída é realimentado à entrada.

### 2.2.3. Treinamento da RNA

O processo de aprendizagem da rede se dá a partir de conjuntos de dados de entrada e saída (dados representativos do processo), com o ajuste dos parâmetros de modo a minimizar uma função, que pode ser o erro entre a saída calculada pela rede e os respectivos valores desejados. Para o teste da rede, utilizam-se dados não utilizados na etapa de treinamento (CAMPOS e SAITO, 2004).

Existem três tipos de treinamentos: supervisionado (é fornecido à rede um conjunto-exemplo de dados de entrada e saída), por estímulo (a rede recebe um indicador de desempenho em intervalos de tempo regulares) ou por auto-aprendizagem (utiliza-se um algoritmo de aprendizagem por competição). (CAMPOS e SAITO, 2004).

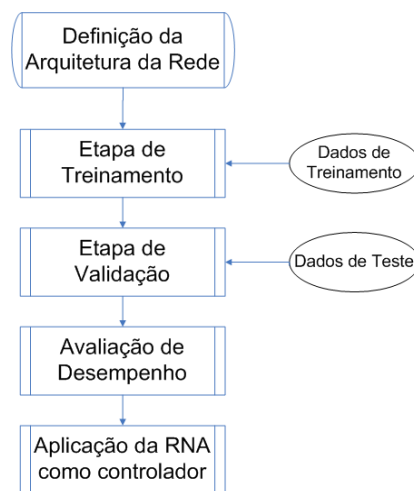
### 2.2.4. Algoritmo de retropropagação

O surgimento deste algoritmo aplicado a redes multicamadas representou um grande avanço para as redes neurais. Este algoritmo possui uma etapa inicial de avaliação, em que a rede calcula a saída a partir de uma entrada, gerando um erro ( $y - y_{ref}$ ) por comparação com a saída desejada e uma segunda etapa de retropropagação do erro. Pode-se dizer que o algoritmo da retropropagação modifica os pesos proporcionalmente à sua influência no erro cometido pela rede neural. (CAMPOS e SAITO, 2004)

Frequentemente utiliza-se a diferença entre a saída da rede e a saída real como função a ser minimizada durante o ajuste dos pesos da rede, segundo o critério da soma dos quadrados dos erros.

Normalmente, o banco de dados disponíveis é separado em dois conjuntos: 75% para o treinamento e 25% para a validação.

Em suma, o ciclo que começa com o processo de aprendizagem da rede e termina na sua aplicação como controlador pode ser visualizado na Figura 2.1:



Fonte: CAMPOS e SAITO, 2004

Figura 2.1: Processo de aprendizagem da rede neural artificial

### 2.2.5. Função de ativação

As funções de ativação utilizadas nos neurônios são responsáveis pela ativação (ou inibição) da saída da unidade processadora. O neurônio aplica um somatório aos sinais recebidos, ponderando-os pelos pesos e *bias*, utilizando este resultado como entrada da função de ativação.

Para problemas não lineares, a função de ativação sigmoide é frequentemente aplicada:

a) função logística:

$$f(x) = \frac{1}{1 + e^{-ax}} \quad \text{Equação 2.5}$$

Onde  $a$  é o parâmetro de inclinação da função.

b) função tangente hiperbólica (normalizada entre -1 e 1):

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad \text{Equação 2.6}$$

### 2.2.6. Aplicações

De uma forma geral, pode-se dizer que as redes neurais artificiais encontram aplicações nas seguintes classes de problemas: aproximação, predição, classificação, otimização e categorização.

Problemas de aproximação consistem de ajustes de modelos pela RNA, em que ela interpola valores de uma função dentro de um domínio de atuação. Para predição de estados futuros de determinadas variáveis num processo, também pode-se aplicar as redes neurais, sendo que esta previsão é baseada num histórico de dados do processo. O reconhecimento de padrões para reconhecer vozes e imagens constitui-se de um problema de classificação, em que atribuem-se padrões de entrada a uma classe de elementos (que podem ser pessoas, por exemplo) entre um conjunto de classes conhecidas. A RNA responde então, quando apresentada a um determinado elemento, a qual classe ele pertence. No caso de problemas de categorização, características estatisticamente relevantes de um conjunto de dados são reconhecidas pela rede neural, sendo que esta atua no sentido de separar em conjuntos os elementos com características semelhantes (REZENDE, 2003).

Mais especificamente, em Engenharia Química pode-se citar: detecção de falhas em sistemas de controle, identificação e localização de erros em processos, otimização de processos e controle de processos. Na Medicina: diagnósticos e tratamentos e em áreas comerciais: análise de riscos para investimentos. Podem ser citadas ainda, aplicações como: processamento de

sinais (filtragem de ruídos em linhas telefônicas) e reconhecimento de padrões (NAGY, 2006; FAUSETT, 1994).

### *2.2.7. Vantagens e limitações*

A RNA é um modelo que possui a vantagem da facilidade de aplicação e eficiência computacional. Trata-se de um modelo interessante para modelagem de processos devido à sua capacidade de capturar dinâmicas não-lineares por meio de simples equações algébricas.

Como todo método estatístico, a RNA requer uma base de dados representativos do processo, de forma a abranger todo o domínio de atuação da rede (CAMPOS e SAITO, 2004).

Sabe-se da literatura que as redes neurais são mais indicadas para interpolação de dados.

## **2.3. Bromelina**

O Brasil já é considerado o maior produtor mundial de abacaxi, segundo o Informativo Mensal da Equipe Técnica de Abacaxi do primeiro quadrimestre de 2007 (Revista Embrapa Mandioca e Fruticultura Tropical – CUNHA (2007)). As principais espécies cultivadas no país são a variedade Pérola e Smooth Caienne. As partes não comercializadas do fruto, como folhas, caule, talo e casca são ricas em bromelina, uma enzima proteolítica de importante uso farmacêutico.

### *2.3.1. Definição*

A bromelina é uma glicoproteína que possui um resíduo oligossacarídeo por molécula ligado de forma covalente à cadeia peptídica. Encontra-se bromelina até mesmo no estágio maduro do fruto do abacaxi, o que não ocorre com outras proteases vegetais, como a papaína (mamão) e a ficina (figo). Esta, portanto, é uma vantagem de se utilizar a bromelina em pesquisas científicas (MURACHI, 1976).

### *2.3.2. Aplicações*

Podem-se citar diversos usos da bromelina: na indústria alimentícia (amaciamento de carnes vermelhas, produção de farinha de trigo de elevado teor proteico, produção de ovos desidratados, hidrólise de complexos proteína-taninos que causam turvação da cerveja e produção de isolados proteicos), na indústria têxtil (amaciamento de couro, tratamento de lã e de seda) e na medicina e principalmente na indústria farmacêutica (suturação de feridas, antiinflamatório, absorção de outros medicamentos, facilitador da digestão e inibição de células cancerígenas) (MATTOS, 2005).

### *2.3.3. O Processo de Precipitação*

Para a recuperação de enzimas, utilizam-se técnicas como: precipitação, extração com solvente e filtração, técnicas estas que possuem baixo poder de purificação. Para obter uma pureza maior, é necessário investir-se em tecnologias mais avançadas, como a cromatografia de afinidade, eletroforese, extração em duas fases aquosas e extração com micela reversa. Esta etapa de purificação representa a maior parcela do custo de produção – de 80 a 90% (CESAR, 2005). Constata-se que este é um fator preponderante no que diz respeito à não existência de indústrias dedicadas à recuperação da bromelina no Brasil.

O processo de precipitação se dá através de interações eletrostáticas entre regiões de cargas opostas da própria molécula de proteína, quando o solvente destrói a camada de hidratação hidrofóbica desta. O método da precipitação por etanol possui como vantagens o baixo custo e a disponibilidade deste solvente, além da possibilidade de reaproveitamento do etanol através de um sistema de destilação. Como desvantagens, tem-se a inflamabilidade do solvente e a necessidade de se realizar a precipitação a baixas temperaturas, pois este solvente pode causar desconformação irreversível na estrutura tridimensional da proteína. Trabalhando-se em baixas temperaturas, reduz-se a flexibilidade da molécula e a capacidade de penetração do solvente (SCOPES, 1994).

## 2.4. Estado da arte

### 2.4.1. Redes Neurais Artificiais

Martins e Coelho (2000) estudaram a integração de redes neurais artificiais *feedforward* com controladores baseados em teorias convencionais para aprimorar o desempenho destes últimos. Os valores da variável controlada preditos pela rede foram incorporados à estrutura do sistema de controle convencional através da mudança nos valores de *set-point*. Simulações com controle PI, controle PID e uma estrutura de controle-cascata baseada em um controlador PID mostraram que a RNA de fato contribuiu para melhoria da performance dos controladores convencionais (diminuindo o tempo para atingir o estado estacionário, *overshoot* e oscilações na variável controlada) e corroboraram a eficiência do modelo neural para a predição dos valores da variável controlada. O algoritmo não-linear simplex de Nelder e Mead foi utilizado para o treinamento da rede neural, através de programação computacional.

Ainda no campo da simulação, observou-se que redes neurais têm sido aplicadas para o controle de biorreatores de fermentação. Uma das formas de se reduzir a variabilidade de um bioprocessos, aumentando-se assim a produtividade e portanto a lucratividade, é investindo na supervisão dos bioprocessos com o uso de técnicas de inteligência artificial, como as redes neurais artificiais e outros sistemas baseados em conhecimento empírico (GLASSEY et al., 1997).

Gadkar, Mehra e Gomes (2005) propuseram a aplicação de redes neurais para o monitoramento das variáveis de estado em um tanque de fermentação alcoólica. Através de um algoritmo de atualização dos pesos, o modelo neural foi implementado *online*. Resultados comparativos mostram que o erro gerado pelo modelo com adaptação dos pesos (erro de 12,5% para predição do perfil da biomassa e 22,2% para predição do perfil do etanol) foi menor do que aquele obtido pelo modelo sem adaptação dos pesos (erro de 25% para predição do perfil da biomassa e 44,4% para predição do perfil do etanol). Nagy (2006) também estudou o controle de um biorreator de fermentação utilizando redes neurais e demonstrou a eficiência dos modelos gerados e a superioridade de desempenho em relação a controladores convencionais.



Observa-se que o número de aplicações *online* tem crescido, porém com poucos casos bem-sucedidos, conforme descrito por Hussain (1999). Em sua revisão sobre aplicações de redes neurais para controle de processos químicos, 24 publicações foram estudadas acerca de aplicações *online* desta técnica. Algumas delas, como a de Khalid et al. (1993), a de Van Can et al. (1995), a de Evans et al. (1993 *apud* HUSSAIN, 1999), a de Seborg (1994) e a de Draeger et al. (1995), fizeram comparações com controladores convencionais e/ou esquemas de controle preditivo com modelo linear, chegando à conclusão de que o uso de redes neurais apresentou melhor resultado para controle de temperaturas, pressão e nível de tanques. Hussain também apontou que as redes neurais *feed forward* multicamadas com função de ativação sigmoïdal e tangente hiperbólica são as mais utilizadas, o que atesta a suficiência destas estruturas para os problemas abordados. Com relação a aplicações *online*, o autor destaca que poucos são os estudos em plantas-piloto, sendo a maioria deles voltados para plantas em escala de laboratório. Isto fatalmente estimula os pesquisadores a investigarem o uso *online* da tecnologia de redes neurais artificiais para controle de processos.

Um estudo de dois anos sobre aplicações industriais de redes neurais artificiais foi publicado em 2001 por Lennox et al. Um dos processos estudados foi a extrusão de um polímero (para cobertura de fios de cobre), buscando a manutenção da qualidade do produto, medida em relação ao *flow number* do mesmo. A implementação *online* foi feita acoplado um computador IBM com um processador 286 ao sistema de controle e aquisição de dados existente. A predição da rede neural manteve-se em torno de  $\pm 10$  unidades do *flow number* em relação ao *set-point* para a maioria dos dados testados, quando o objetivo era manter aproximadamente em torno de  $\pm 20$  unidades, comprovando mais uma vez a vantagem de se utilizar um controlador baseado num modelo neural ao invés de utilizar um modelo linear. Um outro processo estudado foi a determinação dos níveis ótimos de ozônio e de coagulante para filtração de águas do Rio Tâmis. O modelo neural foi construído para prever a turbidez das águas ao longo do tempo, saída esta utilizada numa função objetivo (a ser minimizada) para a

implementação *online*. Para fins de teste, o modelo neural foi implementado num computador com processador 386, conectado à rede de monitoramento das águas do Tâmis. Assim, era possível obter-se dosagens de ozônio e de coagulante sugeridas pelo sistema. Esta implementação encontra-se ainda em fase de estudo. Neste mesmo trabalho, a habilidade de adaptação dinâmica a mudanças no processo foi verificada na determinação da razão ar-combustível num motor a gasolina. O modelo gerado foi implementado num sistema de controle para regular esta proporção e os resultados mostraram que o modelo neural teve um desempenho superior quando comparado a um modelo linear, no que diz respeito ao tempo para se adaptar às novas condições operacionais. Isto é explicado pela capacidade das RNAs de identificar as não linearidades do processo. Como conclusões desta pesquisa de dois anos, os autores encontraram um aumento entre 10 a 20% na exatidão do modelo neural, em comparação ao modelo linear. No entanto, problemas de estabilidade de controladores baseados neste tipo de técnica ainda representam um significativo argumento para resistência à sua implementação na indústria. Este artigo também demonstrou que as redes neurais possuem grande potencial de aplicação em processos que requerem controle adaptativo.

Padmavathi et al. (2005) estudaram um processo de fabricação de borracha de polibutadieno. A uniformidade da qualidade deste produto pode ser garantida através da inferência do valor da viscosidade *Mooney* a partir de medidas das variáveis do processo, utilizando redes neurais artificiais. Para o treinamento da rede neural foram utilizados códigos gerados no MATLAB. Diversas estruturas de rede foram testadas (entre elas, uma estrutura com 1 camada intermediária utilizando a função de ativação tangente hiperbólica) e os autores concluíram que o melhor modelo foi aquele gerado pela rede neural com regressão generalizada (GRNN – *Generalized Regression Neural Networks*). Neste caso, o erro quadrado médio obtido nos testes foi de 0,0252. Para tornar o modelo portátil, os códigos do MATLAB foram transportados para uma planilha do Microsoft Excel, que compreendia os cálculos da rede neural para predição das viscosidades desejadas. Com o modelo definido, foi possível implementá-lo no sistema de controle distribuído disponível, sendo as ações corretivas tomadas pelo

operador. Apesar de não ter sido finalizada a automação deste processo, considerou-se no trabalho citado o uso posterior do modelo desenvolvido como um *soft-sensor* para controle da planta.

Em 2006, Fileti, Pacianotto e Cunha comprovaram a eficiência das redes neurais artificiais na implementação *online* de um modelo neural inverso no ajuste operacional do período final de sopro de um processo de aciaria a oxigênio. Trata-se de um complexo processo transiente sujeito a variações na matéria-prima, sendo de suma importância, portanto, a disponibilidade de um modelo preciso para definir as condições adequadas de operação. Desta maneira, evita-se retrabalho do produto obtido, pois suas especificações são prontamente atendidas ao final do processo. Os autores relataram um aumento no acerto de predição de temperatura e porcentagem de carbono de 66% (modelo comercial) para 82,5% (modelo neural) em 40 bateladas.

Segundo Hernández et al. (2009), uma outra aplicação pertinente das redes neurais artificiais é na determinação *online* do coeficiente de performance (COP) de um sistema integrado de purificação de água e de um ciclo de absorção (*absorption heat transformer*). Neste processo, a predição do COP por modelos termodinâmicos é imprecisa devido a hipóteses adicionadas que na prática não se concretizam. Assim, o uso de redes neurais surge como uma alternativa, já que esta técnica permite que não linearidades sejam incorporadas ao modelo gerado. Utilizando o *software* MATLAB, os autores concluíram que o modelo neural estava apto a ser aplicado *online* no sistema em estudo. Utilizando uma segunda rede neural (desta vez inversa), o objetivo era determinar, através de uma estratégia de otimização, qual a pressão ótima no absorvedor para um determinado COP. Nesta segunda etapa, foi utilizado o *toolbox* de otimização do MATLAB, aplicando o método simplex de Nelder-Mead. Os valores de pressão obtidos através da otimização e experimentalmente foram comparados e um erro de 1,04% foi obtido, o que é aceitável segundo o artigo.

Gonzaga et al. (2009) utilizaram redes neurais *feed forward* no desenvolvimento de um *soft-sensor* para estimar em tempo real a viscosidade do polímero politereftalato de etila (PET), para posterior aplicação num sistema de

controle. *Software-sensors* são dispositivos capazes de realizar medidas de diversas variáveis ao mesmo tempo e de processá-las através de algoritmos, provendo medidas de variáveis imensuráveis, em modo *online*. Os autores utilizaram as RNAs para previsão da viscosidade do PET, saída esta que era então enviada ao sistema de controle, que mantinha as condições operacionais adequadas para cada valor de viscosidade. A integração entre o *soft-sensor* desenvolvido e o sistema supervisor utilizado (SETCIM, AspenTech) foi feita utilizando uma Interface de Programação de Aplicativos (*Application Programming Interface*, API), através de programação computacional em linguagem FORTRAN e C. Este trabalho foi importante para mostrar uma alternativa aos viscosímetros usualmente utilizados nas indústrias, que apresentam problemas operacionais. O uso do modelo neural mostrou-se eficiente na predição da viscosidade do polímero em questão, tanto nos testes *offline* quanto nos testes *online*. Na aplicação do modelo para controle do processo, ambos os casos servo e regulatório apresentaram resultados bem sucedidos, tendo sido obtidas respostas rápidas e com erro praticamente nulo com relação ao *set-point*.

#### 2.4.2. Bromelina

Cesar (2005) realizou um estudo econômico sobre o processo de recuperação da enzima bromelina. Utilizando o processo de precipitação com etanol, foram testadas diferentes proporções de caldo de abacaxi e etanol, comprovando-se que com 80% v/v, praticamente toda a enzima é precipitada. Esta foi a condição apontada como adequada para precipitação *batch*, ou seja, em um único estágio. Outro dado importante apontado neste trabalho foi a temperatura de referência para a realização da precipitação: 5°C.

Estudos de diferentes processos de separação desta enzima foram estudados por vários autores. Bertevello (2001) investigou a eficiência de separação e purificação de bromelina num sistema de extração de duas fases aquosas, tendo obtido uma recuperação de atividade enzimática de 58%. Lopes et al. (2005) utilizaram com sucesso um sistema de membranas planas, que constituem barreiras seletivas que promovem a separação dos componentes da

mistura, para recuperar a enzima bromelina, apontando este método como uma alternativa aos processos de separação existentes.

Voltando a pesquisa da literatura para a aplicação de controle de processos à recuperação da bromelina, observam-se dois trabalhos relevantes: Santos (2006) e Leite (2007).

Santos (2006) estudou a aplicação de controladores convencionais PI, PID e PID adaptativo ao processo de precipitação de bromelina num reator de vidro e obteve melhor desempenho do controlador adaptativo em relação ao convencional (menor valor de ITAE, menor valor de *overshoot* e menor valor de tempo de estabilização), demonstrando assim a necessidade de um controle adaptativo ao processo em questão quando este é realizado em batelada alimentada. No entanto, algumas limitações foram observadas no sistema utilizado: havia uma ineficiência de troca térmica devido ao fato de o tanque ser feito de vidro, o que acarretou na impossibilidade de se manter a variável controlada em torno do *set-point*.

Leite (2007) deu prosseguimento ao trabalho de Santos e utilizou um controlador avançado baseado em lógica *fuzzy* aplicado ao mesmo processo, porém num reator de aço-inox e demonstrou a eficiência e a superioridade deste controlador avançado em relação a controladores convencionais, obtendo uma recuperação da atividade enzimática 80% maior em relação aos controladores convencionais e menor consumo de energia elétrica.

## 2.5. Otimização

Para esta seção, utilizou-se como referência o livro de otimização de Edgar, Himmelblau e Lasdon (2001).

Otimizar é fazer uso de determinados métodos para maximizar ou minimizar uma função-custo, de modo a encontrar uma solução adequada para um problema.

Técnicas de otimização vêm sendo estudadas com o intuito de trazer melhorias aos processos industriais. Isto foi motivado pelo aumento nos custos de

energia, no rigor das leis ambientais e na competitividade do mercado. As ferramentas de otimização são aplicadas visando ao aumento da eficiência dos processos e portanto da produtividade, gerando conseqüentemente mais lucratividade.

Aplicações desta técnica são encontradas no mundo dos negócios (obtenção de máximo lucro, utilizando o mínimo de recursos, por exemplo) e na engenharia (encontrar as quantidades mínimas de reagentes para obtenção da máxima produtividade).

Basicamente, um problema de otimização consiste de três elementos: a função-objetivo, restrições de igualdade (equações) e restrições de desigualdade (inequações).

Quando a função-objetivo e as restrições são lineares, pode-se resolver o problema de otimização através de uma programação linear simples, usualmente na forma apresentada no Quadro 2.1.

Quadro 2.1: Equacionamento para encontrar a solução de um programa de otimização utilizando programação linear

“Encontrar  $x = (x_1, x_2, \dots, x_n)$ , de modo a minimizar a função:

$$f = \sum_{j=1}^n c_j x_j$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m$$

$$l_j \leq x_j \leq u_j, \quad j = 1, 2, \dots, n$$

Onde:

$c_j$  são os coeficientes das  $n$  funções-objetivo

$a_{ij}$  e  $b_i$  são os parâmetros das  $m$  restrições lineares

$l_j$  e  $u_j$  são os limites inferior e superior de  $x_j$ ”

Para encontrar a solução ótima de um problema deste tipo, pode-se aplicar o método simplex, em que num primeiro momento define-se uma solução preliminar, se esta existir e num segundo momento utiliza-se essa primeira solução como valor inicial para encontrar a solução através da minimização (ou maximização) da função.

Outra situação que pode ser encontrada é a existência de uma função-objetivo não linear, com restrições lineares ou não lineares. Neste caso, um método indicado para solucionar este problema é o método do gradiente descendente generalizado (*generalized reduced gradient*, GRG), cujo algoritmo consiste dos passos descritos no Quadro 2.2.

Quadro 2.2: Algoritmo do método do gradiente descendente generalizado

1. Calcule o gradiente de  $f(x)$  no ponto atual  $x_c$ , gerando  $\nabla f(x_c)$ .
2. Se o ponto  $x_c$  está próximo do ponto ótimo, pare.
3. Calcule a direção do gradiente,  $d_c$ , usando o gradiente  $\nabla f(x_c)$  e alguma informação anterior, como a direção do gradiente anterior.
4. Determine qual a distância a ser percorrida ( $\alpha_c$ ) a partir de  $x_c$ , ao longo da direção  $d_c$ .  $\alpha_c$  é o valor de  $\alpha$  que minimiza a função  $f(x_c + \alpha d_c)$  e que é utilizado para definir o próximo ponto  $x_n = (x_c + \alpha_c d_c)$ .
5. Substitua  $x_c$  pelo valor encontrado,  $x_n$ , e retorne ao primeiro passo.

Edgar, Himmelblau e Lasdon (2001) apresentam como vantagens deste método a sua versatilidade (pode ser utilizado em problemas com restrições lineares e não lineares ou sem restrições) e o fato de que, uma vez que a solução seja possível, ela permanecerá possível e o método pode ser interrompido a qualquer momento com uma solução. No entanto, este método requer que a cada passo do algoritmo todas as restrições sejam atendidas.

O método do GRG descrito anteriormente é utilizado na ferramenta *Solver* do Microsoft Excel. Este suplemento possui uma interface de fácil interação com o usuário. Maiores detalhamentos sobre o uso desta ferramenta podem ser encontrados em Fylstra et al. (1998).

## 2.6. Comunicação Digital *Fieldbus*

A referência utilizada para este tópico foi o manual 'Como implementar projetos com *Foundation Fieldbus*' (SMAR, 1998).

A necessidade de minimização de custos, melhorias em procedimentos de manutenção e maior integração entre as diversas áreas levou à introdução do conceito de redes.

As redes são classificadas de acordo com o tipo de equipamento conectado a elas e com o tipo de dado que trafega por elas:

- rede *sensorbus*: dados em formato de *bits* (sinais discretos do tipo *ON/OFF*);
- rede *devicebus*: dados em formato de *bytes* (pacotes de sinais discretos e/ou analógicos) e
- rede *fieldbus*: dados em formato de bloco (pacotes de informações de tamanhos diversos).

A rede *Fieldbus* é um sistema de comunicação digital bidirecional, capaz de cobrir distâncias maiores que as redes *Sensorbus* e *Devicebus*, que interliga instrumentos de campo a estações de controle. Instrumentos acoplados a redes *fieldbus* são chamados de instrumentos inteligentes, pois são capazes de desempenhar funções complexas e de realizar cálculos avançados, sendo, portanto, adequados para utilização em controle de processos. O padrão *Ethernet* operando com o protocolo TCP/IP é o mais comumente utilizado no nível de informação da rede.

Observa-se a vantagem de se utilizar a rede *Fieldbus* frente à tecnologia de Sistema Digital de Controle Distribuído (SDCD) no detalhamento do projeto,



pois o grau de complexidade de alguns elementos do projeto (diagrama de malhas e planta de instrumentação, por exemplo) é menor no caso do projeto *Fieldbus*.

No projeto *Fieldbus*, é de suma importância que se faça um planejamento cuidadoso da localização de cada equipamento da rede, de modo a se tirar o máximo proveito do desempenho do barramento e das derivações, utilizando-se comprimentos adequados destes elementos. Além deste cuidado, também deve-se atentar para o número máximo de dispositivos ligados à rede, o que é definido pela fonte de alimentação, que deve ser suficiente para alimentar todos os transmissores.

Nos projetos *Fieldbus*, existem diversas possibilidades de topologias:

- barramento com *spurs*: existe um barramento principal, do qual partem derivações, onde os instrumentos serão conectados;
- ponto-a-ponto: todos os equipamentos da aplicação são conectados em série;
- árvore: acopladores concentram a ligação de vários equipamentos, formando uma estrutura semelhante a um “pé de galinha”, outro nome dado a esta topologia;
- *end-to-end*: apenas dois equipamentos são conectados diretamente;
- mista: diferentes tipos de topologia são utilizados numa mesma aplicação.

Os componentes utilizados num projeto *Fieldbus* são listados a seguir:

- cabos: par de fios trançados com blindagem seguindo a norma ISA-S50.02;
- conectores: dispositivos opcionais, indicados para uso em redes que possuem equipamentos que devem ser frequentemente desconectados;
- blocos de terminais: possibilitam conexões de vários barramentos;

- *Host Device*: equipamento que permite a configuração, monitoramento e interação com o processo sem a necessidade de se permanecer no campo (computador);
- repetidor: utilizado para estender um segmento *Fieldbus*;
- *bridge*: utilizado para conectar segmentos *Fieldbus* de diferentes velocidades, formando uma extensa rede;
- *gateway*: utilizado para conectar os segmentos *Fieldbus* a outros protocolos de comunicação (*Ethernet*, por exemplo);
- instrumentos (transmissores, conversores).

Por fim, observa-se que o uso desta tecnologia traz alguns benefícios como o aumento da disponibilidade de informações de controle, assim como de informações extra controle, além de vantagens econômicas (baixo custo de implantação e de expansão).

## 2.7. Conclusões do capítulo

Como pode ser observado, Redes Neurais Artificiais têm sido vastamente empregadas para modelagem de processos devido à sua capacidade de captar dinâmicas não lineares e à sua fácil implementação (já que se elimina a necessidade de se modelar os processos a partir das equações básicas de conservação). Esta ferramenta tem se provado eficiente não só em modelagem, como também na predição de variáveis de processos para posterior implementação de alguma estratégia de controle.

Dentre os estudos envolvendo a aplicação de redes neurais, nota-se que a grande maioria utiliza códigos de programação computacional ou *softwares* comerciais de custo elevado para a implementação *online*, quando explicitado. Assim, a proposta de emprego do *software* Microsoft Excel para este fim visa simplificar o uso da técnica, uma vez que esta planilha possui uma interface amigável e está amplamente disponível na maioria dos computadores de uso pessoal e comercial.

O caso-estudo da enzima bromelina pode se mostrar vantajoso com relação ao uso de resíduos que normalmente não seriam aproveitados, obtendo-se um produto de importante uso farmacêutico de alto valor agregado a partir de matérias-primas de baixo custo. Já o uso do sistema digital *Fieldbus* para este estudo implica em atualização tecnológica.



## CAPÍTULO 3: MATERIAIS E MÉTODOS

O presente capítulo traz a descrição dos materiais utilizados no desenvolvimento do projeto, assim como o detalhamento dos procedimentos realizados em cada etapa. Inicialmente aborda-se o desenvolvimento do modelo neural, seguindo-se com o detalhamento do caso-estudo. Ao final, explicita-se o procedimento de sintonia do controlador convencional PID.

### 3.1. Desenvolvimento do Modelo Neural

A primeira parte do projeto foi o desenvolvimento do modelo neural. Esta fase consistiu de várias etapas, a serem detalhadas nos itens subsequentes, conforme esquematizado na Figura 3.1.

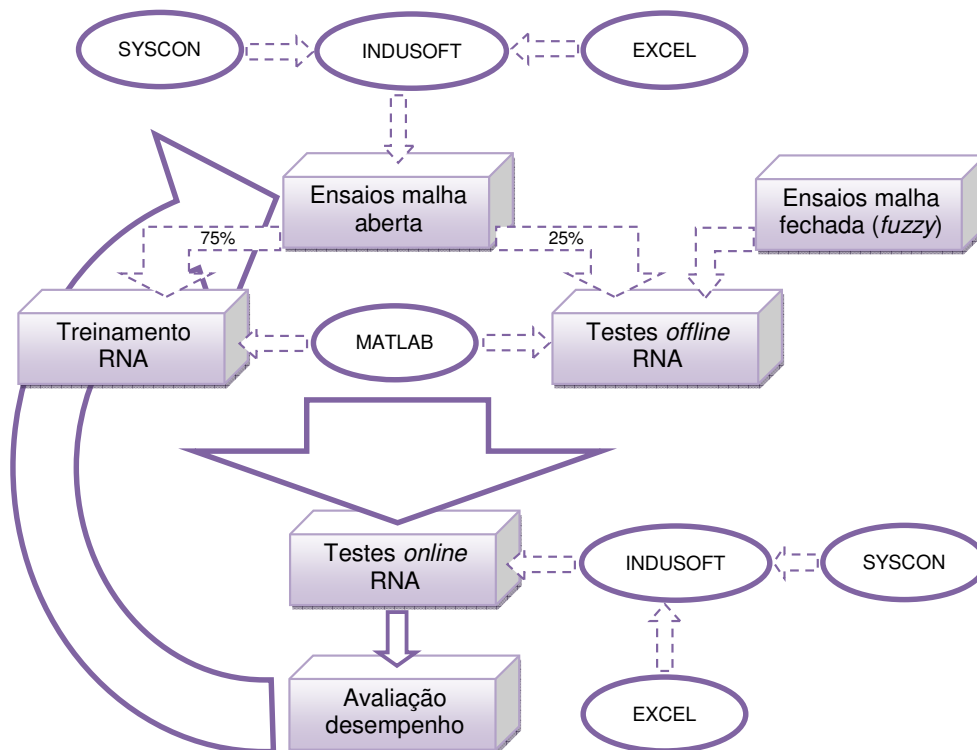


Figura 3.1: Etapas envolvidas no desenvolvimento do modelo neural e os softwares utilizados

### 3.1.1. Configuração de softwares

Como pode ser observado, diversos *softwares* estiveram envolvidos, tendo sido necessária a configuração adequada para a comunicação entre eles. A Figura 3.2 resume a relação entre os elementos do processo:

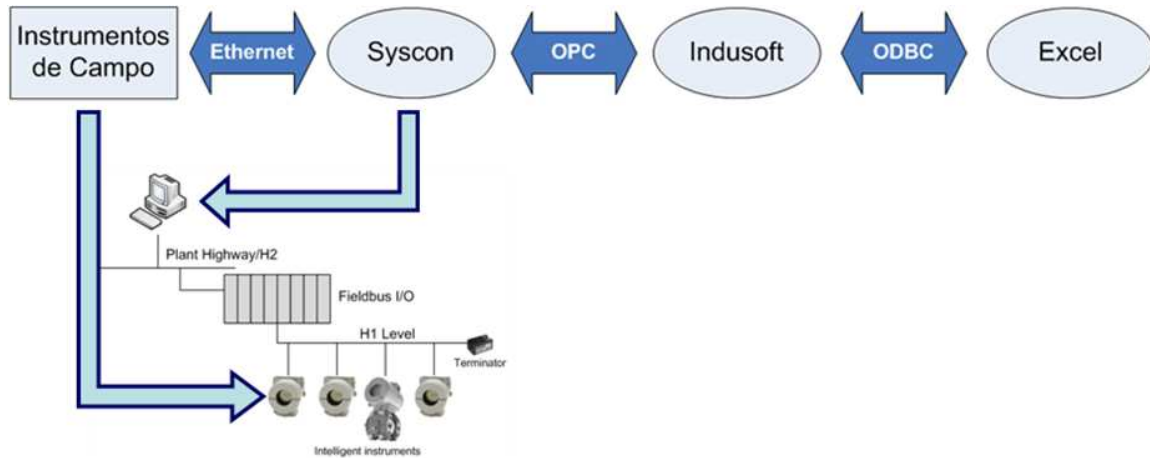


Figura 3.2: Comunicação entre os elementos do processo

A configuração da rede *Fieldbus* foi feita através do *software* Syscon 7.0 (*Smar*), no qual cria-se um projeto para especificação dos instrumentos (*Fieldbus Foundation*) utilizados e para determinação dos parâmetros dos seus respectivos blocos funcionais. Neste *software* são definidas as faixas de operação das variáveis controlada e manipulada e até mesmo a estratégia de controle do controlador PID.

As informações lidas pelos instrumentos de campo são enviadas à placa DFI (*Distributed Fieldbus Interface – DFI302, Smar*), gerenciadora de informações, utilizando o padrão *Ethernet* de comunicação operando com o protocolo TCP/IP.

O sistema supervisor é a interface usuário/sistema e é através dele que o operador envia as ordens aos instrumentos de campo e monitora as variáveis da planta. Faz-se necessário, portanto, a criação de uma tela de comandos para a interação do usuário com o sistema. O *software* empregado foi o Indusoft Web Studio 6.1 (*IWS do Brasil*). Esta tela de comandos foi criada a partir de uma

biblioteca de gráficos de equipamentos, tubulações e instrumentos, disponível no próprio programa. O Indusoft e o Syscon comunicam-se via porta OPC (*Ole for Process Control*).

A planilha eletrônica do Microsoft Excel 2003 (*Microsoft*) foi utilizada para implementação do modelo neural e desta forma, para possibilitar a escrita dos dados do processo (do Indusoft para o Excel) e a leitura da saída calculada pela rede (do Excel para o Indusoft), configurou-se no Indusoft um *driver* ODBC (*Open Data Base Connectivity*), uma interface de acesso a sistemas gerenciadores de bancos de dados.

O primeiro passo para se configurar este *driver* foi a criação de uma nova fonte de dados no Windows, cujos passos estão descritos no Apêndice A. Para os ensaios em malha aberta, a planilha ODBC foi configurada de modo a realizar a leitura dos valores de rotação da bomba de propilenoglicol do Excel, a cada 4 segundos. Esta amostragem foi determinada de modo a garantir tempo suficiente para que a bomba pudesse responder a cada novo valor de rotação implementado. Para os ensaios em malha fechada, foram configuradas duas planilhas ODBC: uma para a escrita dos vetores de entrada da rede neural e outra para a leitura do valor calculado pela rede, utilizando sempre a amostragem de 4 segundos. Também no Apêndice A, são mostrados os passos para configuração destas planilhas no Indusoft.

### 3.1.2. Ensaios em Malha Aberta

Para o treinamento da rede neural, é necessária a disponibilidade de um banco de dados sobre o processo em estudo.

No Laboratório de Controle e Automação de Processos da FEQ/UNICAMP, estudos haviam sido realizados acerca do processo de precipitação da bromelina, utilizando controlador PID e *fuzzy*, dados estes que poderiam ser utilizados no treinamento da rede neural. No entanto, isto poderia causar a impressão da necessidade de implementação prévia de um outro controlador para permitir a aplicação do controlador neural, o que

consequentemente poderia levar a se questionar a utilidade e a necessidade do estudo deste último.

Na verdade a implementação do controlador neural não é viabilizada somente com a implementação prévia de outro controlador, sendo possível ensinar a dinâmica do processo à rede neural através de ensaios em malha aberta, que mostrariam a ela qual o comportamento da variável controlada frente a determinadas perturbações em degrau na variável manipulada.

Desta forma, ensaios de precipitação em malha aberta foram realizados com o objetivo de se construir um banco de dados sobre o processo para posterior utilização destas informações no treinamento e nos testes da rede neural. A forma como os mesmos foram planejados está descrita no item 3.3.3. Procedimento para os ensaios em malha aberta.

### 3.1.3. Treinamento e Testes da RNA

Os dados experimentais do processo em malha aberta foram utilizados para o treinamento (75% da base de dados) e para os testes (25% da base de dados) da rede neural artificial.

Considerou-se uma estrutura de rede como a apresentada na Figura 3.3, com uma camada de entrada, uma intermediária (configuração recomendada para funções contínuas e bem comportadas, conforme descrito por Cybenko (1989) e uma de saída.

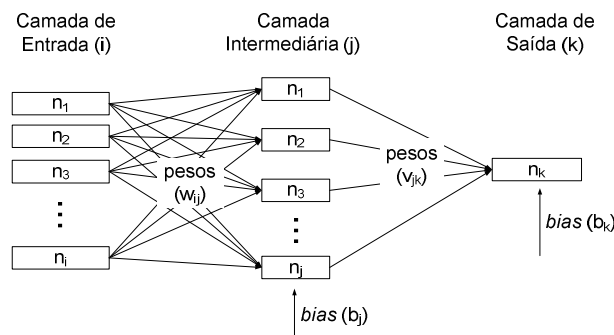


Figura 3.3: Estrutura de rede estudada



O modelo neural foi gerado a partir do *toolbox* de redes neurais artificiais do MATLAB v. 7.0.1 (*The Math Works Inc.*) e a arquitetura proposta consiste de uma rede *feed forward* com a camada de entrada composta pelas variáveis escolhidas em cada treinamento; o número de neurônios na camada intermediária (com a função tangente hiperbólica) sendo o dobro do número de variáveis da camada de entrada e um neurônio na camada de saída com a função linear.

Frequentemente trabalha-se com o algoritmo de aprendizagem por retropropagação de Levenberg-Marquardt, que foi o escolhido para o treinamento das redes neurais estudadas neste trabalho.

Os testes *offline* foram realizados também no *software* MATLAB, com dados não vistos no treinamento. A programação contempla a mesma normalização de dados utilizada no treinamento e faz a simulação com os novos dados, gerando como resultados gráficos comparativos das saídas real e calculada.

Os *scripts* de treinamento e de teste utilizados neste projeto encontram-se no Apêndice B, com os comentários marcados pelo símbolo de porcentagem.

Além dos ensaios em malha aberta, para o teste também foram utilizados alguns ensaios em malha fechada utilizando controlador *fuzzy*, dados estes provenientes de outros projetos desenvolvidos no Laboratório de Controle e Automação de Processos, na mesma planta. Estes ensaios foram incluídos para verificar a capacidade de predição da rede neural também em malha fechada, uma vez que esta seria a situação encontrada na realidade, quando da implementação do otimizador para controle da temperatura do meio precipitante. Os dados destes ensaios foram cedidos por Leite (2007) e por Ferrarezi (2008).

#### 3.1.4. Validação online do modelo neural

Após a realização dos testes *offline* no MATLAB, seguiu-se com uma etapa de validação *online* do modelo neural gerado.

Tendo em vista uma aplicação industrial, propõe-se o uso do *software* Microsoft Excel para implementação experimental do modelo neural gerado, pela sua popularidade e facilidade de utilização.

Para isto, construiu-se uma planilha eletrônica, de forma a reproduzir as operações algébricas envolvidas no cálculo da rede neural: os parâmetros definidos no treinamento da rede no MATLAB (pesos e *bias*) são inseridos na planilha, que executa automaticamente o cálculo da temperatura do meio precipitante um instante de amostragem à frente, em função dos vetores de entrada (Figura 3.4).

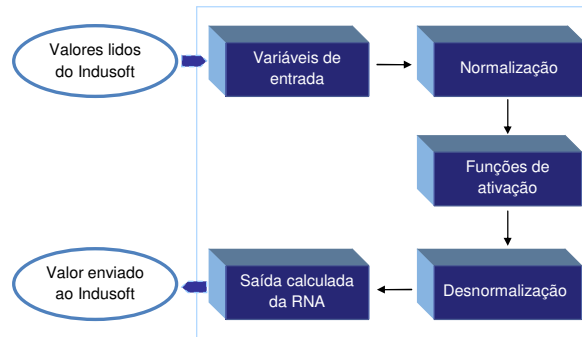


Figura 3.4: Etapas para predição da temperatura do meio precipitante utilizando o modelo neural na planilha eletrônica

A planilha foi estruturada conforme mostra a Figura 3.5:

MODELO NEURAL PARA O PROCESSO DE PRECIPITAÇÃO DE BROMELINA												
Modelo Direto - Topologia 7 x 1 x 1												
Máximos e Mínimos												
	T(k)	Tout(k)	Tin(k)	MV(k)	AMV(k)	nivel(k)	T(k-1)					
max	22.3	10.16	8.09	4.9	100	90	749.8					
min	16.82	-2.16	-0.35	-0.39	0	-30	150					
Camada de Entrada												
	T(k)	T(k)	T(k)	T(k)	T(k)	T(k)	T(k)					
	20.88	5.5	0.00	0.55	0.00	0.00	2000.00					
Camada Intermediária												
	Pesos w <sub>jk</sub>						Bias b <sub>j</sub>		Saída			
	-0.120	0.081	2.543	-0.735	0.341	-0.108	-8.844	-8.023	-0.795			
	1.338	8.984	-13.180	-36.987	4.395	14.202	94.381	79.452	0.981			
	0.024	-0.010	-0.339	0.230	-0.080	0.150	0.181	-2.270	-0.212			
	-0.029	-0.039	0.416	-0.282	0.111	-0.194	-0.222	0.303	0.241			
	0.682	1.028	-12.038	-8.000	3.155	-2.088	36.305	40.904	0.636			
	0.878	1.625	-9.378	10.966	-2.814	4.067	8.279	8.216	1.000			
	0.040	2.502	-13.852	5.775	1.703	10.550	9.656	5.133	0.051			
	-0.328	0.259	3.034	-2.871	0.877	-2.088	-2.049	-1.422	-0.731			
	-0.365	-0.274	3.707	-3.726	1.143	-2.092	-1.969	-2.114	-0.958			
	-0.381	-0.294	3.896	-3.907	1.152	-2.189	-2.032	-2.148	-0.961			
	-0.744	-1.092	-13.834	6.270	-3.598	2.590	-39.893	-45.636	-0.841			
	0.387	-0.362	-2.898	3.304	-1.203	2.904	1.704	-3.621	0.694			
	4.830	10.070	-39.203	-5.905	12.213	19.419	46.823	26.078	1.000			
	0.594	0.957	9.533	-5.646	2.504	-1.338	31.108	34.149	0.831			
Camada de Saída												
	Pesos w <sub>jk</sub>						Bias b <sub>j</sub>		Saída			
	0.091	0.132	-24.213	-19.752	11.400	-0.072	0.008	-0.103	-4.820	4.323	6.630	-0.175
	5.018	-4.981	-5.563	-0.172								
	Saída Calculada (Desnormalização)											
	T(k-1) = 5.44											

Figura 3.5: Modelo neural desenvolvido em planilha eletrônica

Esta planilha também foi utilizada nos ensaios em malha fechada utilizando o otimizador.

Os valores das variáveis de entrada, provenientes da planta, passam por uma normalização, de acordo com a Equação 3.1:

$$pn = 2 \frac{(p - \min p)}{(\max p - \min p)} - 1 \quad \text{Equação 3.1}$$

Onde:

$pn$  = valor normalizado da variável  $p$

$p$  = valor real da variável  $p$

$\min p$  = valor mínimo da variável  $p$

$\max p$  = valor máximo da variável  $p$

A função de ativação aplicada na camada intermediária foi a tangente hiperbólica (terminologia do MATLAB: *tansig*); sendo assim, o sinal proveniente da camada de entrada é recebido pela camada intermediária e ponderado pelos pesos e *bias*, sendo processado segundo a Equação 3.2:

$$a = \tan \operatorname{sig}(n_j) = \frac{2}{(1 + \exp(-2n_j))} - 1 \quad \text{Equação 3.2}$$

$$\text{Onde: } n_j = \left( \sum_i n_i w_{ij} \right) + b_j$$

Analogamente, a camada de saída processa o sinal dos neurônios da camada intermediária segundo a função linear (terminologia do MATLAB: *purelin*) (Equação 3.3):

$$purelin(n_k) = a$$

Equação 3.3

A saída calculada pela Equação 3.3 deve ser desnormalizada para implementação no processo e este cálculo é feito conforme a Equação 3.4, que nada mais é que a Equação 3.1 isolada em  $p$ .

$$p = 0,5 \cdot (pn + 1) \cdot (\max p - \min p) + \min p$$

Equação 3.4

Onde:

$pn$  = valor normalizado da variável  $p$

$p$  = valor real da variável  $p$

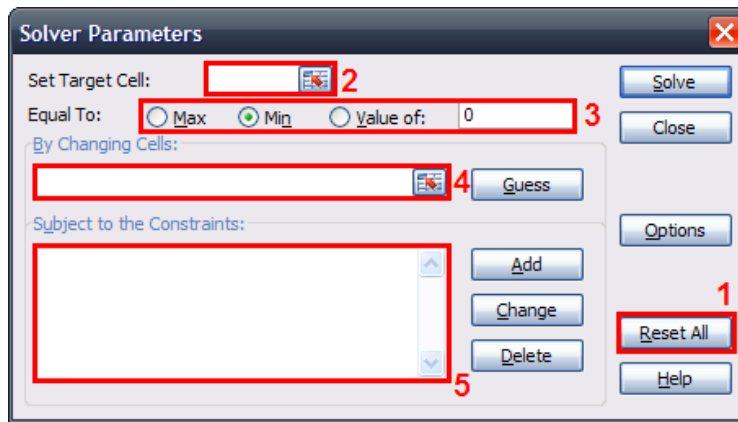
$\min p$  = valor mínimo da variável  $p$

$\max p$  = valor máximo da variável  $p$

Para a realização destes ensaios de validação, utilizou-se o controlador PID do *software* Syscon para variar a rotação da bomba de propilenoglicol, visando testar a capacidade de predição da rede em malha fechada. O modelo neural foi acionado em paralelo com o controlador PID com o único objetivo de se monitorar a temperatura calculada, para posterior comparação com a temperatura real, cuja aquisição de dados foi feita através do Indusoft.

### 3.2. Configuração do otimizador

O Microsoft Excel possui uma ferramenta de otimização chamada *Solver*, que funciona com uma interface conforme mostrado na Figura 3.6:



- 1: redefine todos os parâmetros para o valor *default*;  
 2: função-objetivo;  
 3: valor desejado para a função-objetivo;  
 4: valor a ser variado;  
 5: campo de restrições.

Figura 3.6: Interface para configuração do Microsoft Excel *Solver*

Entretanto, para o controle efetivo da planta, foi necessário desenvolver uma programação computacional em *Visual Basic for Applications* (VBA) para integrar o otimizador ao modelo neural, para que o cálculo de otimização fosse executado a cada instante de amostragem para cada novo vetor de entrada do modelo e não uma única vez, como ocorreria se a interface tivesse sido utilizada.

Fazendo uso de uma biblioteca de comandos existente no Microsoft Excel, foi gerada uma rotina que é executada a cada atualização da planilha de cálculo, isto é, a cada período de amostragem (4 s), em que um novo vetor de entrada é gerado (novas condições na planta). O valor de temperatura do tanque previsto pelo modelo gera um erro em relação ao *set-point* que, elevado ao quadrado, passa a ser a função-objetivo do otimizador *Solver*. A solução é encontrada minimizando esta função através do método *quasi-Newton* do gradiente descendente generalizado, variando a rotação da bomba de propilenoglicol, estando sujeita às restrições da própria limitação física da bomba (0 a 100% de rotação). Uma outra restrição foi adicionada para suavizar a ação da variável manipulada, limitando os degraus (positivos ou negativos) a um valor máximo. Também foi necessário adicionar uma restrição para que a bomba desligasse quando a temperatura do meio precipitante atingisse um determinado valor, para

que esta temperatura não diminuísse muito, fazendo com que seu valor ficasse fora da faixa de atuação da rede neural, isto é, da faixa de valores em que a rede foi treinada. O *script* desenvolvido é mostrado no Apêndice B.

Para um melhor entendimento, a Figura 3.7 contém um esquema do fluxo de informações no acoplamento do modelo neural com o otimizador:

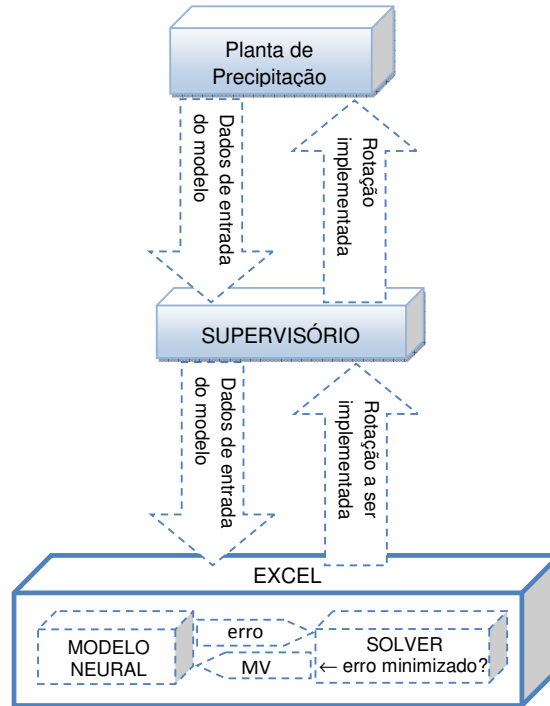


Figura 3.7: Fluxo de informações no acoplamento do modelo neural com o otimizador

### 3.3. Caso-estudo: precipitação de bromelina

#### 3.3.1. Protótipo da planta de precipitação de bromelina

A planta-piloto de precipitação de bromelina consiste de um tanque de aço inoxidável encamisado agitado (agitador *Fisatom 715*). A refrigeração é feita através da circulação de um fluido refrigerante – solução de propilenoglicol a 50% - pela camisa do tanque. Sensores de temperatura do tipo Pt-100 (termorresistências) estão alocados de forma a medir a temperatura interna do tanque, a do álcool etílico (*Chemco*, pureza 99,5%) e as temperaturas do propilenoglicol na entrada e na saída da camisa. Um transmissor de pressão

(Smar, Série LD300) é utilizado como medidor de nível do tanque e o elemento final de controle (atuador) é um inversor de frequência (fabricante Danfoss, modelo VLT 2800). Uma bomba de deslocamento positivo (fabricante RZR, modelo 500) é utilizada para promover a circulação do propilenoglicol, enquanto uma bomba peristáltica (Masterflex Pump Controller, do fabricante Cole Parmer Instrument Co.) é utilizada para promover a adição do álcool no tanque. Tanto o agitador quanto o sensor de temperatura do tanque foram instalados de modo que ficassem numa altura em que fosse possível o contato com o líquido desde 150 mL até o final da batelada, sendo que esta posição não interferiu nas medidas do sensor. Na Figura 3.8, pode-se visualizar o fluxograma de instrumentação da planta.

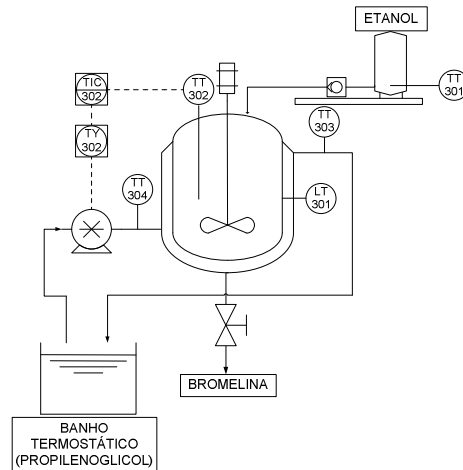
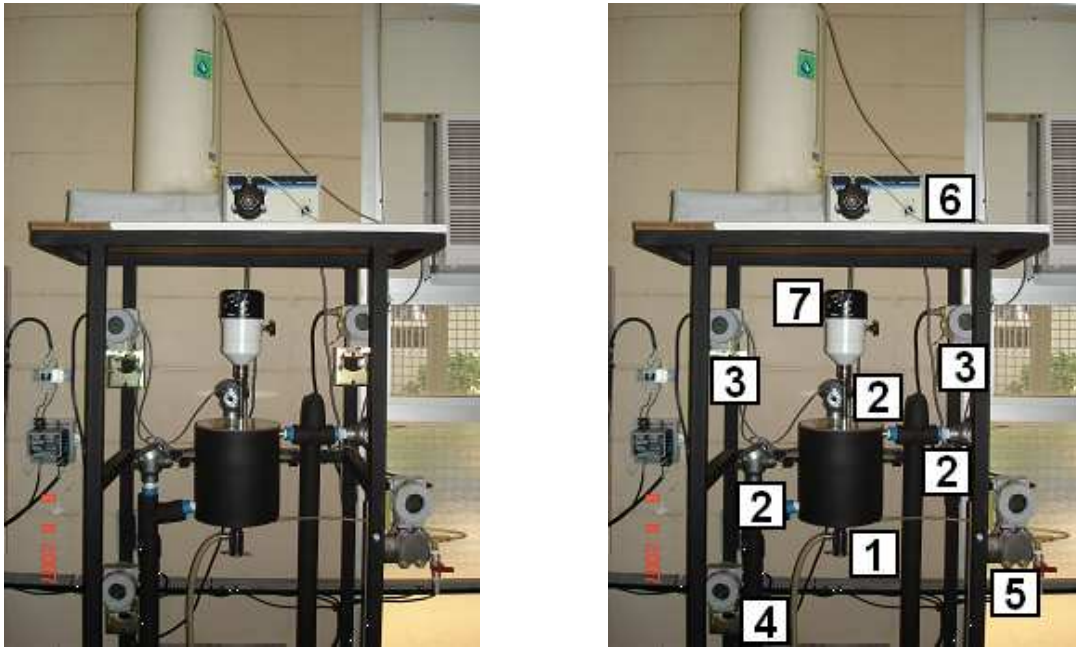


Figura 3.8: Fluxograma de instrumentação do protótipo de precipitação de bromelina

Instrumentos inteligentes estão conectados aos sensores de temperatura, de pressão e inversor de frequência de modo a haver a transmissão de dados para a placa DFI.

Apresenta-se na Figura 3.9 o sistema experimental utilizado.

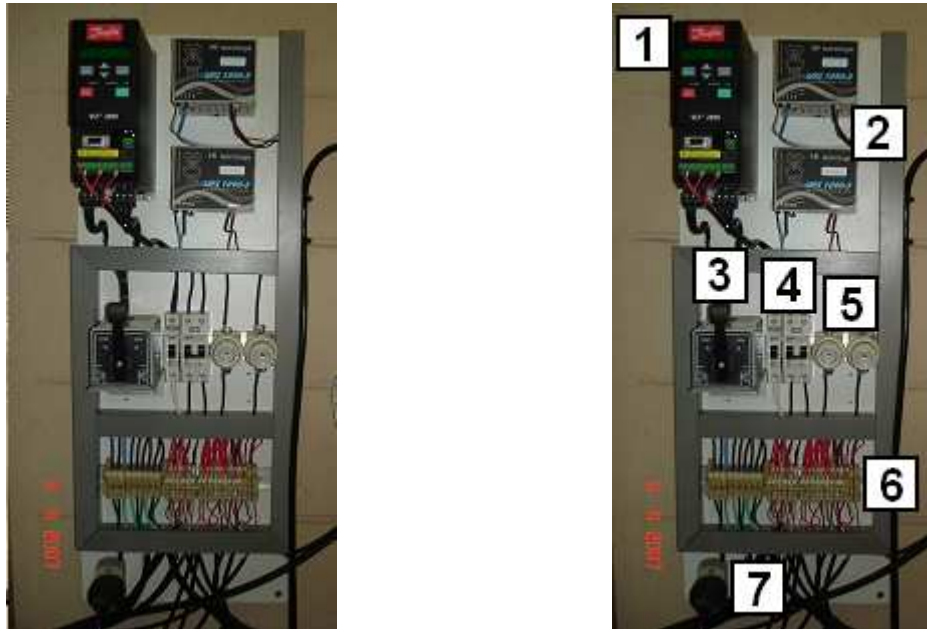


- 1: tanque de aço-inox encamisado e isolado com espuma armaflex;  
 2: sensores de temperatura do tipo Pt-100 (termorresistências);  
 3: transmissores de temperatura *Fieldbus* (TT302, *Smar*);  
 4: conversor *Fieldbus*-corrente (FI302, *Smar*), para variação da rotação da bomba de fluido de resfriamento;  
 5: transmissor de pressão *Fieldbus*, utilizado como medidor de nível (LD300, *Smar*);  
 6: bomba peristáltica para adição do álcool;  
 7: agitador.

Figura 3.9: Visão geral do sistema experimental

Foi montado também um painel (Figura 3.10) para organizar as ligações elétricas do sistema, garantindo a integridade de todos os elementos do circuito (fontes, fusíveis e disjuntores) e a acessibilidade para eventuais procedimentos de manutenção.





- 1: inversor de frequência;
- 2: fontes de alimentação dos instrumentos;
- 3: chave comutadora para alternância entre duas plantas;
- 4: disjuntores;
- 5: fusíveis;
- 6: régua de bornes;
- 7: tomada bifásica.

Figura 3.10: Painel elétrico utilizado no sistema experimental

### 3.3.2. Preparo das amostras

As amostras de suco de abacaxi para os ensaios de precipitação foram preparadas utilizando-se resíduos da fruta, como casca e talo (cilindro central fibroso). Estes resíduos foram triturados num liquidificador de uso doméstico com um volume conhecido de água destilada.

O caldo formado teve seu volume medido e foi então coado, utilizando-se um coador de tecido, com o objetivo de separar o caldo das fibras.

Como era desejável realizar-se os ensaios em condições uniformes (amostra de suco homogênea), preparou-se uma grande quantidade de amostras, que foram utilizadas para todos os ensaios. No entanto, a capacidade do

liquidificador doméstico é limitada ao volume aproximado de 1,5 L. Assim, as bateladas eram misturadas e homogeneizadas para garantir a representatividade da amostra.

Ao final, o volume total de suco obtido leva em consideração o volume de água adicionada para auxiliar no processo de trituração, conforme mostra a Equação 3.5:

$$V_{total} = V_{suco} + V_{água}$$

$$V_{suco} = V_{total} - V_{água} \quad \text{Equação 3.5}$$

O objetivo era ter uma amostra com a proporção de uma parte de suco para uma parte de água. Deste modo, deve-se contabilizar a quantidade de água necessária para se completar a proporção:

$$V_{água\,completar} = V_{suco} - V_{água} \quad \text{Equação 3.6}$$

As amostras preparadas eram armazenadas à temperatura de -18°C (congelador doméstico) e descongeladas à temperatura ambiente.

### 3.3.3. Procedimento para os ensaios em malha aberta

O planejamento dos ensaios em malha aberta foi feito visando à abrangência de uma ampla faixa de valores adequados da variável manipulada, de forma a cobrir todo o domínio de atuação da rede, inicialmente com base na observação do comportamento da variável controlada durante um ensaio em malha aberta, com a variável manipulada fixa em 40%.

Neste ensaio, observou-se que no início do processo, ocorre uma elevação de temperatura brusca, mostrando a necessidade de uma forte atuação

da bomba, sendo que no decorrer do ensaio a atuação da bomba pode ser reduzida.

Assim, foram planejados ensaios em que a rotação da bomba de propilenoglicol varresse uma faixa de valores que subiam até o valor máximo de 100% de rotação no período inicial do ensaio. A partir daí, a rotação foi reduzida para valores menores e voltava a ser aumentada, porém até valores menores, como 30% de rotação, por exemplo (foram realizados ensaios com diferentes valores para este limite). Foram utilizados diferentes degraus positivos e negativos na rotação da bomba, de forma a ensinar à rede que ela poderia fazer esta variação com diferentes valores. O tempo de amostragem utilizado foi mantido constante durante todo o ensaio, isto é, mantinha-se a rotação da bomba em cada valor pelo mesmo período de tempo.

Para a realização destes ensaios, foram construídas planilhas no Microsoft Excel com as diversas situações de variação da variável manipulada, conforme planejado. Assim, mantendo-se as condições iniciais definidas no planejamento experimental (temperatura de entrada do propilenoglicol de 0°C, vazão do álcool de 0,60 mL/s e rotação da bomba de propilenoglicol de 40%) (LEITE et al., 2008), aguardava-se até que a temperatura do meio precipitante atingisse 5°C e iniciavam-se simultaneamente a adição do álcool e as perturbações, monitorando-se a variável controlada e realizando-se a aquisição de dados via Indusoft. A agitação era mantida constante durante todo o ensaio, num valor de 125 rpm.

#### *3.3.4. Procedimento para os ensaios em malha fechada*

Para a realização dos ensaios em malha fechada, tanto para os ensaios com o controlador PID quanto para os ensaios com o controlador neural, acionava-se a bomba de propilenoglicol, mantendo-a numa rotação de 40%, até a estabilização da temperatura na camisa do tanque (temperatura de entrada do propilenoglicol igual a zero, conforme condições estabelecidas no planejamento experimental).

Adicionavam-se 150 mL de caldo de abacaxi no tanque e aguardava-se até que a temperatura do meio chegasse a 5 °C, valor utilizado como referência, sendo esta uma condição escolhida como ideal para a precipitação, segundo o trabalho de Cesar (1999).

Acionava-se então, quase que simultaneamente, o modo automático do controlador e a bomba peristáltica para a adição do etanol. A condição de parada do ensaio era o tempo decorrido de 1002 segundos, que corresponde ao tempo necessário para se completar 750 mL de líquido no reator (proporção 1:4 suco:álcool (CESAR, 1999)), conforme a vazão da bomba peristáltica.

Nestes ensaios a agitação também foi mantida em 125 rpm durante todo o ensaio.

No caso dos ensaios com o controlador neural, o modo automático correspondeu a iniciar o intercâmbio de informações entre o Indusoft e o Excel.

### *3.3.5. Tratamento de resíduos*

Após a realização dos ensaios, obtinha-se um resíduo constituído por aproximadamente 80% de álcool e 20% de suco de abacaxi.

Como a única destinação dada aos resíduos pela Faculdade de Engenharia Química (FEQ) era a incineração, optou-se por recuperar o álcool numa coluna de destilação em batelada existente no Laboratório de Ensino de Engenharia Química da FEQ.

Para análise da composição do produto obtido foi utilizado um refratômetro, tendo sido necessário realizar-se a calibração do mesmo.

A calibração do refratômetro foi realizada preparando-se soluções de suco de abacaxi e álcool, com concentrações conhecidas, e os respectivos valores do índice de refração foram anotados para a construção da curva de calibração.

Utilizou-se uma razão de refluxo de 0,1, para um volume de 2 L de resíduo por batelada.

Para operação da coluna, aguardava-se até que o regime permanente fosse atingido, isto é, até que a temperatura do topo fosse aproximadamente igual à do fundo da coluna, para se iniciar a coleta do destilado. A temperatura do fundo era mantida constante, em torno de 78°C (temperatura de ebulição do etanol), evitando-se assim arraste de água para o destilado. Quando a temperatura do fundo começava a aumentar, indicando que água estava sendo evaporada, terminava-se a destilação.

### **3.4. Calibração de bombas e sensores**

#### *3.4.1. Bombas*

A vazão da bomba peristáltica, responsável pela adição do etanol no tanque de precipitação, deve ser mantida constante para todos os ensaios. Assim, foi preciso realizar a calibração desta bomba, com o auxílio de um cronômetro e uma proveta.

Para cada valor de rotação da bomba (de zero a 100% de rotação, com degraus de 10%), media-se o tempo necessário para se completar determinado volume (1000 mL, por exemplo). Os valores obtidos destas duas grandezas foram anotados e construiu-se uma curva de calibração desta bomba (vazão no eixo das ordenadas e rotação no eixo das abscissas).

A bomba de propilenoglicol também foi calibrada, seguindo-se o mesmo procedimento utilizado para a bomba peristáltica, desta vez variando-se a posição do seletor. A curva de calibração foi construída com a vazão de etanol no eixo das ordenadas e a posição do seletor no eixo das abscissas.

#### *3.4.2. Sensores de temperatura*

Os sensores de temperatura foram calibrados utilizando-se um termômetro-padrão de mercúrio e um banho termostático de solução de propilenoglicol a 50% em volume.

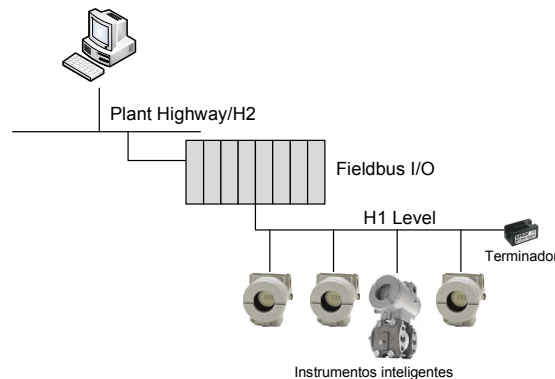
Foi varrida a faixa de -20 a 50°C e o procedimento foi realizado da seguinte forma: mergulharam-se os sensores e o termômetro-padrão próximos

uns aos outros na solução de propilenoglicol do banho termostático. A cada temperatura definida no painel do banho termostático, após aguardar-se até que a mesma se estabilizasse, anotavam-se os valores marcados por cada sensor e o valor correspondente da temperatura medida pelo termômetro-padrão.

Assim, foi possível construir-se curvas de calibração de cada sensor (temperatura do termômetro-padrão no eixo das ordenadas e temperatura do sensor no eixo das abscissas).

### 3.5. Rede *Fieldbus*

O sistema *Fieldbus* empregado está instalado no Laboratório de Controle e Automação de Processos (LCAP) do Departamento de Engenharia de Sistemas Químicos (DESQ/FEQ/UNICAMP) e foi montado com a topologia de barramento com *spurs* (Figura 3.11).



Fonte: Smar Equipamentos Industriais Ltda

Figura 3.11: Topologia da rede *Fieldbus* utilizada (barramento com *spurs*)

A rede é constituída de módulos de entrada e saída analógicas e digitais e uma placa DFI. Acoplados ao barramento principal, há instrumentos inteligentes *Foundation Fieldbus* da Série 302 do fabricante Smar:

- dois transmissores de temperatura TT302: conversão do sinal de resistência para sinal digital (para monitoramento das temperaturas do processo);

- um conversor FI302: conversão do sinal digital para sinal analógico (para atuação da bomba de propilenoglicol);
- um conversor IF302: conversão do sinal analógico para sinal digital (para monitoramento da potência da bomba e do nível do tanque).

Para evitar interferências no sinal da rede, foi instalado um terminador BT302 no final do barramento.

### 3.6. Desenvolvimento e Sintonia do Controlador Convencional PID

O controlador convencional Proporcional Integral Derivativo (PID) é o mais largamente utilizado na indústria, pela facilidade de implementação e sintonia, não exigindo mão-de-obra especializada para tal.

Os *softwares* de configuração de Controladores Lógicos Programáveis (CLP) e de Redes Digitais como a Rede *Fieldbus Foundation* possuem atualmente blocos funcionais que já incluem em sua programação os cálculos de um controlador PID, sendo necessário que o operador apenas insira os valores dos parâmetros do controlador: ganho ( $K_c$ ), constante de tempo integral ( $\tau_I$ ) e constante de tempo derivativa ( $\tau_D$ ). No projeto desenvolvido, utilizou-se como controlador PID o bloco funcional do *software* Syscon da Smar. A estratégia de controle utilizada pode ser visualizada na Figura 3.12:

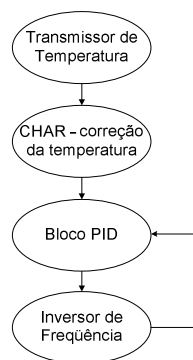


Figura 3.12: Estratégia de controle configurada no Syscon.

Para a sintonia inicial do controlador PID, foi aplicado o Método de Ziegler-Nichols 1, descrito pela Equação 3.7 à Equação 3.9 (SEBORG, EDGAR e MELLICHAMP, 1989). Este método é aplicável a sistemas contínuos, o que foge à proposta do trabalho. Mas como não se encontra na literatura métodos desenvolvidos para sistemas operando em batelada alimentada, optou-se por utilizá-lo mesmo assim, para ter um ponto de partida.

$$K_c = \frac{1,2\tau}{K \cdot t_d} \quad \text{Equação 3.7}$$

$$\tau_I = 2t_d \quad \text{Equação 3.8}$$

$$\tau_D = 0,5t_d \quad \text{Equação 3.9}$$

$$\text{Validade das equações: } 0,1 < \frac{t_d}{\tau} < 1.$$

Como já dito anteriormente, neste método, para obtenção da curva de reação (em formato S) é necessário realizar-se os ensaios em regime contínuo. Assim, para simular esta situação, utilizou-se um artifício baseado no princípio de vasos comunicantes, em que o líquido era alimentado (por uma bomba de engrenagens) e retirado do tanque à mesma vazão, continuamente, através de uma mangueira instalada à saída do tanque, visando manter o nível do tanque constante. A Figura 3.13 ilustra a montagem descrita:



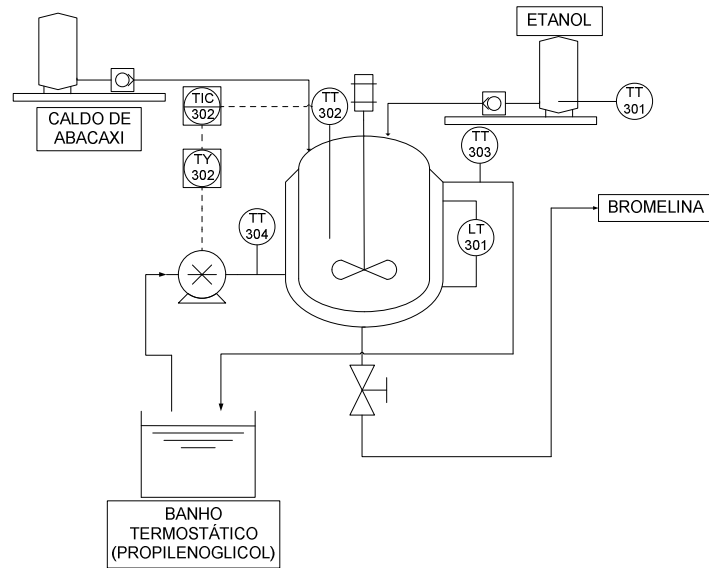


Figura 3.13: Diagrama da montagem utilizada nos ensaios de sintonia do controlador PID.

Para a realização dos ensaios para obtenção da curva de reação, adicionava-se 150 mL de caldo de abacaxi no tanque, sob agitação constante. Quando a temperatura do meio atingia 5,0°C, iniciava-se simultaneamente a adição do álcool e aplicava-se uma perturbação em degrau na variável manipulada. Esta perturbação fazia a bomba sair da condição inicial, de 40% de rotação, para 70% de rotação. Caldo de abacaxi a 5,0°C era alimentado ao tanque paralelamente à adição de álcool.

A curva de resposta obtida foi submetida a um ajuste não linear, segundo a função de transferência de um modelo de 1ª ordem com tempo morto (SEBORG, EDGAR e MELLICHAMP, 1989):

$$G(s) = \frac{K \exp(-t_d \cdot s)}{\tau \cdot s + 1} \quad \text{Equação 3.10}$$

Pelo método gráfico, os parâmetros do processo: ganho ( $K_p$ ), constante de tempo ( $\tau$ ) e tempo morto ( $t_d$ ) podem ser obtidos conforme esquematizado na Figura 3.14:

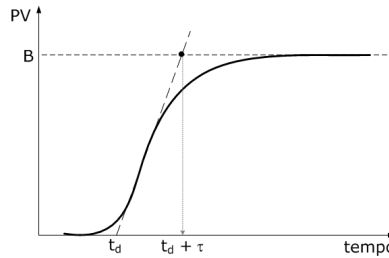


Figura 3.14: Curva de reação para determinação dos parâmetros do processo para sintonia através do método de Ziegler-Nichols.

O ganho do processo é obtido pela Equação 3.11:

$$K_p = \frac{B}{A} \quad \text{Equação 3.11}$$

Onde  $B$  é o novo valor do estado estacionário e  $A$  é o valor da perturbação em degrau.

Com os parâmetros do processo em mãos, utilizam-se as equações do método para obter os parâmetros do controlador.

### 3.7. Comparação entre os controladores

Para a comparação entre os controladores, foram analisados os critérios:

- integral do valor absoluto do erro vezes tempo (ITAE), pois este critério leva em consideração os erros persistentes ao longo do tempo;

- *overshoot*, pois este parâmetro interfere de forma relevante na atividade enzimática, pois define o limite de temperatura de desnaturação da bromelina em solvente orgânico;

- tempo de acomodação, parâmetro que indica melhor desempenho para o controlador que apresentar o menor valor;

- consumo energético, critério este que se traduz em retorno financeiro.

## CAPÍTULO 4: RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados obtidos e as respectivas análises, tendo-se acrescentado eventuais explicações adicionais, quando necessário. Primeiramente são apresentados os resultados dos ensaios em malha aberta que constituem o banco de dados do processo. Em seguida, apresentam-se as telas construídas no Indusoft. Na sequência, apresentam-se os resultados obtidos para o controlador baseado no modelo neural e os resultados obtidos com o controlador convencional. Por fim, são apresentados os parâmetros utilizados para a comparação entre os controladores.

### 4.1. Construção do banco de dados – ensaios em malha aberta

Como primeiro passo para esta etapa, foi realizado um ensaio em malha aberta, com a bomba fixa em 40% de rotação, para que se pudesse observar o comportamento da variável controlada. Este resultado é apresentado na Figura 4.1.

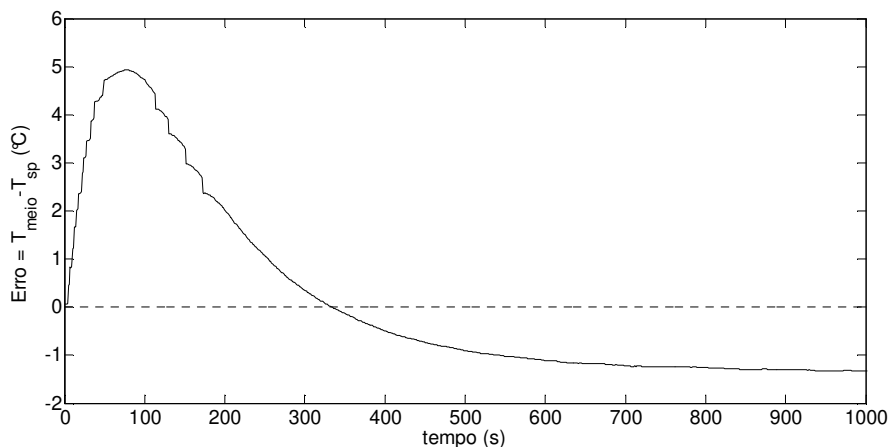
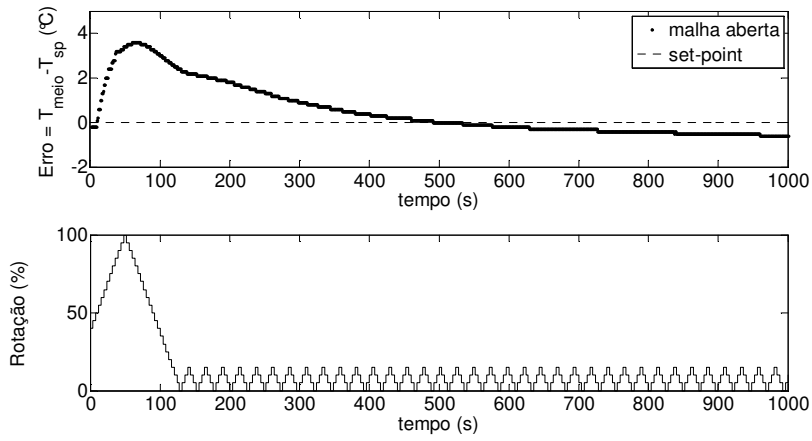


Figura 4.1: Comportamento da variável controlada com a variável manipulada fixa em 40% de rotação

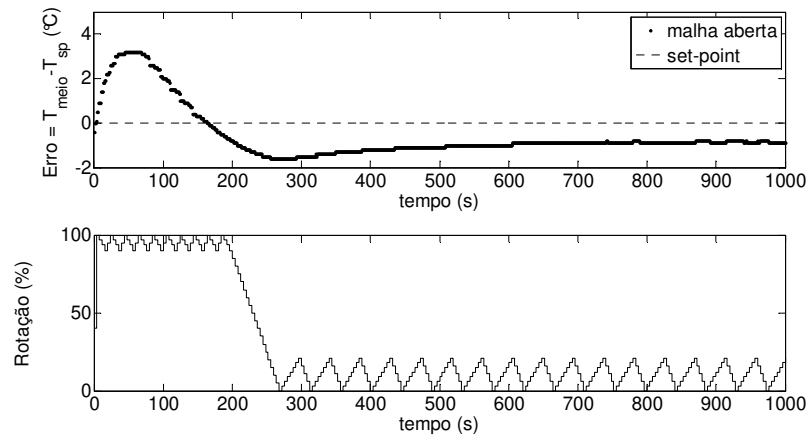
Este perfil pode ser explicado por dois fatores: uma vez que o processo é realizado em batelada, tem-se uma variação na área de troca térmica ao longo do ensaio e uma variação também no efeito do calor de dissolução do álcool; no início do ensaio, não há nenhum álcool no meio precipitante, apenas caldo de abacaxi e portanto o calor de dissolução do álcool apresenta um efeito expressivo, que é reduzido significativamente ao final da batelada, quando já existe uma grande quantidade de álcool no tanque. Aliado a isto, existe a variação na área de troca térmica, que no início do ensaio é bastante reduzida uma vez que há uma pequena quantidade de líquido no tanque. Ao final do ensaio, com uma quantidade maior de líquido no tanque, aumenta-se a área de troca térmica, reduzindo-se a necessidade de refrigeração.

Observando a Figura 4.1, conclui-se que o comportamento esperado da variável manipulada, quando da atuação do controlador, tenha um perfil de atuação máxima no início do processo e uma atuação menos intensa com o avanço da batelada.

Assim, visando cobrir esta faixa de atuação, em princípio foram realizados ensaios com degraus, positivos e negativos, de 3%, 5%, 7%, 10%, 12%, 15% e 20%, com uma amostragem de 4 segundos, dado que a bomba de propilenoglicol levava de 2 a 3 segundos para responder; essas perturbações faziam com que a rotação da bomba variasse de 40 a 100% no início, diminuindo em seguida e depois variando de zero a valores menores (até 10%, 15%, 20% ou 30%), de forma contínua e uniforme. Dois destes ensaios podem ser visualizados na Figura 4.2. Com isto, foram obtidos 38 ensaios. Também foram realizados ensaios em que a bomba ficasse completamente desligada, após subir até 100% (Figura 4.3). Novos ensaios foram realizados, selecionando aleatoriamente algumas planilhas, modificando apenas a parte final, deixando a bomba zerada. Nesta etapa mais 4 ensaios foram acrescentados ao treinamento.



(a)



(b)

Figura 4.2: Perturbações na variável manipulada (ensaios em malha aberta) com (a) degraus uniformes e (b) degraus variados

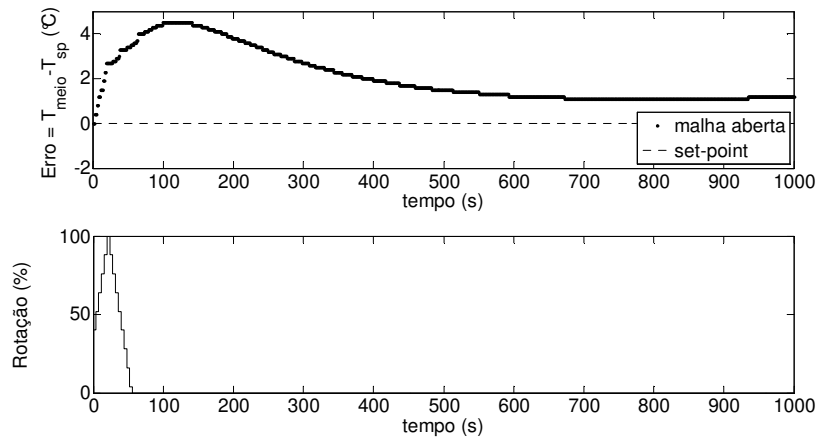


Figura 4.3: Ensaio em malha aberta com a bomba desligada

Conforme os testes iam progredindo, suas análises levavam à realização de novos ensaios em malha aberta, a serem acrescentados no treinamento.

Desta forma, sentiu-se a necessidade de se acrescentar ensaios semelhantes aos iniciais, porém com desligamentos da bomba em diversos momentos do ensaio, conforme a Figura 4.4.

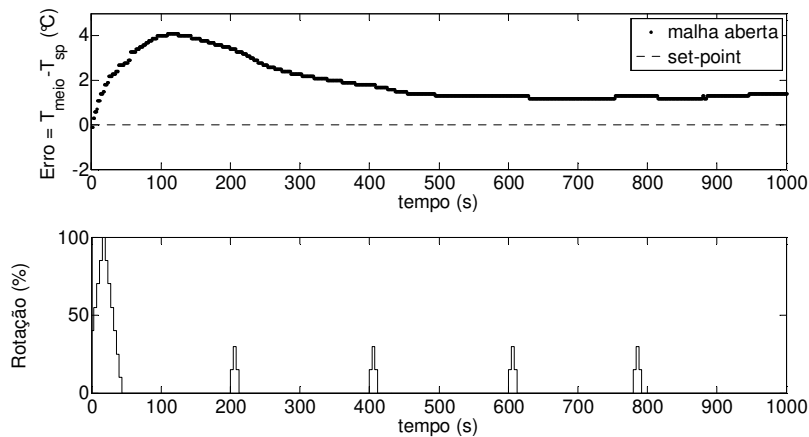


Figura 4.4: Ensaio em malha aberta com desligamentos e religamentos da bomba

## 4.2. Desenvolvimento do Sistema Supervisório

As telas de interface construídas para cada tipo de controle são apresentadas na Figura 4.5 à Figura 4.8. Para a criação destas interfaces, priorizou-se a facilidade de interação com o operador, alocando-se as variáveis de processo em posições estratégicas para facilitar a visualização das mesmas.

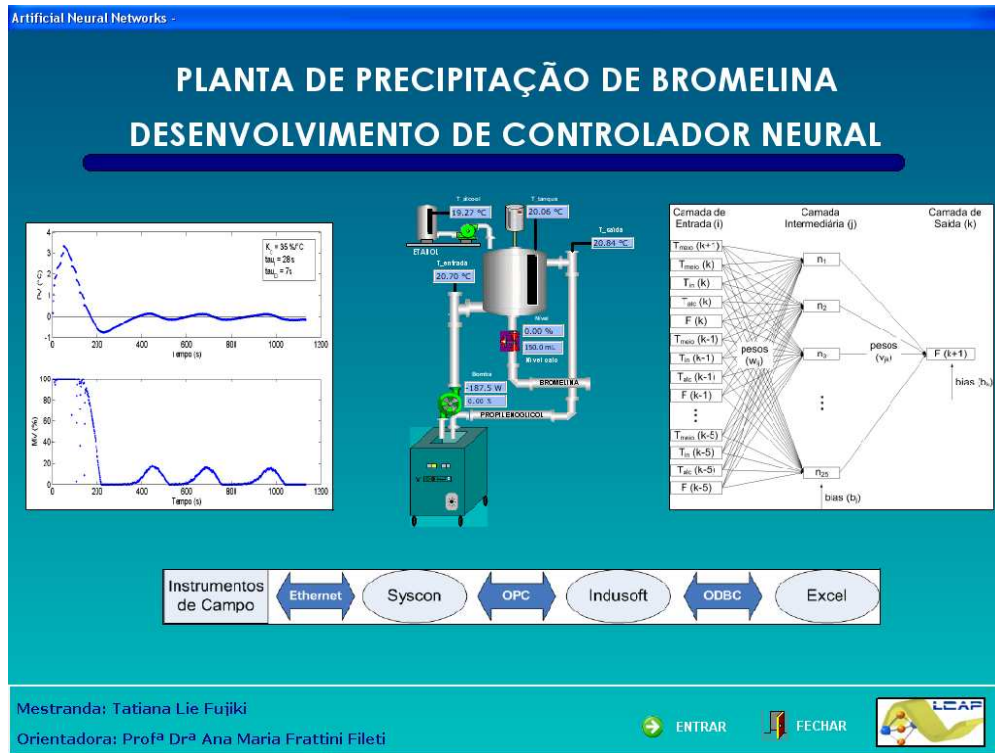


Figura 4.5: Tela inicial do sistema supervisório

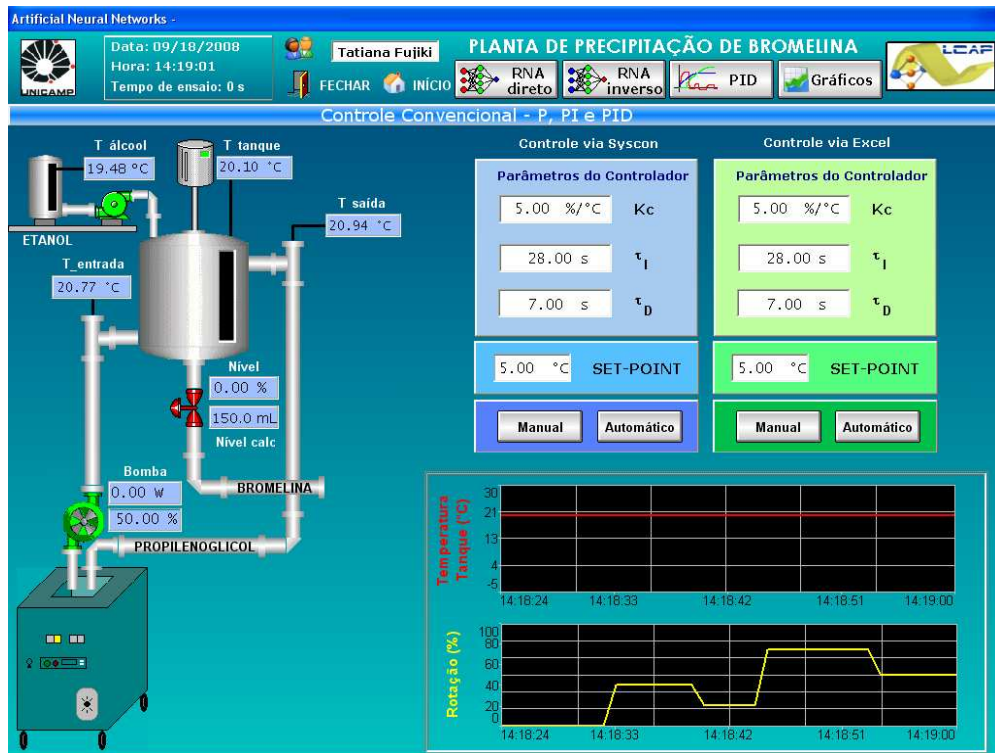


Figura 4.6: Tela do controlador PID do sistema supervisório

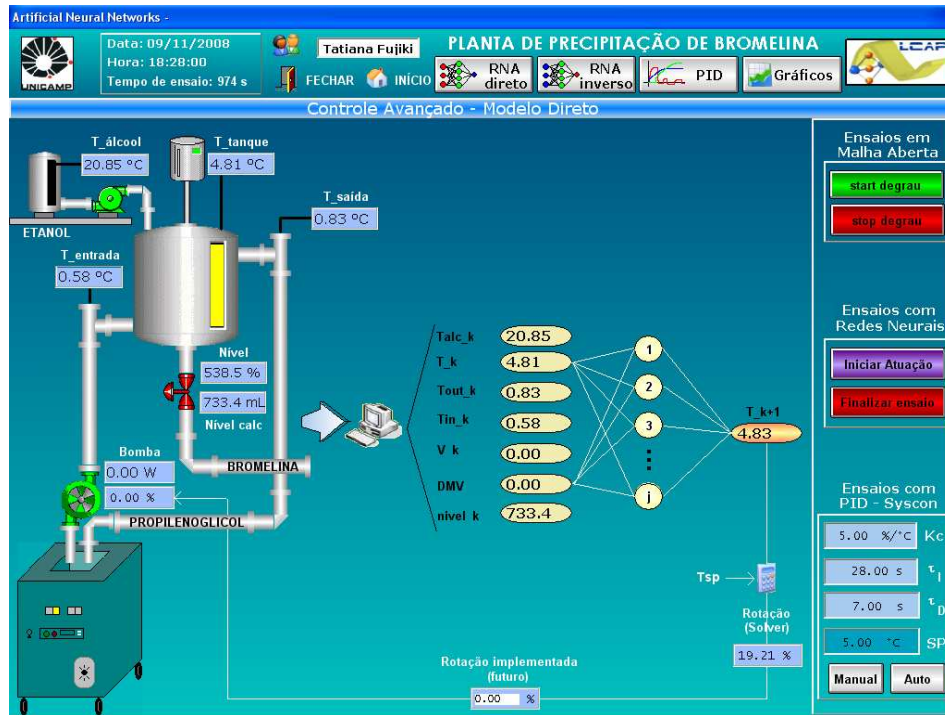


Figura 4.7: Tela do controlador neural do sistema supervisorio

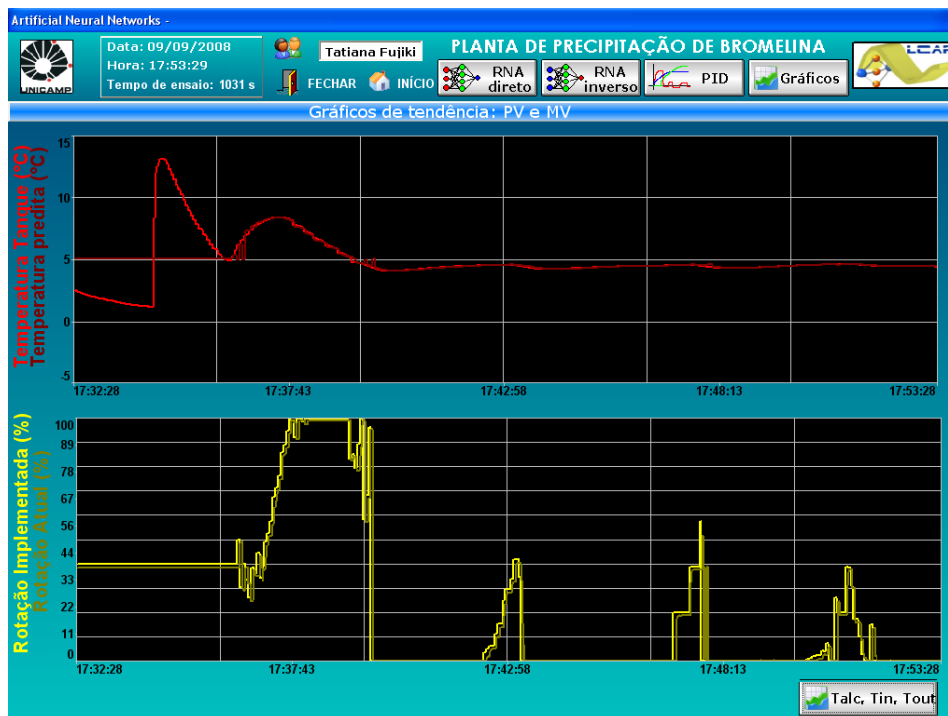


Figura 4.8: Tela de gráficos do sistema supervisorio



### 4.3. Modelagem do processo por Redes Neurais Artificiais

Quando se propõe trabalhar com redes neurais artificiais, o primeiro passo deve ser a definição das variáveis de entrada da rede. Neste estágio, o conhecimento sobre o processo pode ser de grande valor, para que se faça uma identificação pertinente das variáveis do processo, sugerindo-se a melhor combinação entre as mesmas.

Esta etapa envolveu diversas tentativas, isto é, diferentes combinações de variáveis de entrada foram utilizadas, de forma que o treinamento fosse suficiente para a rede responder adequadamente. Neste caso, essa resposta adequada foi avaliada comparando-se a temperatura do meio precipitante medida pelo sensor no campo e a calculada pelo modelo neural nos testes em modo *offline*, utilizando o MATLAB. Testes *online* também foram realizados, somente para os modelos considerados bons no teste *offline*. Foram utilizados como critérios de avaliação os parâmetros do ajuste linear (coeficientes angular, linear e de determinação) dos gráficos de temperatura real *versus* temperatura calculada pelo modelo neural. Uma avaliação visual também foi feita para verificar a concordância entre os pontos experimentais e os calculados.

A Tabela 4.1 contém, de forma sucinta, uma comparação entre os modelos estudados.

#### 4.3.1. Modelo Neural 1

As variáveis escolhidas inicialmente para a modelagem do processo de precipitação da bromelina foram:

- temperatura do álcool etílico no instante atual ( $T_{alc,k}$ ): o álcool foi mantido sempre à temperatura ambiente e por esta razão, sua temperatura estava sujeita a alterações conforme o tempo meteorológico, apesar da existência de um ar-condicionado, uma vez que a planta estava alocada numa sala que não era isolada termicamente;

Tabela 4.1: Comparação entre os modelos testados

Modelo	Entrada RNA							Saída RNA
	$T_{alc,k}$	$T_{meio,k}$	$T_{in,k}$	$T_{out,k}$	$L_k$	$MV_k$	$\Delta MV_k$	
Modelo Neural 1	x		x	x		x		$T_{meio,k}$
Modelo Neural 2	x		x	x	x	x		$T_{meio,k}$
Modelo Neural 3	x		x	x	x	x		$T_{meio,k}$
Modelo Neural 4a	x	x	x	x	x	x	x	$T_{meio,k+1}$
Modelo Neural 4b*	x	x	x	x	x	x	x	$T_{meio,k+1}$
Modelo Neural 5	x		x	x	x	x	x	$T_{meio,k+1}$
Modelo Neural 6	x	x	x	x	x	x		$T_{meio,k+1}$

\* A estrutura utilizada neste treinamento foi a mesma do Modelo Neural 4a, com a única diferença do acréscimo de novos ensaios ao treinamento.

- temperatura de entrada ( $T_{in,k}$ ) e de saída ( $T_{out,k}$ ) do propilenoglicol na camisa no instante atual: estas duas variáveis fornecem informações à rede, a respeito da troca térmica na camisa de resfriamento do tanque de precipitação;

- rotação da bomba de propilenoglicol no instante atual ( $MV_k$ ): esta é a variável manipulada do processo e é justamente o seu valor que influenciará o valor da variável controlada, sendo, portanto, uma variável importante na rede neural.

A variável de saída utilizada foi a temperatura do meio precipitante no instante atual ( $T_{meio,k}$ ).

Convém destacar que este treinamento foi realizado paralelamente à realização dos ensaios em malha aberta e assim, o banco de dados ainda estava incompleto, com apenas 16 bateladas.

Na camada intermediária, tomou-se como número de neurônios o dobro do número de variáveis na camada de entrada para cada um dos modelos (heurístico).

Na Figura 4.9 apresenta-se a estrutura utilizada neste modelo:

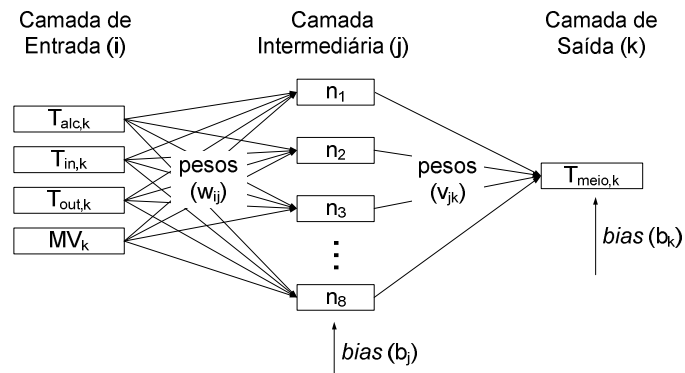


Figura 4.9: Estrutura da RNA para o Modelo Neural 1.

Com esta arquitetura, o treinamento utilizando o algoritmo de Levenberg-Marquardt levou aos resultados mostrados na Figura 4.10 à Figura 4.12:

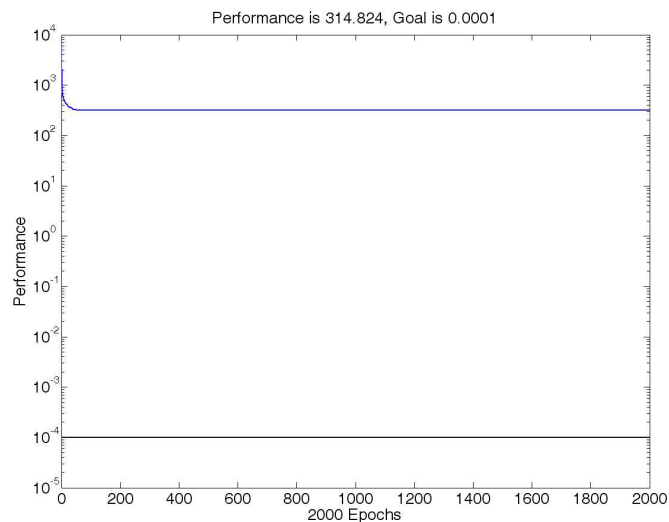


Figura 4.10: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 1.

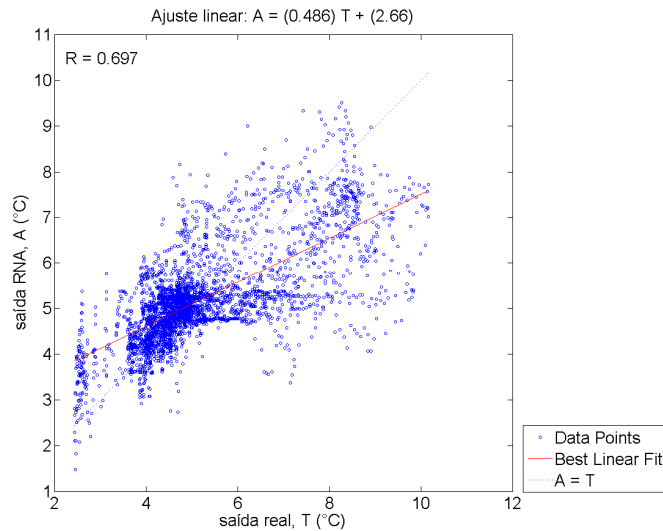


Figura 4.11: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 1.

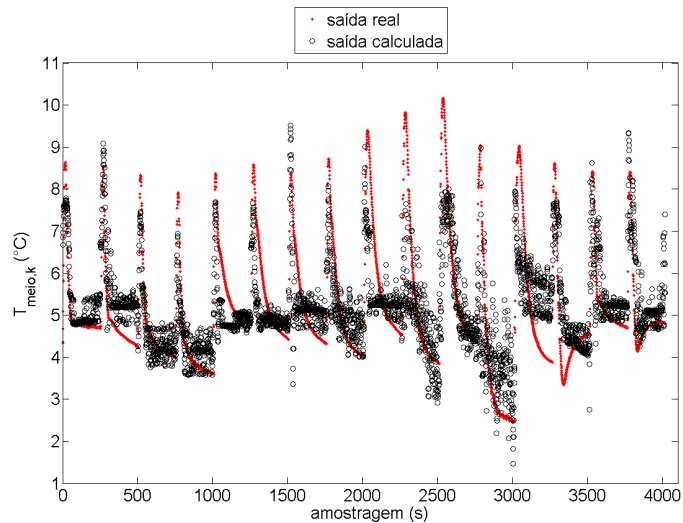


Figura 4.12: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 1.

Como pode ser observado na Figura 4.10, o erro do treinamento permaneceu num valor da ordem de  $10^2$ , quando o desejado era que fosse da ordem de  $10^{-4}$ , tendo estabilizado com menos de 200 iterações. Este erro grande, tão distante da meta, refletiu nos gráficos da Figura 4.11 e da Figura 4.12: pode-se perceber, pela equação do ajuste da reta, que a rede não foi capaz de prever

corretamente a temperatura, indicando que as variáveis de entrada não estavam fornecendo informações suficientes para a rede capturar a dinâmica do processo.

#### 4.3.2. Modelo Neural 2

Assim, a partir dos mesmos dados, realizou-se um novo treinamento, desta vez incluindo como variável de entrada da rede o nível de líquido no tanque, tendo-se utilizado o volume para representar esta variável. O nível de líquido do tanque de precipitação indica à rede o tempo do ensaio, diferenciando assim vetores de entrada iguais correspondendo a um valor diferente de saída. Esta variável também tem a função de fornecer à rede informações a respeito da troca térmica ao longo do experimento. Este modelo pode ser visualizado na Figura 4.13:

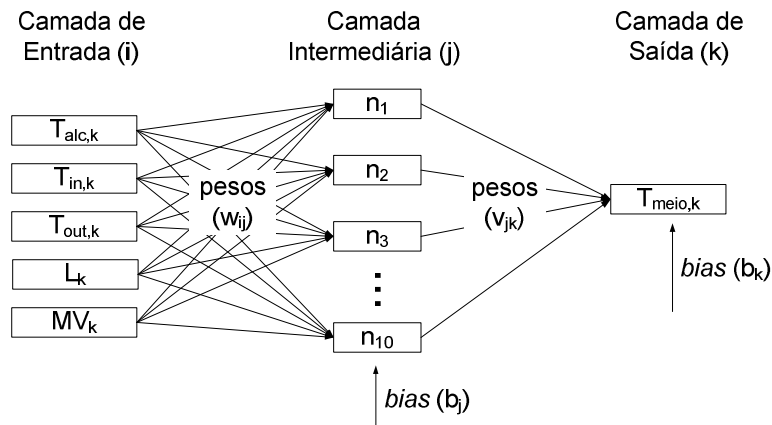


Figura 4.13: Estrutura da RNA para o Modelo Neural 2.

Os resultados correspondentes a esta etapa são mostrados na Figura 4.14 à Figura 4.16.

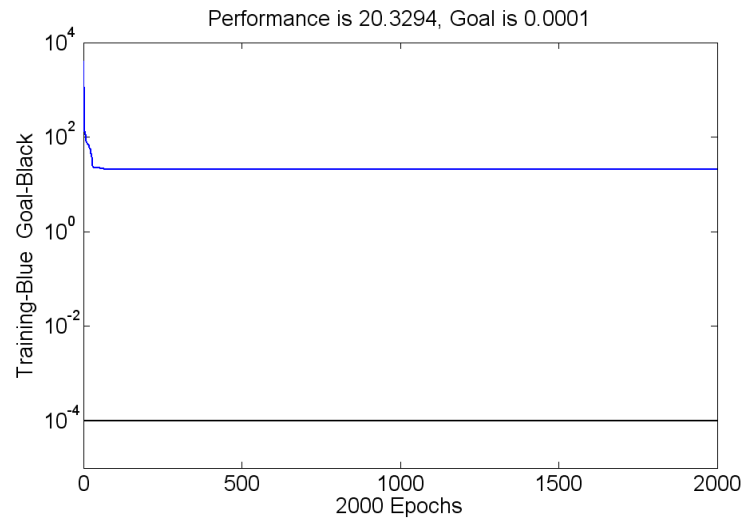


Figura 4.14: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 2.

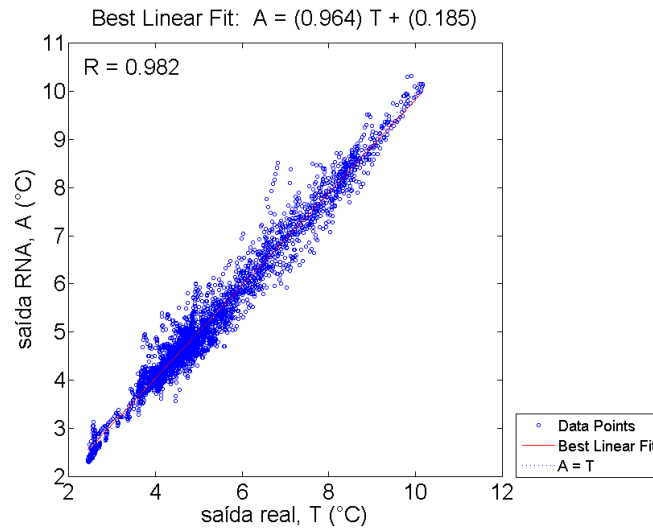


Figura 4.15: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 2.

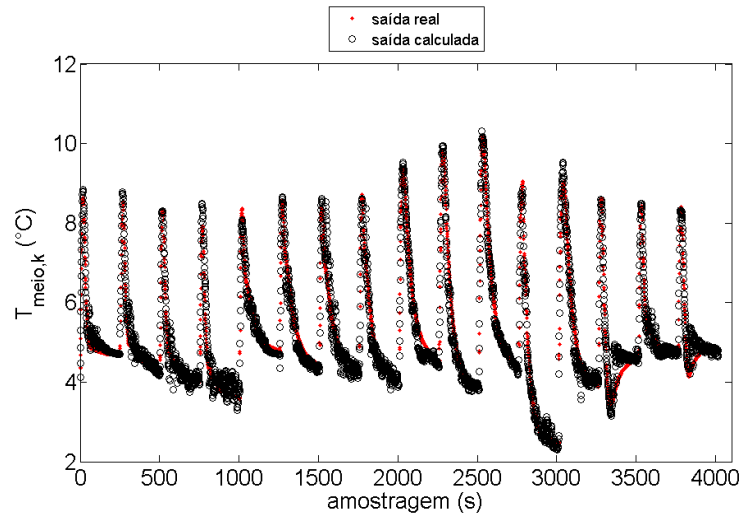


Figura 4.16: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 2.

Pela Figura 4.14 é possível perceber que a introdução da variável nível na camada de entrada da rede de fato contribuiu para que o erro durante o treinamento diminuísse em dez vezes. E este melhor aprendizado da rede pode ser confirmado por uma observação qualitativa da Figura 4.16 e pela comparação dos coeficientes angular e linear do gráfico da Figura 4.15, próximos da unidade e de zero, respectivamente.

Como tratava-se apenas de um treinamento preliminar, os testes *offline* nem chegaram a ser realizados, tendo sido possível a análise somente com os dados de treinamento.

#### 4.3.3. Modelo Neural 3

Com a expressiva melhoria na concordância entre a saída real e a calculada, partiu-se para o treinamento da rede com todos os dados coletados dos ensaios em malha aberta, utilizando 31 bateladas e separando-se 11 ensaios para testes, além de mais dois ensaios em malha fechada utilizando o controlador *fuzzy*. Este foi o treinamento chamado de Modelo Neural 3. Os gráficos obtidos neste treinamento são exibidos na Figura 4.17 à Figura 4.19.

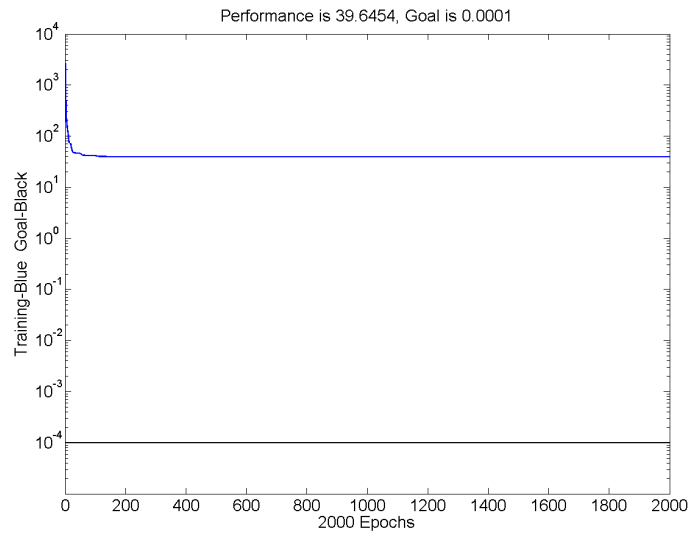


Figura 4.17: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 3.

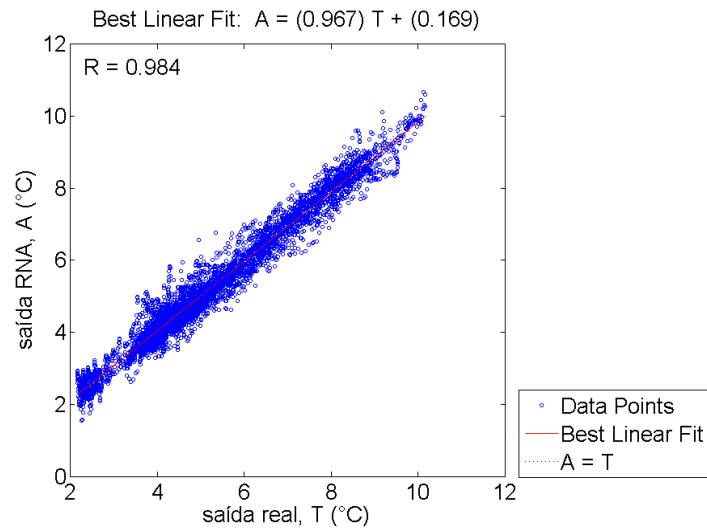


Figura 4.18: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 3.



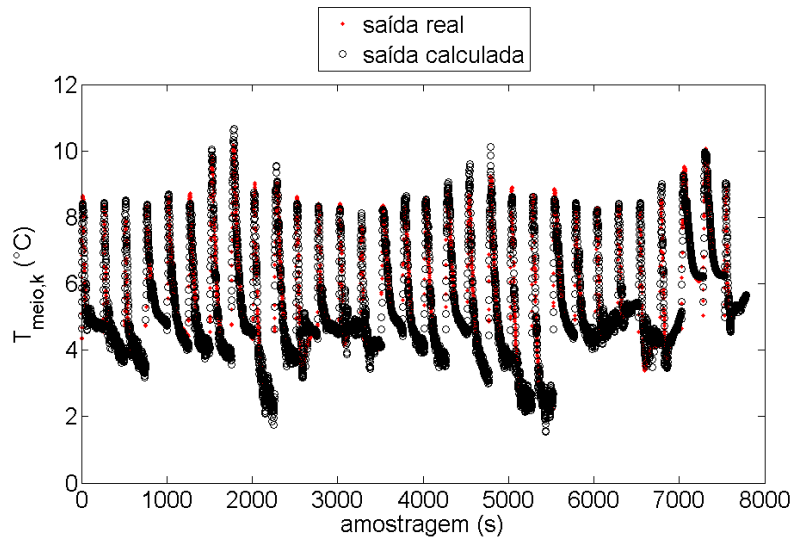
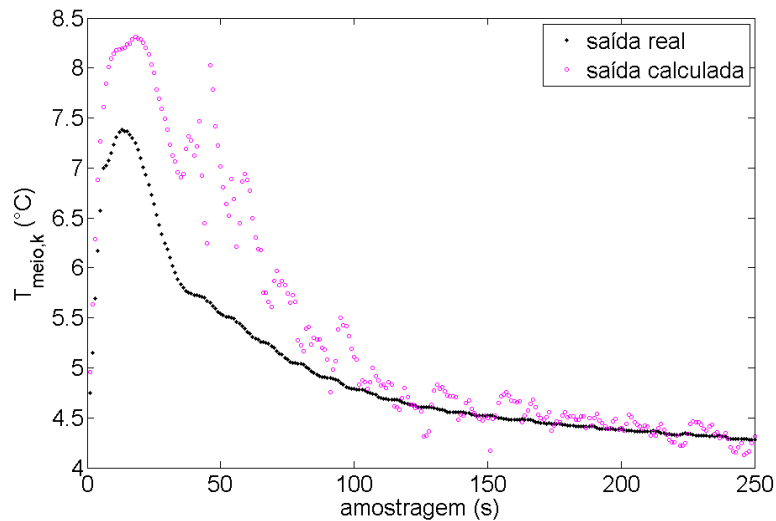
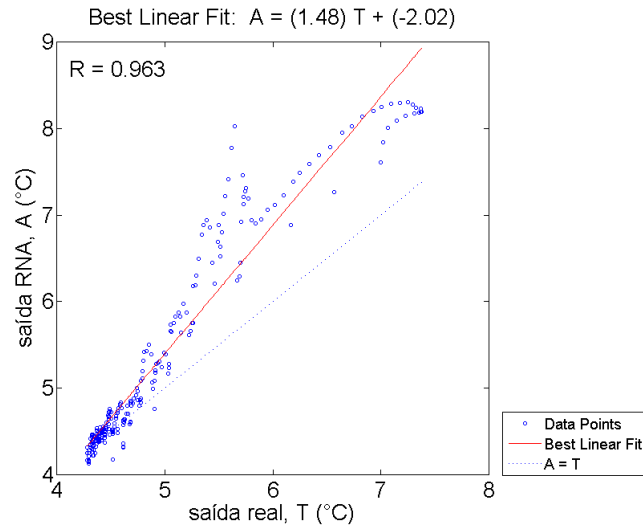


Figura 4.19: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 3.

Os resultados dos testes do modelo são mostrados na Figura 4.20 para um ensaio em malha aberta e na Figura 4.21 para um ensaio em malha fechada (controlador *fuzzy*):

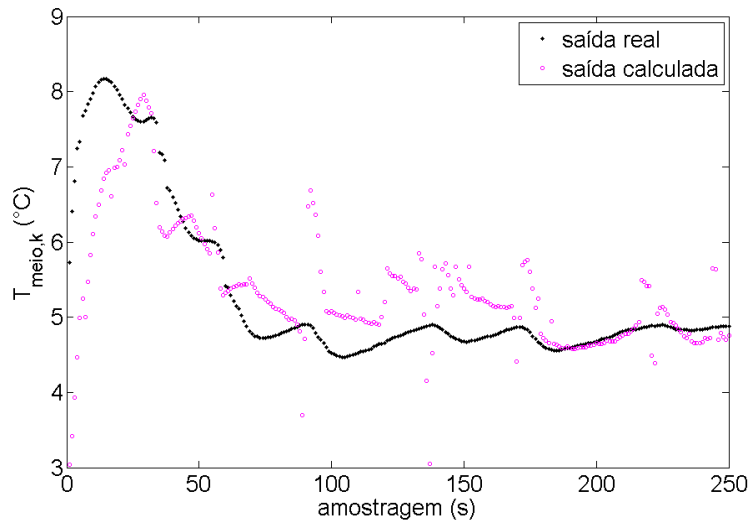


(a)

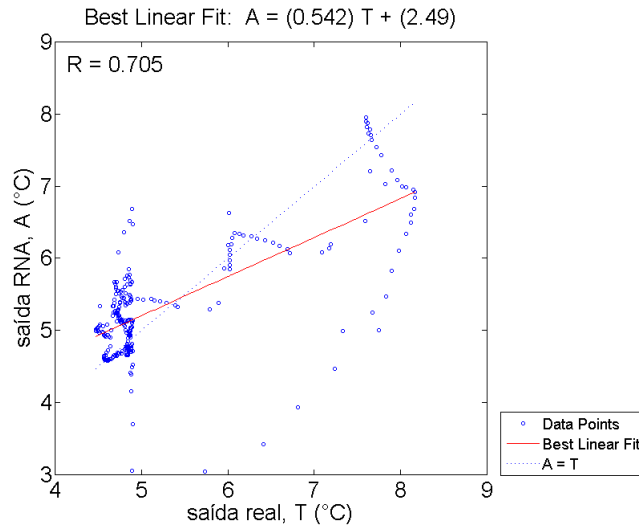


(b)

Figura 4.20: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha aberta, no teste do Modelo Neural 3.



(a)



(b)

Figura 4.21: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha fechada, no teste do Modelo Neural 3.

Como pode ser observado por uma simples análise qualitativa, a rede não foi capaz de prever com exatidão a saída desejada, apesar de ter seguido a tendência do processo. Com isso, concluiu-se que era necessário adicionar ainda alguma variável.

#### 4.3.4. Modelo Neural 4a

A variável controlada do processo reage frente a uma perturbação na variável manipulada. Assim, foi adicionada como variável de entrada da rede a temperatura do meio precipitante no instante atual ( $T_{meio,k}$ ), pois é exatamente a sua variação que determinará a ação da variável manipulada e assim ela serve como uma referência para a rede prever a temperatura do meio precipitante um instante de amostragem à frente ( $T_{meio,k+1}$ ), que foi escolhida como variável de saída da RNA; também foi adicionada a variação da rotação da bomba de propilenoglicol ( $\Delta MV_k$ ), para indicar à rede qual degrau na rotação foi o causador da saída fornecida.

Esta nova estrutura, o Modelo Neural 4a, pode ser observada na Figura 4.22:

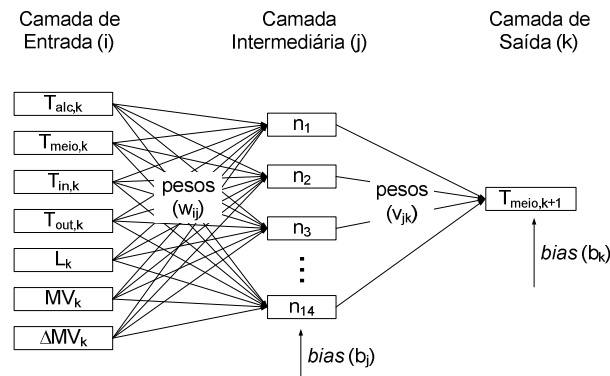


Figura 4.22: Estrutura da RNA para o Modelo Neural 4a.

Com este modelo, os resultados para o treinamento foram os da Figura 4.23 à Figura 4.25.

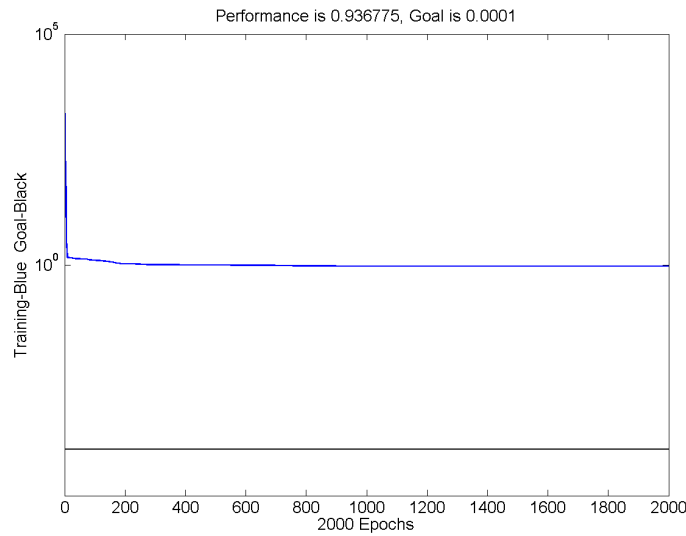


Figura 4.23: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 4a.

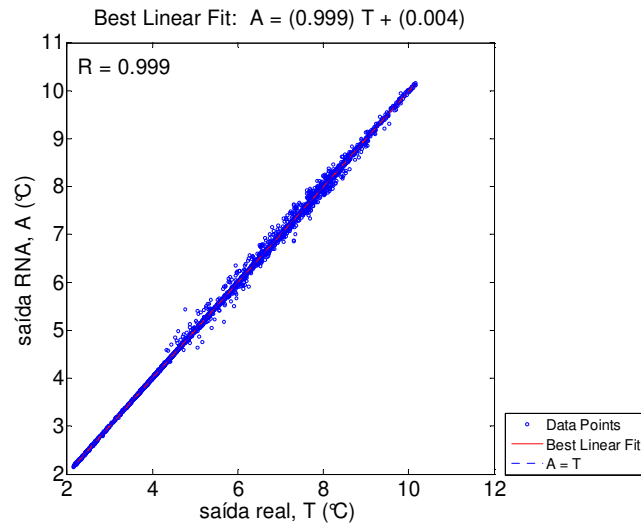


Figura 4.24: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 4a.

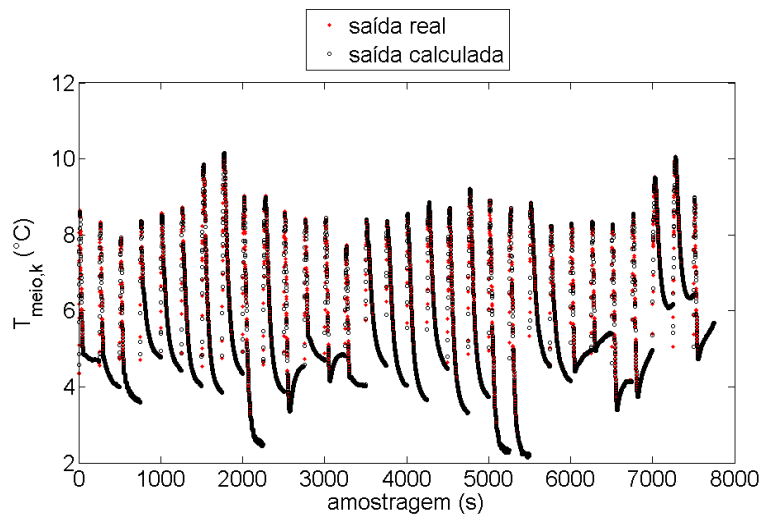
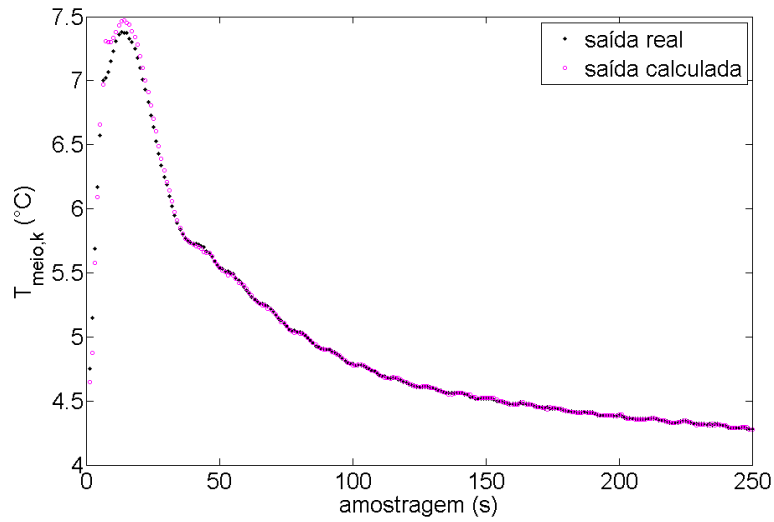
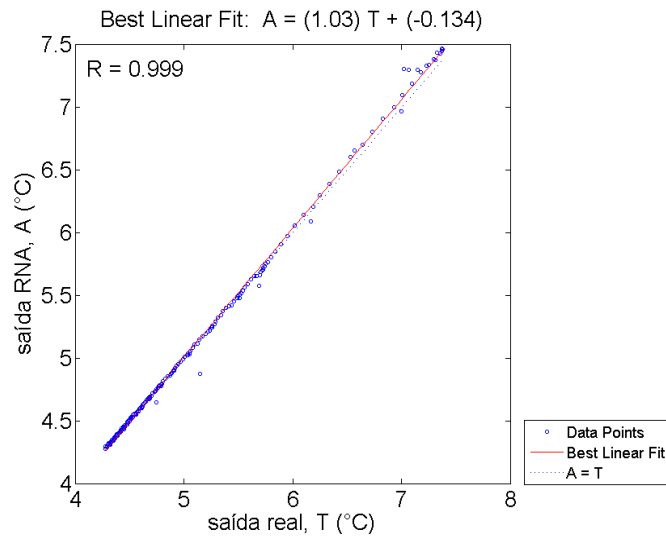


Figura 4.25: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 4a.

Na Figura 4.26 é mostrado o resultado para o teste de uma batelada em malha aberta e na Figura 4.27, para uma batelada em malha fechada.

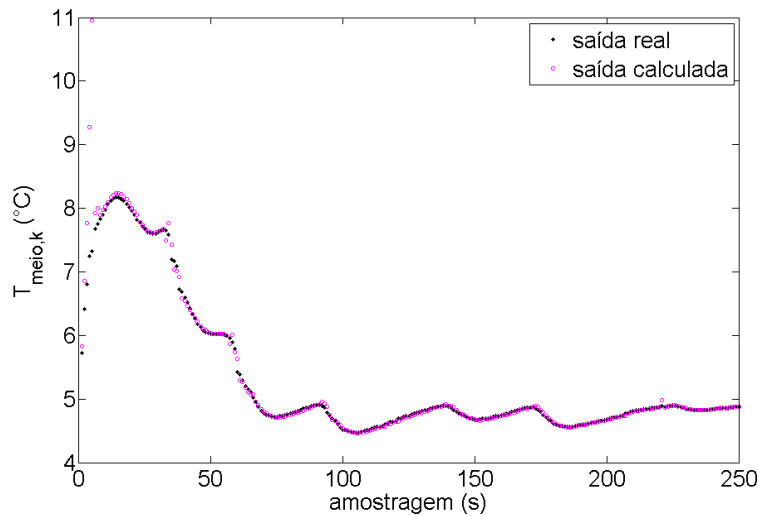


(a)

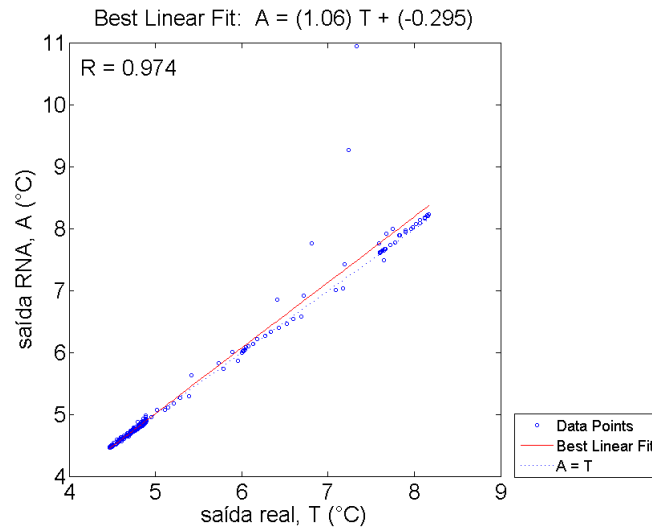


(b)

Figura 4.26: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha aberta, no teste do Modelo Neural 4a.



(a)



(b)

Figura 4.27: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha fechada, no teste do Modelo Neural 4a.

Com esta topologia, a rede foi capaz de prever a saída corretamente, tendo a reta de ajuste da Figura 4.26 coeficientes angular e linear próximos de 1 e de zero, respectivamente. No caso do teste com o ensaio em malha fechada, observam-se alguns erros de previsão apenas no início da batelada.

Sendo assim, partiu-se para a validação *online* do Modelo Neural 4a.

Conforme descrito no item 3.1.4. , “Validação *online* do modelo neural”, realizou-se um ensaio em malha fechada com o controlador PID do *software* Syscon e paralelamente monitorou-se a temperatura calculada pelo modelo na planilha eletrônica. Ao final do ensaio a temperatura medida pelo sensor e a calculada pela RNA foram comparadas, obtendo-se o gráfico da Figura 4.28.

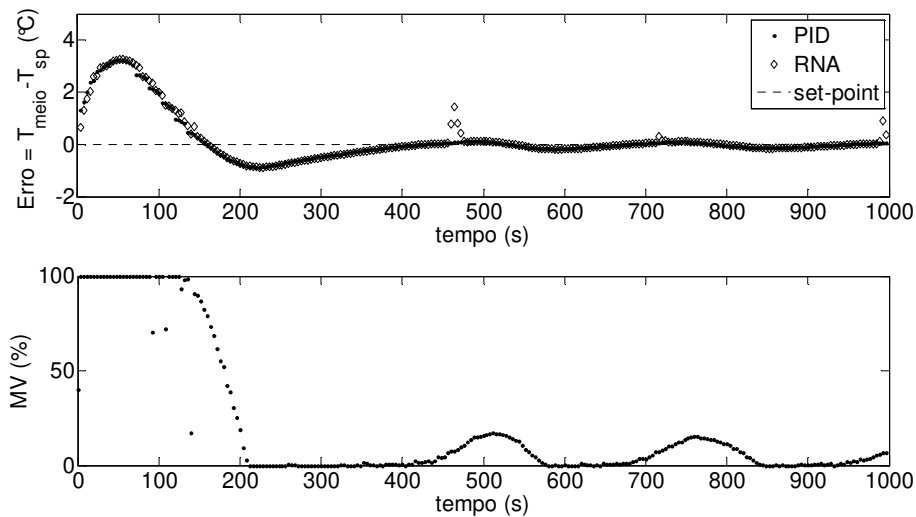


Figura 4.28: Comparação entre a temperatura real e a predita pela RNA na validação *online* do Modelo Neural 4a.

#### 4.3.5. Modelo Neural 4b

Analisando o resultado da validação *online* do Modelo Neural 4a, conforme a Figura 4.29, concluiu-se que os erros de predição deviam-se à elevação brusca da temperatura de entrada do propilenoglicol na camisa. Este fato, por sua vez, é explicado pelo fluido refrigerante ficar estagnado na tubulação no período em que a bomba desliga ( $MV = 0\%$ ). A bomba, ao religar, transmite ao fluido o calor gerado no rotor, causando a elevação brusca observada na temperatura do propilenoglicol.

Desta forma, a rede acaba fazendo uma previsão de que a temperatura do meio precipitante se elevaria, o que na verdade não ocorre. Este problema precisou ser sanado pois estas previsões erradas causariam ações erradas do



controlador, desestabilizando o sistema e comprometendo o controle de todo o processo.

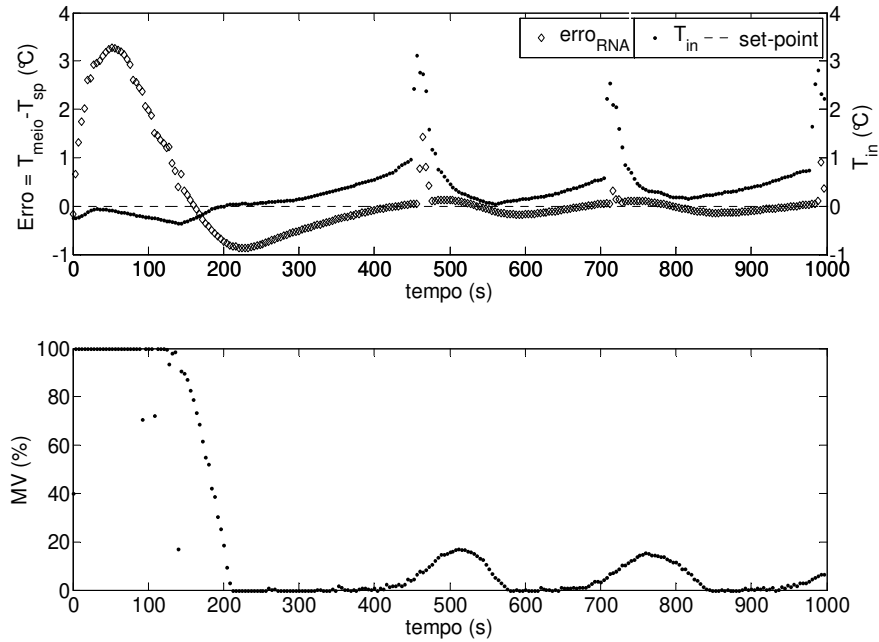


Figura 4.29: Análise da validação *online* do Modelo Neural 4a.

Assim, novos ensaios foram realizados, tendo em vista treinar a rede incluindo situações de desligamentos e re-ligamentos da bomba, em diversos momentos da batelada, para que a RNA aprendesse que mesmo que a temperatura de entrada do propilenoglicol se elevasse, a temperatura do meio precipitante não deveria sofrer alterações bruscas.

Este novo treinamento gerou o chamado Modelo Neural 4b, cujos gráficos são mostrados na Figura 4.30 à Figura 4.32:

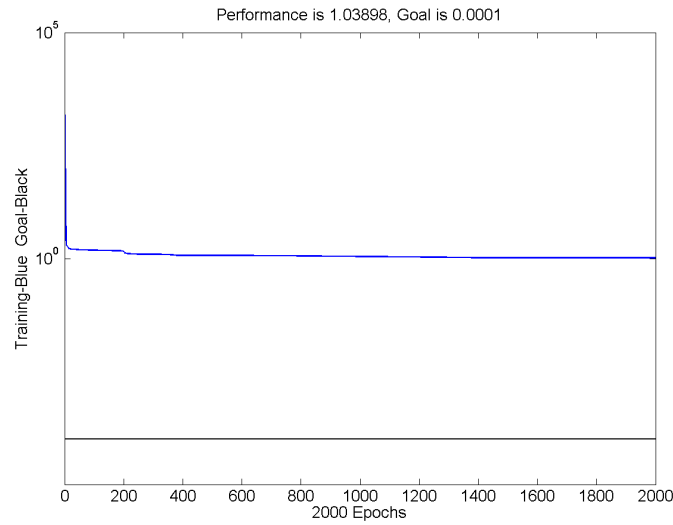


Figura 4.30: Desempenho da RNA durante o treinamento utilizando o algoritmo de Levenberg-Marquardt, para o Modelo Neural 4b.

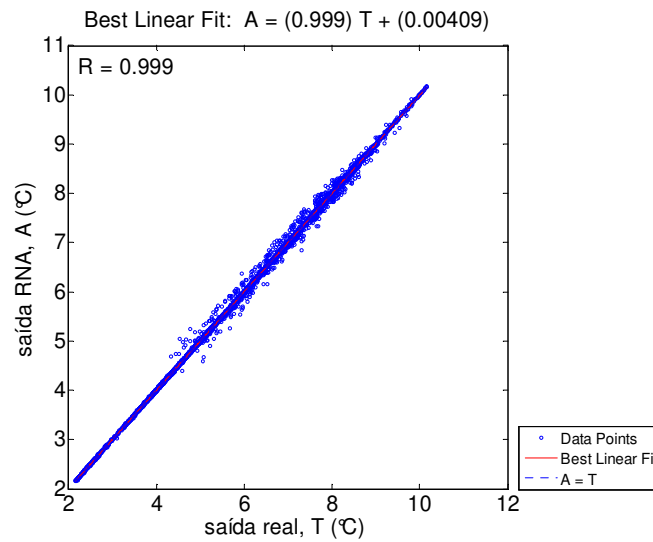


Figura 4.31: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 4b.

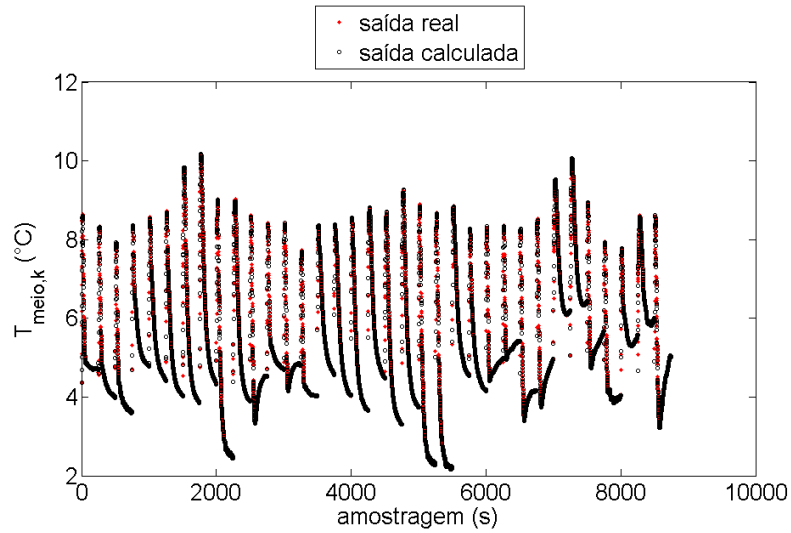
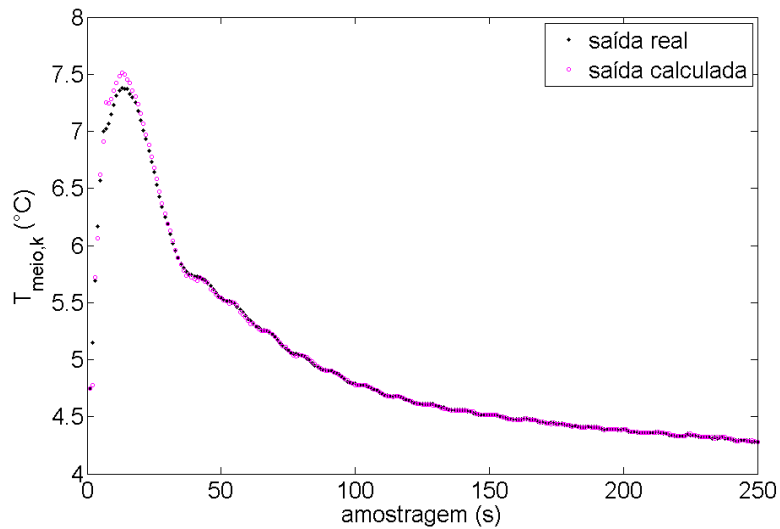
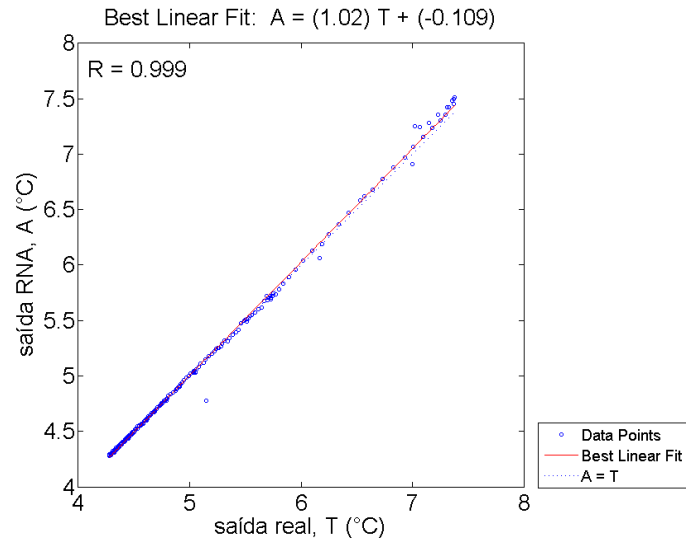


Figura 4.32: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 4b.

Foram realizados então os testes *offline* para este modelo, conforme apresentado na Figura 4.33 e na Figura 4.34:

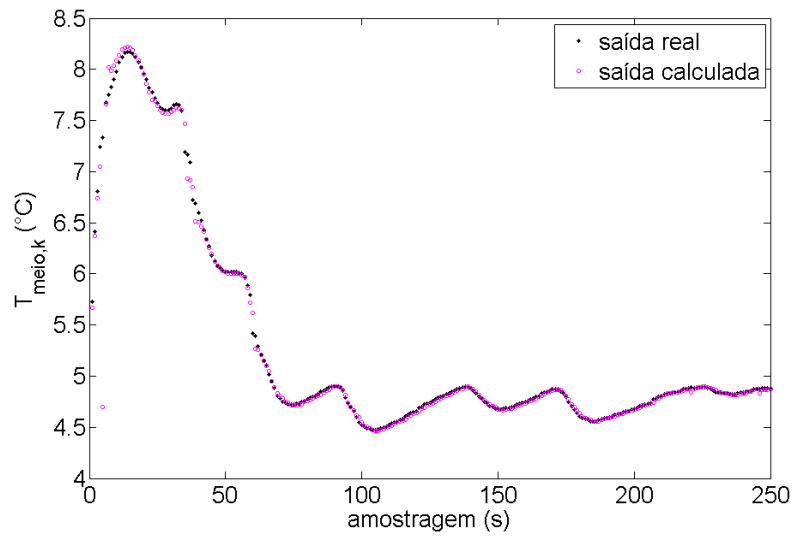


(a)

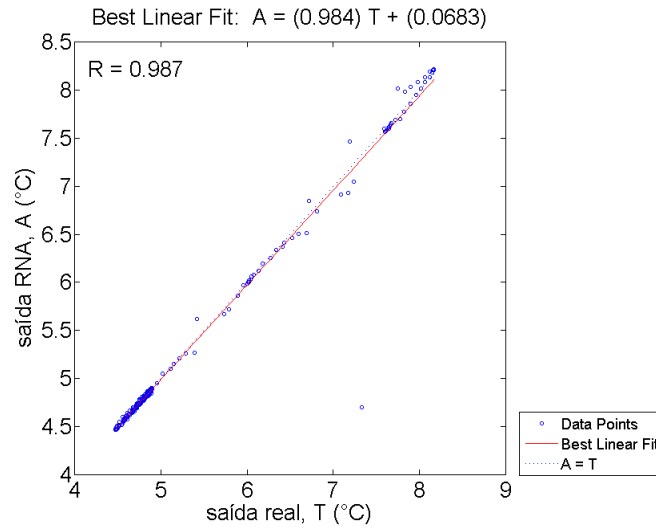


(b)

Figura 4.33: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha aberta, no teste do Modelo Neural 4b.



(a)



(b)

Figura 4.34: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha fechada, no teste do Modelo Neural 4b.

Com os bons resultados observados, a validação do Modelo Neural 4b foi realizada, levando aos resultados mostrados na Figura 4.35.

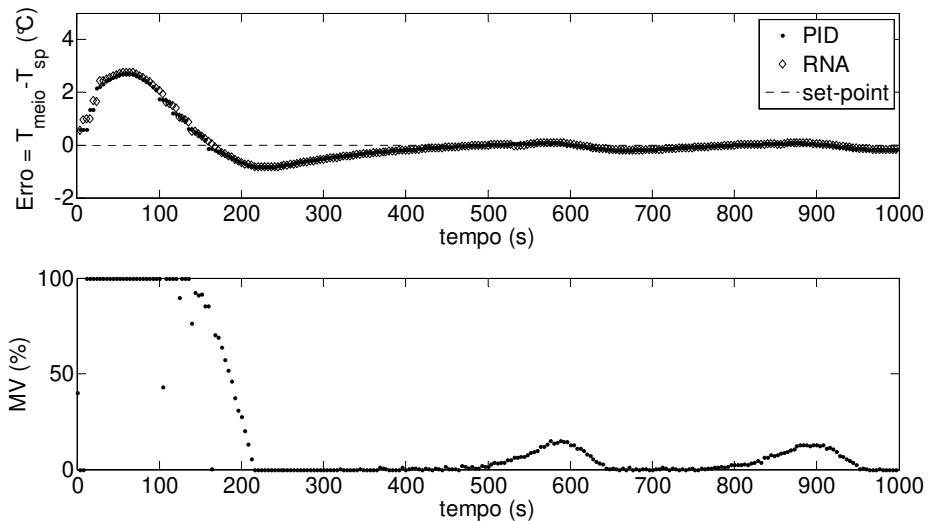


Figura 4.35: Comparação entre a temperatura real e a predita pela RNA na validação *online* do Modelo Neural 4b.

Na Figura 4.36 pode ser observado que mesmo com elevações da temperatura de entrada do propilenoglicol, a previsão da RNA seguiu os valores reais medidos pelo sensor.

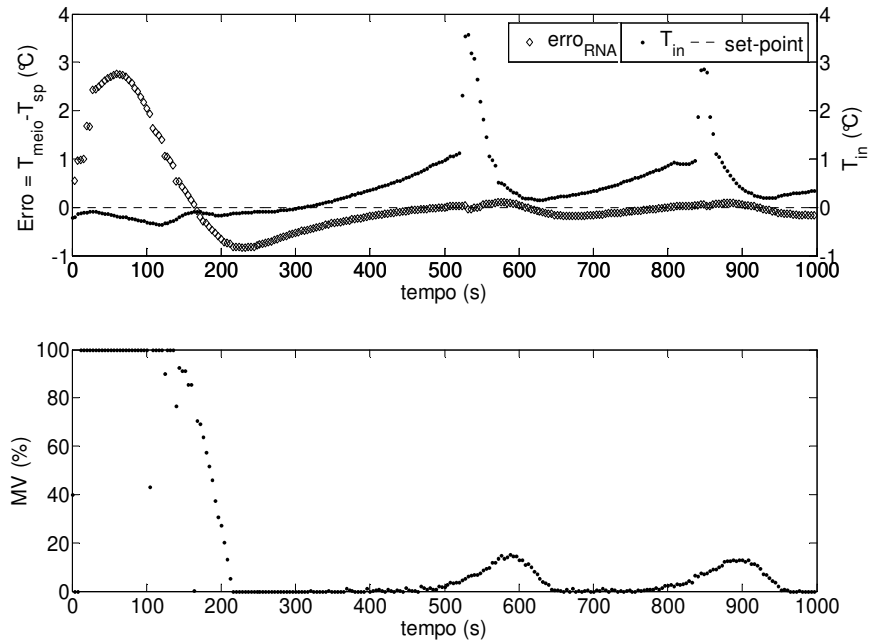


Figura 4.36: Análise da validação *online* do Modelo Neural 4b.

Tendo suprimido as previsões erradas da rede neural, considerou-se o Modelo Neural 4b adequado para utilização no controle da planta com o otimizador. Assim, as matrizes de pesos e *bias* obtidas para este modelo foram:

$$w_{ij} = \begin{bmatrix} -0,1204 & 0,0812 & 2,643 & -0,7350 & 0,3414 & -0,1084 & -8,6438 \\ 1,336 & 8,984 & 13,18 & -36,99 & 4,395 & 14,20 & 94,38 \\ 0,0236 & -0,0097 & -0,3392 & 0,2298 & 0,1107 & -0,1837 & -0,2224 \\ -0,0289 & -0,0387 & 0,4157 & -0,2817 & 0,1107 & -0,1837 & -0,2224 \\ 0,6824 & 1,028 & 12,04 & -6,000 & 3,155 & -2,098 & 36,30 \\ 0,8777 & 1,625 & -9,378 & 10,86 & -2,614 & 4,067 & 6,279 \\ 0,0395 & 2,502 & -13,85 & 5,776 & 1,703 & 10,55 & 9,656 \\ -0,3256 & 0,2587 & 3,034 & -2,871 & 0,8771 & -2,068 & -2,049 \\ -0,3651 & -0,2742 & 3,707 & -3,726 & 1,143 & -2,092 & -1,970 \\ -0,3815 & -0,2837 & 3,886 & -3,907 & 1,192 & -2,169 & -2,032 \\ -0,7442 & -1,092 & -13,83 & 6,270 & -3,598 & 2,560 & -39,89 \\ 0,3868 & -0,3624 & -2,666 & 3,304 & -1,203 & 2,904 & 1,704 \\ 4,930 & 10,07 & -39,20 & -5,905 & 12,21 & 19,42 & 46,82 \\ 0,5945 & 0,9568 & 9,533 & -5,646 & 2,504 & -1,339 & 31,11 \end{bmatrix}$$

$$b_j = \begin{bmatrix} -6,023 \\ 79,45 \\ -0,2702 \\ 0,3027 \\ 40,90 \\ 8,210 \\ 5,133 \\ -1,422 \\ -2,114 \\ -2,148 \\ -45,64 \\ 3,620 \\ 26,08 \\ 34,15 \end{bmatrix}$$

$$v_{jk} = [0,0909 \quad 0,1318 \quad -24,21 \quad -19,75 \quad 11,40 \quad -0,0716 \quad 0,0064 \dots \\ \dots 0,1031 \quad -4,820 \quad 4,323 \quad 6,630 \quad -0,1748 \quad 5,018 \quad -4,981]$$

$$b_k = [-5,563]$$

#### 4.3.6. Modelo Neural 5 e Modelo Neural 6

A evolução do Modelo Neural 3 para o Modelo Neural 4 envolveu o acréscimo de duas variáveis ao mesmo tempo. Para tirar a prova de que realmente as duas variáveis acrescentadas eram importantes, foram realizados mais dois treinamentos da RNA.

No Modelo Neural 5 não foi utilizada a temperatura do meio precipitante no instante  $k$  (Figura 4.37) e no Modelo Neural 6, foi suprimida a variável  $\Delta MV_k$  (Figura 4.41).

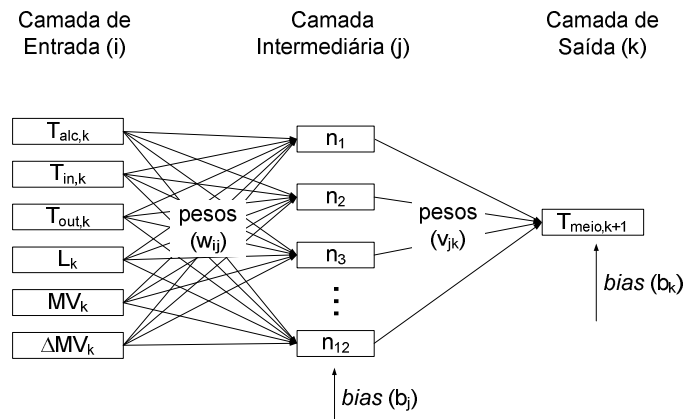


Figura 4.37: Estrutura da RNA para o Modelo Neural 5.

Para o Modelo Neural 5, o treinamento apresentou os resultados mostrados na Figura 4.38 e na Figura 4.39.



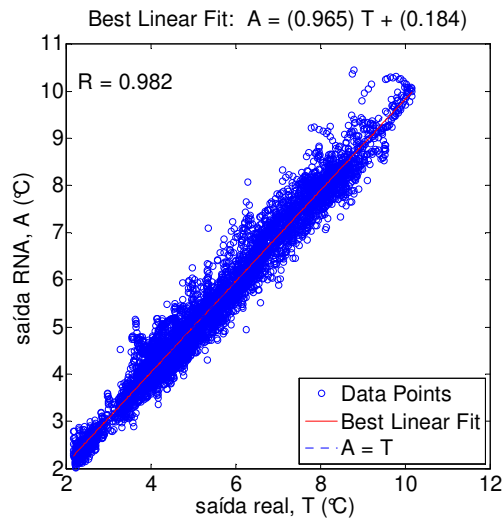


Figura 4.38: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 5.

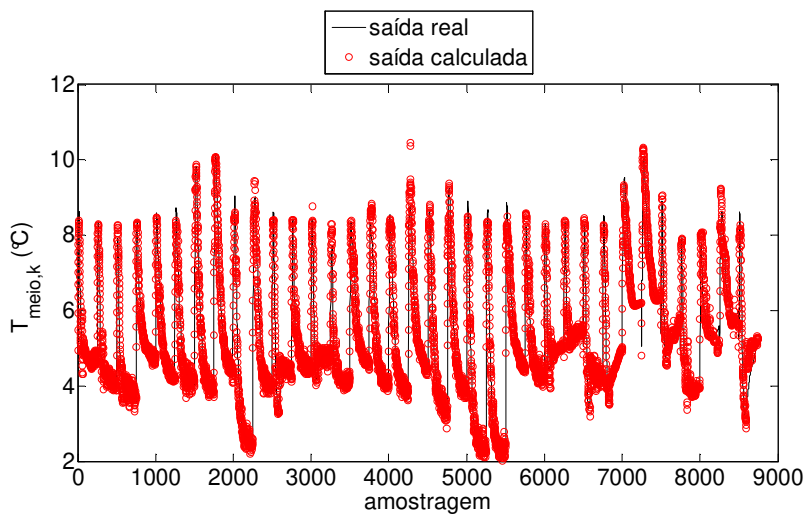
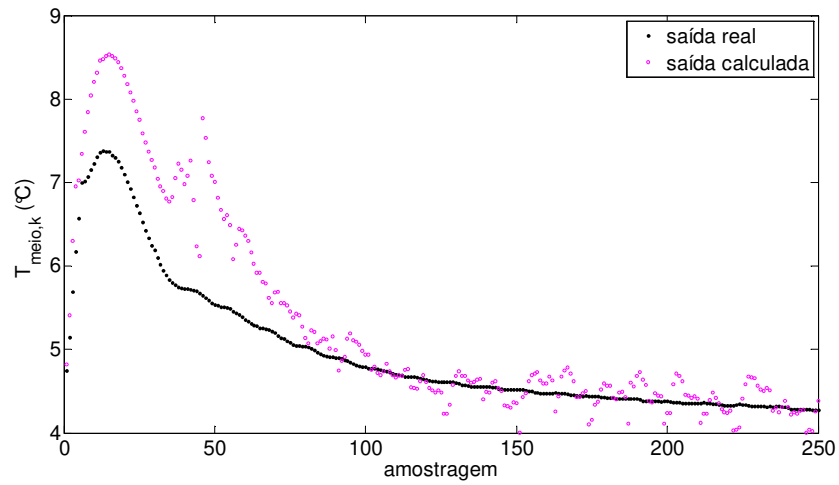
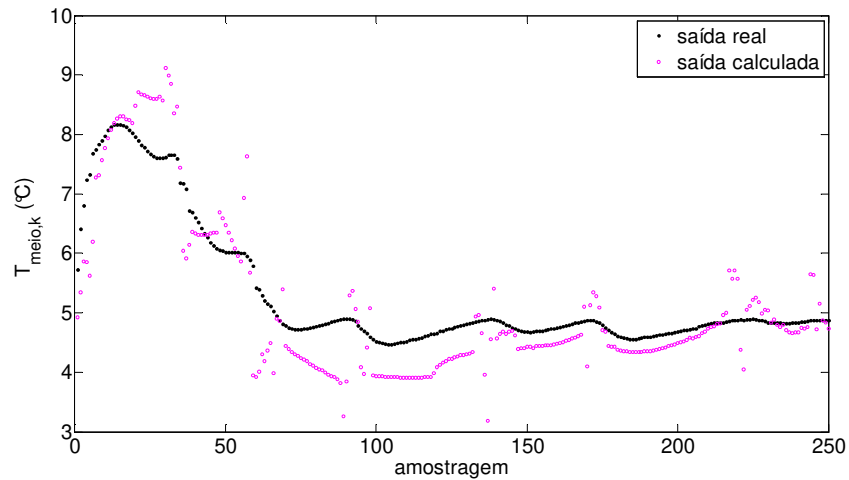


Figura 4.39: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 5.

De imediato, é possível concluir, comparando os gráficos da Figura 4.31 e da Figura 4.38, que o desempenho da rede neural que não contém a temperatura do meio como variável de entrada foi muito inferior, pois a rede perde a referência para prever esta mesma variável no instante de amostragem seguinte. Os testes para este modelo confirmam esta hipótese, como pode ser observado na Figura 4.40:



(a)



(b)

Figura 4.40: Comparação entre a saída real e a calculada pela RNA para um ensaio em malha aberta (a) e para um ensaio em malha fechada (b), no teste do Modelo Neural 5.

O treinamento do Modelo Neural 6 apresentou os resultados mostrados na Figura 4.42 e na Figura 4.43.

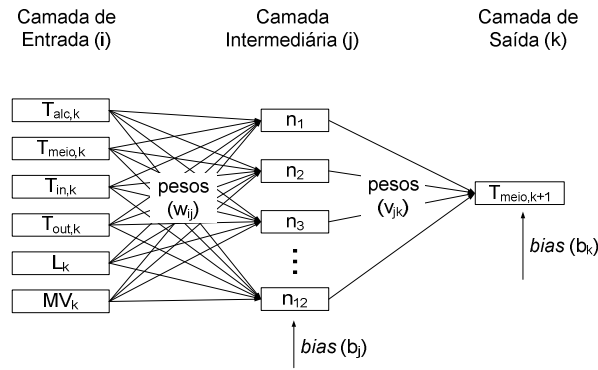


Figura 4.41: Estrutura da RNA para o Modelo Neural 6.

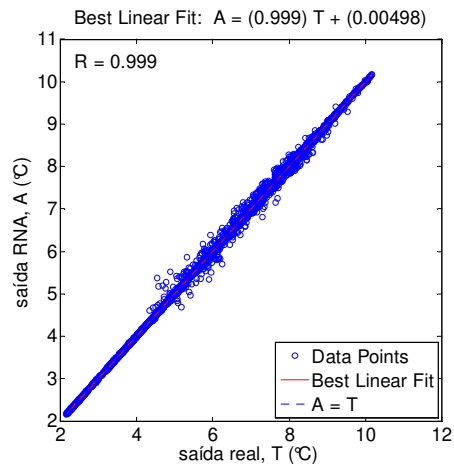


Figura 4.42: Dispersão entre a saída real e a calculada durante o treinamento da RNA, para o Modelo Neural 6.

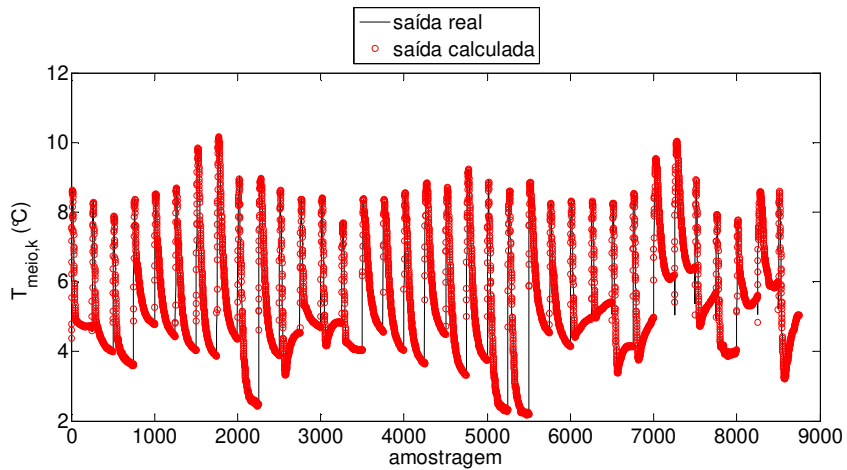
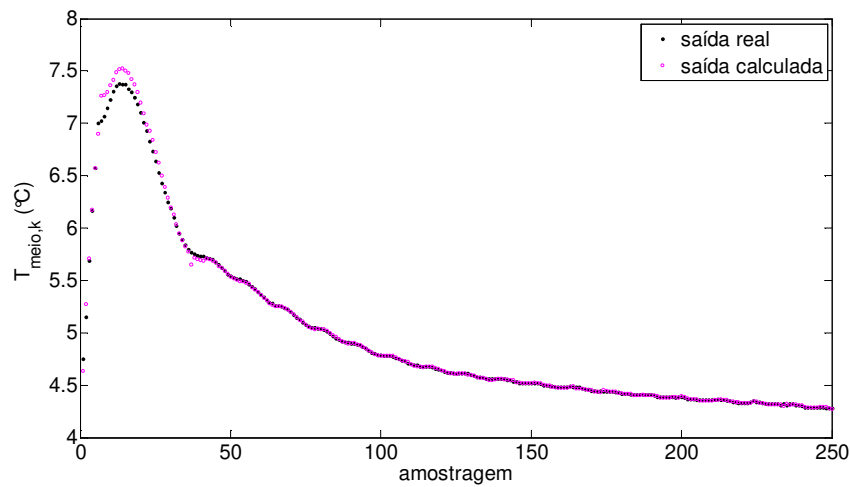


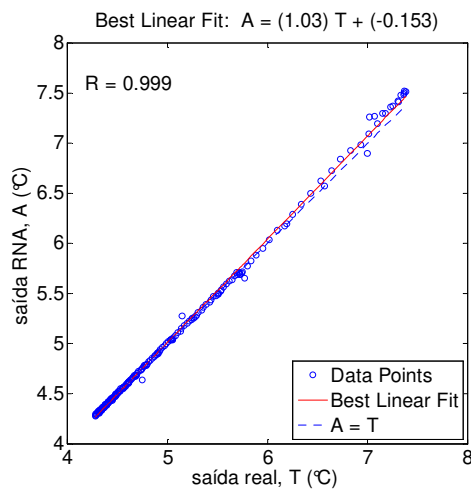
Figura 4.43: Comparação entre a saída real e a saída calculada pela rede para todos os ensaios no treinamento, para o Modelo Neural 6.

Neste treinamento, observou-se que o resultado foi bem próximo ao obtido para o treinamento do Modelo Neural 4b, colocando em dúvida a necessidade da variável  $\Delta MV_k$  na camada de entrada da rede neural.

Os resultados para teste deste modelo são apresentados na Figura 4.44 e na Figura 4.45.

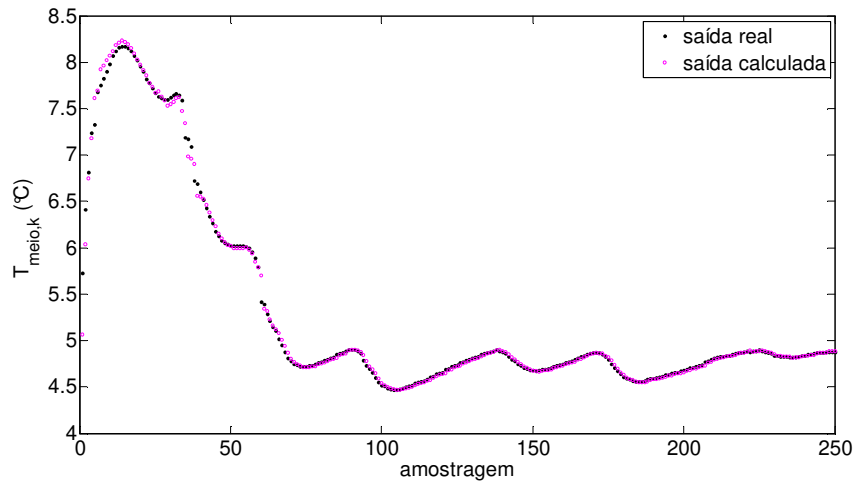


(a)

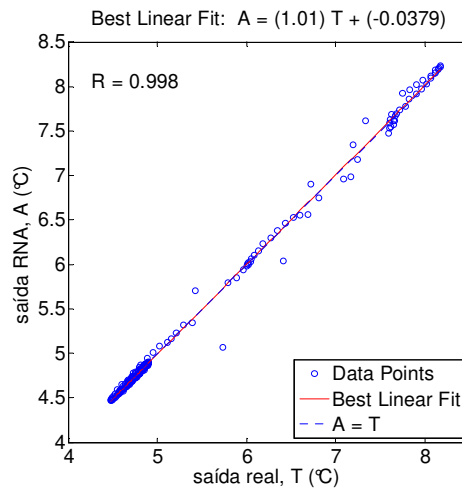


(b)

Figura 4.44: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha aberta, no teste do Modelo Neural 6.



(a)



(b)

Figura 4.45: Comparação entre a saída real e a calculada pela RNA (a) e a respectiva dispersão (b) para um ensaio em malha fechada, no teste do Modelo Neural 6.

Para a comparação entre o Modelo Neural 4b e o Modelo Neural 6, foram realizados os testes com os ensaios não utilizados no treinamento para cada um dos dois modelos e os parâmetros observados foram os coeficientes angular e linear da reta de ajuste dos pontos experimentais nos gráficos de dispersão entre a saída real e a calculada e os coeficientes de correlação (R). A Tabela 4.2 contém os valores destes parâmetros avaliados.

Tabela 4.2: Comparação entre o Modelo Neural 4b e o Modelo Neural 6.

Ensaio	R		Coeficiente angular		Coeficiente Linear	
	Modelo Neural 4b	Modelo Neural 6	Modelo Neural 4b	Modelo Neural 6	Modelo Neural 4b	Modelo Neural 6
1	0,997	0,999	0,992	0,992	0,047	0,037
2*	0,999	0,999	1,00	0,999	-0,001	0,002
3	1,00	1,00	1,00	1,00	-0,000	-0,018
4	0,999	0,998	1,00	1,00	-0,019	-0,017
5	0,999	0,999	0,996	0,995	0,028	0,031
6	1,00	1,00	1,00	0,998	-0,001	0,014
7	0,999	0,999	1,02	1,03	-0,109	-0,153
8	0,999	0,998	1,00	0,994	-0,008	0,030
9	0,999	0,999	1,00	1,00	-0,004	0,001
10	0,996	0,998	1,01	1,01	-0,070	-0,055
11	0,998	0,998	0,996	0,996	0,024	0,021
12	0,997	0,997	1,01	1,01	-0,050	-0,078
<i>Fuzzy 1</i>	0,987	0,998	0,984	1,01	0,068	-0,038
<i>Fuzzy 2</i>	0,997	0,995	1,02	1,02	-0,124	-0,131

\*as células sombreadas indicam os ensaios em que o Modelo Neural 4b apresentou os parâmetros analisados com os valores mais próximos de 1 para o coeficiente de correlação e para o coeficiente angular e mais próximos de zero para o coeficiente linear.

Dos 14 ensaios analisados, 8 apresentaram melhores resultados com o Modelo Neural 4b, tendo-se escolhido, portanto, este modelo para o desenvolvimento do controlador.

#### 4.4. Ensaio de Controle

Com o modelo neural definido, prosseguiu-se com a etapa de configuração do otimizador *Solver* para o controle da planta.

A função-objetivo empregada foi a minimização do erro quadrado (Equação 4.1).

$$\min_{MV} (PV_{(k+1)} - PV_{sp})^2 \quad \text{Equação 4.1}$$

A programação em VBA no Microsoft Excel foi feita de tal modo que a cada novo vetor escrito no Excel, o *Solver* era acionado para calcular um novo valor de rotação da bomba de propilenoglicol, a ser implementado no instante seguinte de amostragem.

O método numérico utilizado foi o método do gradiente descendente generalizado (*quasi-Newton*), disponível no próprio *software*.

A adição de restrições foi feita conforme os ensaios iam sendo realizados. De início, duas restrições foram inseridas, tendo em vista a própria limitação física da bomba: a faixa de soluções da variável  $MV_{k+1}$  deveria ficar entre 0 e 100 (% de rotação da bomba), como mostrado na Equação 4.2:

$$0 \leq MV \leq 100 \quad \text{Equação 4.2}$$

No *Solver*, é necessário separar a Equação 4.2 em duas partes, daí a referência a esta equação como “duas restrições”. O resultado obtido apenas com estas duas restrições foi o apresentado na Figura 4.46:

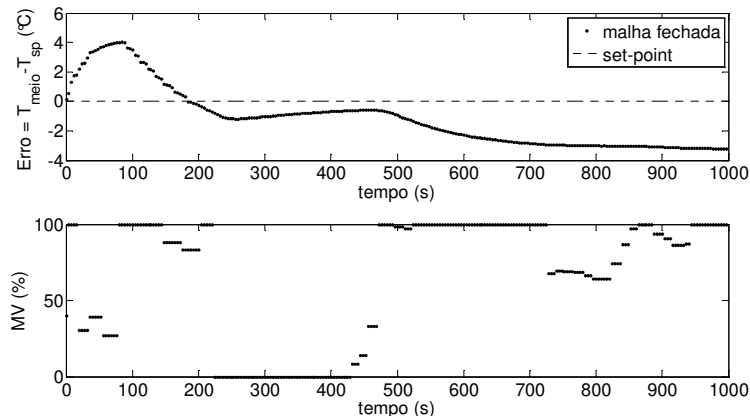


Figura 4.46: Ensaio de controle utilizando a restrição:  $MV_{k+1} = [0;100]\%$ .

Como a atuação da bomba apresentou-se com oscilações muito bruscas, acrescentou-se uma nova restrição, para suavizar esta ação:

$$|MV_{solver} - MV_{k-1}| \leq 20$$

Equação 4.3

A Equação 4.3 nada mais é do que uma restrição do tamanho do degrau da bomba. Um valor escolhido aleatoriamente como tentativa inicial foi de 20%, tendo-se obtido o resultado mostrado na Figura 4.47:

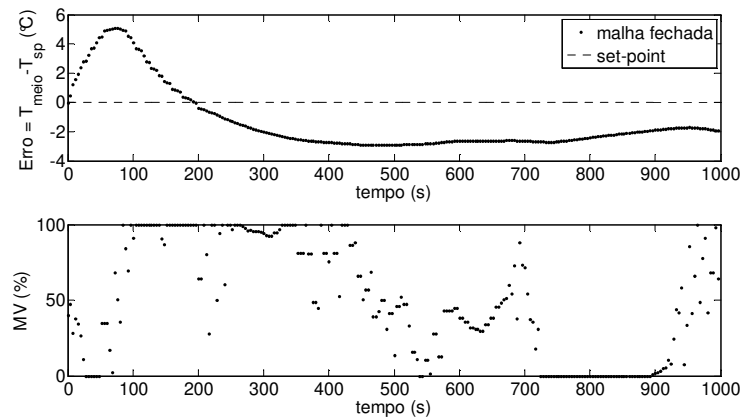


Figura 4.47: Ensaio de controle utilizando as restrições:  $MV_{k+1} = [0;100]\%$  e  $\Delta MV_k = \pm 20\%$ .

Mesmo quando a temperatura do meio precipitante baixou para menos de 5,0°C, a bomba continuou com uma atuação de 100%. Acrescentou-se, então, no Indusoft, uma condição para que a bomba desligasse quando a temperatura fosse menor que um determinado valor, evitando assim que as variáveis atingissem valores fora da faixa de treinamento e desta forma prejudicasse a capacidade de predição da rede neural. Este valor foi inicialmente testado como sendo de 4,0°C, tendo-se obtido o resultado mostrado na Figura 4.48:



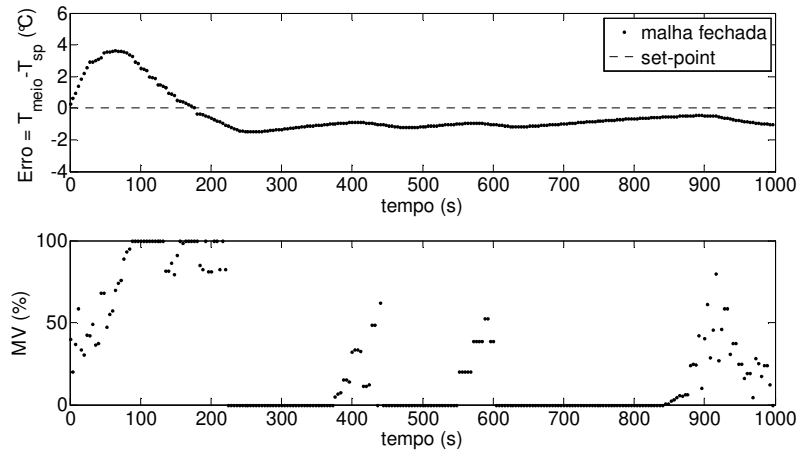


Figura 4.48: Ensaio de controle com as restrições:  $MV_{k+1} = [0;100]\%$  ;  $\Delta MV_k = \pm 20\%$   
e  $T_{k+1} = 4,0^\circ\text{C} \Rightarrow MV_{k+1} = 0$ .

Com a limitação de  $4,0^\circ\text{C}$ , o erro manteve-se em torno de  $-1,0^\circ\text{C}$ , e assim, como o *set-point* desejado era de  $5,0^\circ\text{C}$ , aumentou-se o valor da temperatura para  $4,9^\circ\text{C}$  e o resultado obtido foi o da Figura 4.49:

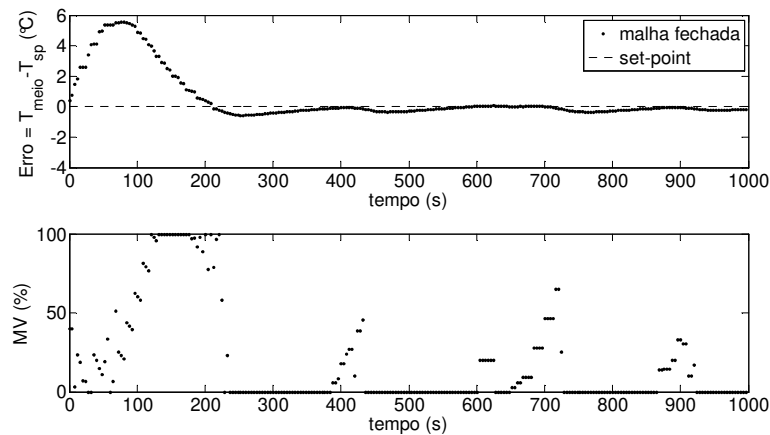


Figura 4.49: Ensaio de controle com as restrições:  $MV_{k+1} = [0;100]\%$  ;  $\Delta MV_k = \pm 20\%$   
e  $T_{k+1} = 4,9^\circ\text{C} \Rightarrow MV_{k+1} = 0$ .

Na Figura 4.49, o *overshoot* obtido foi de  $5,6^\circ\text{C}$ , indicando que o controlador precisava ter uma ação mais rápida sobre o processo.

Percebeu-se então que o valor do degrau poderia ser restringido baseando-se no valor do ganho do controlador PID (os resultados da sintonia do controlador PID são apresentados posteriormente, no item 4.5. ), que é de  $35\%/^{\circ}\text{C}$ , pois o ganho do controlador PID é análogo a esta variável no controlador utilizando o otimizador, dado que ambas as variáveis estão relacionadas à velocidade de atuação do controlador. Assim, obteve-se o seguinte resultado:

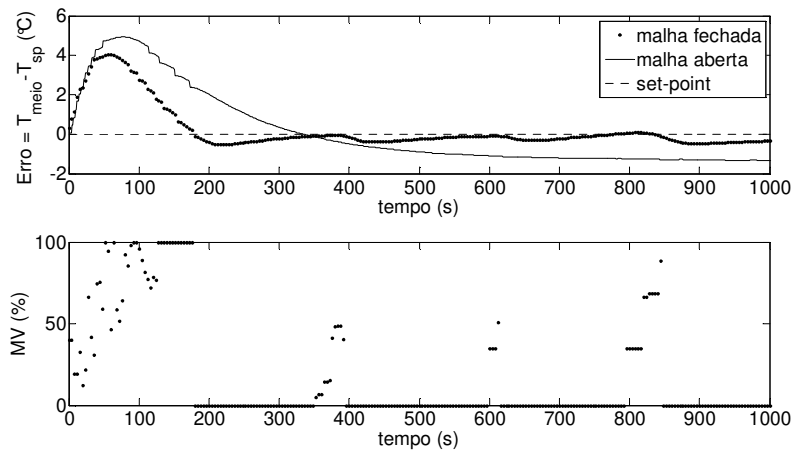


Figura 4.50: Ensaio de controle com as restrições:  $MV_{k+1} = [0;100]\%$  ;  $\Delta MV_k = \pm 35\%$  e  $T_{k+1} = 4,9^{\circ}\text{C} \Rightarrow MV_{k+1} = 0$ .

Com este ensaio, obteve-se um *overshoot* de  $4,1^{\circ}\text{C}$ , um tempo de subida de 180 segundos e não se observou a presença de *off-set*. No que tange a qualidade da enzima, é interessante que se tenha um *overshoot* abaixo de  $5^{\circ}\text{C}$  (isto é, temperatura do meio precipitante de  $10^{\circ}\text{C}$ ), pois acima deste valor, a enzima passa a sofrer desnaturação irreversível (SANTOS, 2006). Esta temperatura refere-se à desnaturação da enzima por ação do agente precipitante, o álcool, e não pela ação da temperatura, tendo em vista que a desnaturação da bromelina pelo calor ocorre em torno de  $70^{\circ}\text{C}$  (CESAR, 1999).

O ensaio em malha aberta apresentou *overshoot* de  $4,9^{\circ}\text{C}$  e tempo de subida de 334 segundos, reafirmando a necessidade de controle deste processo.

#### 4.5. Sintonia do Controlador PID

Para a realização da sintonia do controlador PID, foi utilizado o método de Ziegler-Nichols 1. Para aplicação deste método, realizou-se um ensaio em malha aberta, com uma perturbação em degrau na variável manipulada com observação do comportamento da variável controlada, com a finalidade de se obter os parâmetros do processo.

A curva obtida foi ajustada segundo o modelo de 1ª ordem com tempo morto e o resultado desta etapa é apresentado na Figura 4.51:

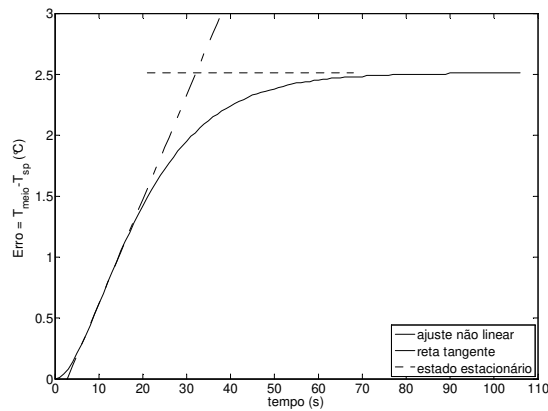


Figura 4.51: Curva de resposta da variável controlada frente a uma perturbação em degrau na variável manipulada.

Assim, os parâmetros encontrados foram:

$$K_p = 0,08^\circ\text{C}/\%$$

$$\tau = 29s$$

$$t_d = 3s$$

Substituindo estes valores devidamente na Equação 3.7 à Equação 3.9, do Método de Ziegler-Nichols, foram obtidos os parâmetros do controlador:

$$K_c = 139\%/^{\circ}C$$

$$\tau_I = 6s$$

$$\tau_D = 1,5s$$

Foi feita também a análise de estabilidade do processo utilizando o método *Root Locus*, conforme mostrado na Figura 4.52:

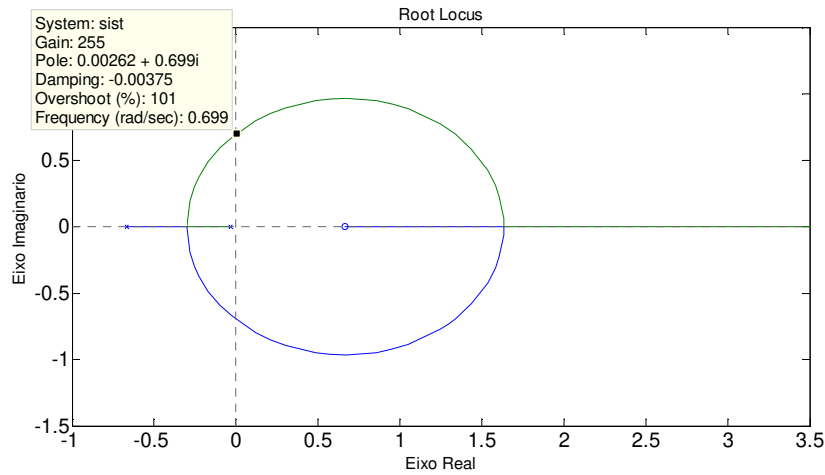


Figura 4.52: Análise de estabilidade - método do *Root Locus*.

Conforme a análise de estabilidade, observa-se que o valor encontrado para o  $K_c$  crítico ( $255\%/^{\circ}C$ ) é maior que o valor de  $K_c$  calculado pelo método de Ziegler-Nichols ( $139\%/^{\circ}C$ ), indicando que o processo ainda estaria na região de estabilidade. No entanto, considerando que o método não foi desenvolvido para sistemas operando em batelada alimentada, o valor de  $K_c$  de  $139\%/^{\circ}C$  foi considerado muito alto. Temendo-se que o processo fosse levado à instabilidade durante a sintonia fina, optou-se por utilizar inicialmente um valor correspondente a cerca de 5% deste valor, mantendo-se os outros parâmetros com o valor encontrado pelo método.

Assim, apresenta-se na Figura 4.53 o ensaio realizado utilizando os parâmetros  $K_c = 7\%/^{\circ}C$ ,  $\tau_I = 6s$  e  $\tau_D = 1,5s$ :

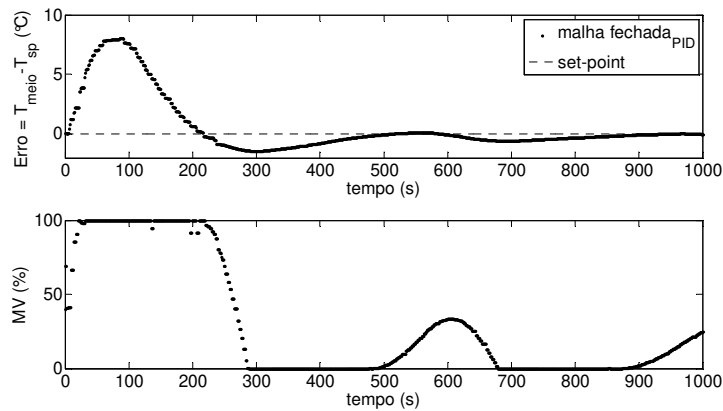


Figura 4.53: Ensaio de controle – sintonia fina do PID, utilizando os parâmetros:  
 $K_c = 7\%/^{\circ}C$ ,  $\tau_i = 6s$  e  $\tau_D = 1,5s$ .

O *overshoot* obtido foi de  $8,0^{\circ}C$ , indicando que o controlador p precisaria ter uma atuação mais rápida, isto é, o parâmetro ganho deveria ser aumentado, obtendo-se um menor valor de *overshoot*. O valor do tempo de subida foi de 216 segundos.

Diversos ensaios foram realizados até se ajustar o valor do ganho do controlador, partindo-se então para a sintonia dos parâmetros  $\tau_i$  e  $\tau_D$ . O valor encontrado para o ganho foi de  $K_c = 35\%/^{\circ}C$ . O ensaio realizado utilizando os parâmetros  $K_c = 35\%/^{\circ}C$ ,  $\tau_i = 18s$  e  $\tau_D = 1,5s$  é apresentado na Figura 4.54:

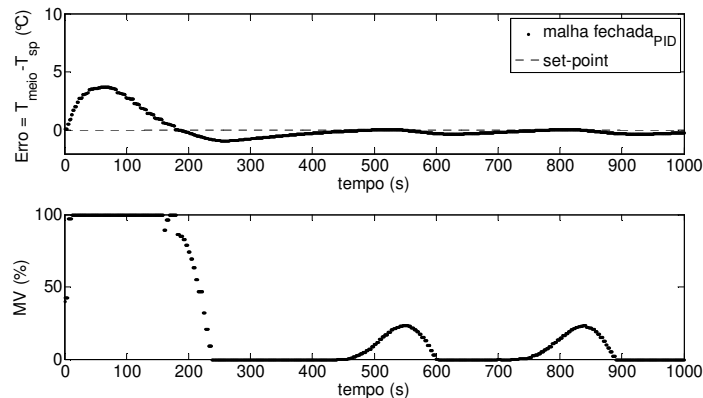


Figura 4.54: Ensaio de controle – sintonia fina do PID, utilizando os parâmetros:  
 $K_c = 35\%/^{\circ}C$ ,  $\tau_i = 18s$  e  $\tau_D = 1,5s$ .

O *overshoot* obtido foi de  $3,7^{\circ}\text{C}$ , com um tempo de subida de 188 s. Como uma tentativa de se melhorar a curva de resposta da variável controlada, aumentou-se o valor de  $\tau_I$  para 28s, com a expectativa de que, introduzindo mais oscilações na variável controlada, esta retornasse mais rapidamente para o *set-point*, reduzindo assim o tempo de estabilização. O resultado obtido para esta situação é apresentado na Figura 4.55.

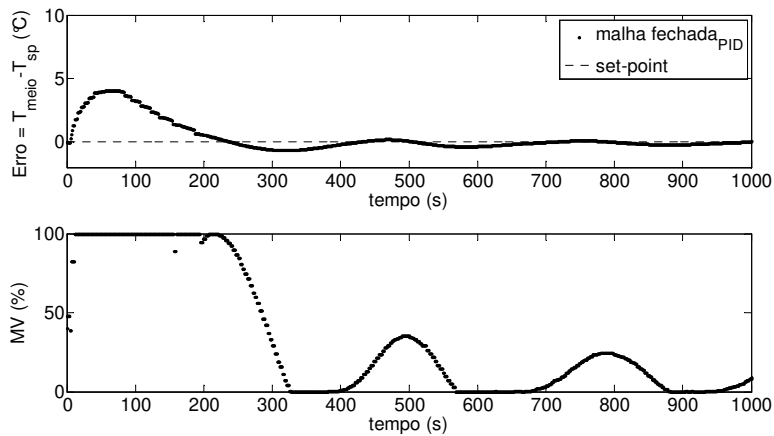


Figura 4.55: Ensaio de controle – sintonia fina do PID, utilizando os parâmetros:

$$K_c = 35\%/^{\circ}\text{C}, \tau_I = 28\text{s} \text{ e } \tau_D = 1,5\text{s}.$$

Observa-se que o *overshoot* aumentou para  $4,1^{\circ}\text{C}$  e o tempo de subida aumentou para 240 s. Apesar destes parâmetros terem apresentado valores piores que no ensaio anterior, a temperatura do meio precipitante voltou a atingir o *set-point* pela segunda vez num tempo menor (420 s) comparado ao ensaio anterior (490 s), fazendo a temperatura oscilar em torno do valor de referência mais rapidamente.

Assim, para melhorar o *overshoot*, aumentou-se desta vez o valor do parâmetro  $\tau_D$  para 7 s.

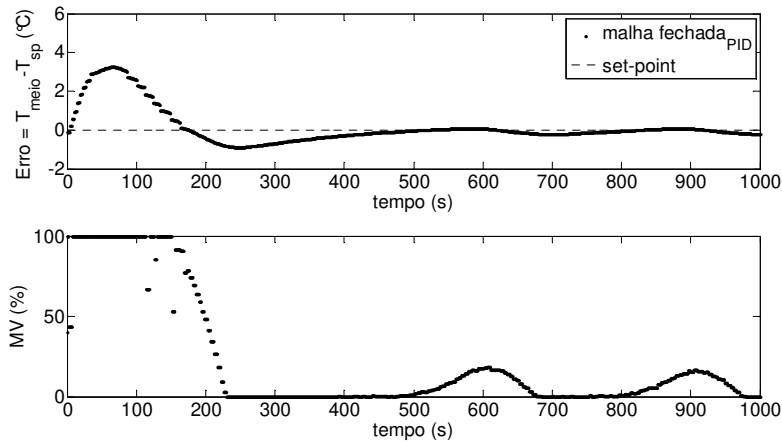


Figura 4.56: Ensaio de controle – sintonia fina do PID, utilizando os parâmetros:

$$K_c = 35\%/^{\circ}\text{C}, \tau_I = 28\text{s} \text{ e } \tau_D = 7\text{s}.$$

Na Figura 4.56, observa-se que o *overshoot* diminuiu para 3,2°C e as oscilações foram amortecidas pelo aumento na constante de tempo derivativa. Foi obtido um tempo de subida de 175 segundos.

Uma comparação dos ensaios apresentados na sintonia fina do controlador PID pode ser melhor observada na Tabela 4.3:

Tabela 4.3: Comparação entre os parâmetros do controlador PID na etapa de sintonia fina

Controlador PID	$K_c$ (%/°C)	$\tau_I$ (s)	$\tau_D$ (s)	<i>overshoot</i> (°C)	tempo de subida (s)	ITAE
1	7,0	6,0	1,5	8,0	216	245022,4
2	35	18	1,5	3,7	188	128815,6
3	35	28	1,5	4,1	240	124331,4
4	35	28	7,0	3,2	175	104553,2

Como se pode observar na Tabela 4.3, o controlador 4 ( $K_c = 35\%/^{\circ}\text{C}$ ,  $\tau_I = 28\text{s}$  e  $\tau_D = 7\text{s}$ ) foi o que apresentou menor valor de *overshoot*, menor tempo de subida e menor valor de ITAE, tendo sido portanto, escolhido como o

controlador PID sintonizado a ser utilizado como base de comparação com o controlador alternativo (RNA).

#### 4.6. Comparação entre os Controladores

Foi realizada uma comparação entre os controladores PID e alternativo e o ensaio em malha aberta (bomba fixa em 40% de rotação), como mostrado na Figura 4.57.

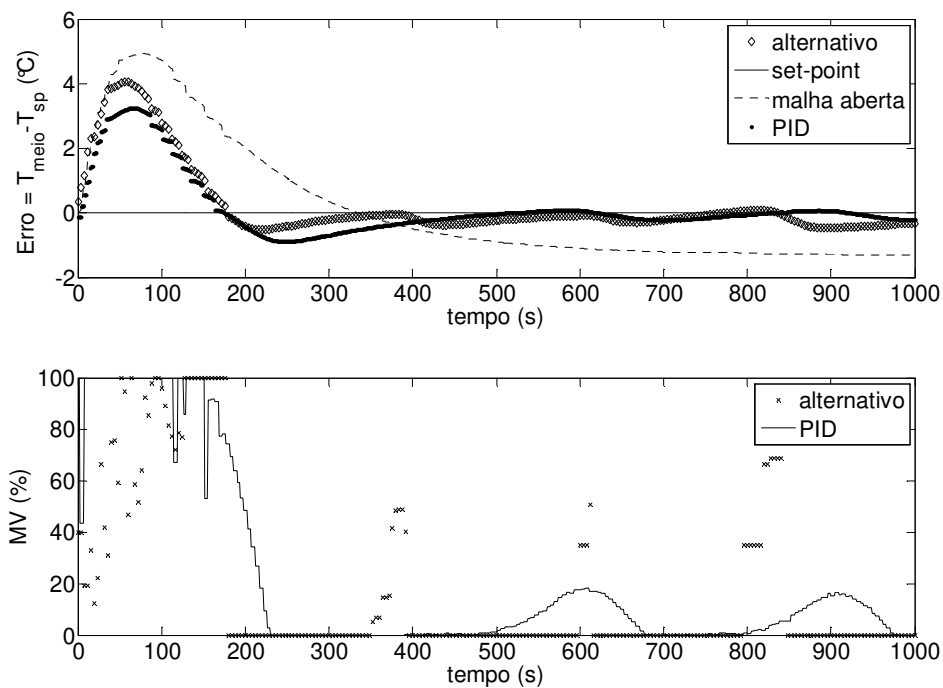


Figura 4.57: Comparação entre os controladores PID e alternativo e um ensaio em malha aberta.

Pela Figura 4.57, observa-se que a curva do ensaio em malha aberta, além de ter apresentado elevado valor de *overshoot* e de tempo de subida, não retornou para o *set-point*, indicando a necessidade de implementação de controle neste processo.

Também foi feita uma análise de parâmetros, conforme mostrado na Tabela 4.4.



Tabela 4.4: Comparação entre os parâmetros dos controladores PID e alternativo e do ensaio em malha aberta

<b>Controlador</b>	<b><i>Overshoot</i> (°C)</b>	<b>tempo de subida (s)</b>	<b>ITAE</b>	<b>Energia consumida (kWh)</b>
PID	3,2	175	104553,2	0,040
Alternativo	4,1	180	149081,2	0,038
Malha aberta	4,9	334	597343,5	0,060

O menor *overshoot* obtido foi o do controlador PID, com uma diferença de 0,8°C em relação ao controlador alternativo.

Para o parâmetro tempo de subida, quase não houve diferença entre os dois controladores, tendo-se notado uma grande melhoria quando da comparação com o ensaio em malha aberta.

O valor do parâmetro ITAE foi menor para o controlador PID, tendo sido 30% menor que o ITAE do controlador alternativo.

No caso da energia consumida, pode-se notar a grande vantagem de se realizar este processo de precipitação em malha fechada, pois com o controlador PID houve uma redução de 33% no consumo e com o controlador alternativo, uma redução de 37%.

A comparação de cada parâmetro pode ser visualizada nos gráficos da Figura 4.58.

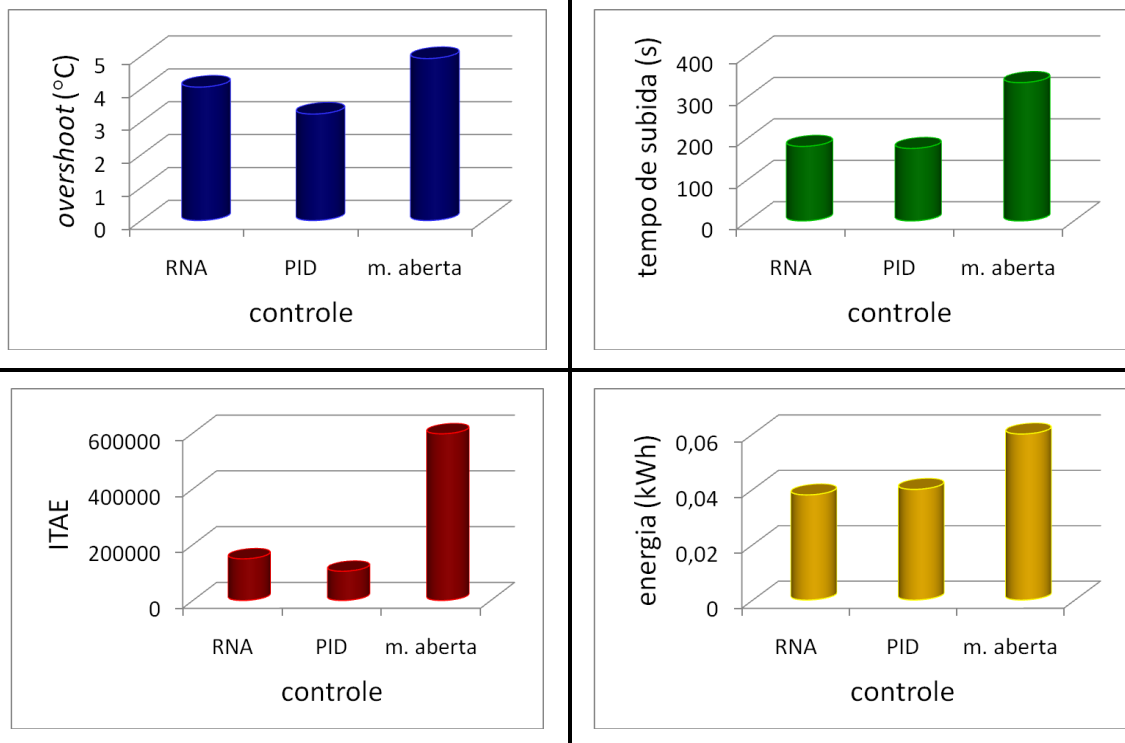


Figura 4.58: Comparação entre os controladores PID e alternativo e o ensaio em malha aberta

Novamente, através de uma análise visual, observa-se que não houve diferença significativa entre o desempenho dos controladores PID e alternativo, podendo-se notar a grande vantagem destes sobre o ensaio em malha aberta.

## CAPÍTULO 5: CONCLUSÕES

Este capítulo, com as conclusões obtidas ao longo do trabalho, foi dividido em quatro seções: a primeira diz respeito à modelagem por redes neurais artificiais, a segunda traz comentários sobre o uso do Microsoft Excel, a terceira sobre a comparação entre o controlador desenvolvido e o convencional e a última aborda um comentário sobre a enzima bromelina.

No presente trabalho propôs-se o desenvolvimento e implementação de um controlador alternativo para um processo de precipitação de bromelina, utilizando modelagem por redes neurais artificiais em conjunto com o *Solver* do Microsoft Excel. O modelo neural foi capaz de prever a variável controlada com exatidão e foi implementado com sucesso em conjunto com o *Solver* para manter a temperatura do meio precipitante em torno do *set-point*, com baixo consumo de energia. O sistema de controle alternativo desenvolvido reuniu os benefícios das redes neurais artificiais em modelar relações não lineares e a acessibilidade da planilha eletrônica, fazendo desta metodologia uma maneira promissora de se lidar com problemas de controle de processos, sem despende esforços desnecessariamente em modelagem matemática. A expansão da gama de aplicações deste método a outros processos é um ponto relevante para pesquisas científicas, uma vez que diferentes campos podem aplicar esta solução a muitos de seus problemas.

### 5.1. Redes Neurais Artificiais

Com este trabalho, foi possível atestar o uso de tecnologias avançadas de Inteligência Artificial como as Redes Neurais Artificiais como uma importante ferramenta de modelagem de processos não lineares.

Especificamente para o processo estudado, conclui-se que a escolha adequada das variáveis de entrada da rede neural é imprescindível para uma

modelagem exata. Todas as variáveis que apresentem influência sobre o processo devem ser incluídas na camada de entrada do modelo neural. Assim, para o processo de precipitação de bromelina, tem-se que:

- a variável temperatura do etanol,  $T_{alc}$ , apresenta grande influência no *overshoot* obtido na curva da variável controlada; é possível diminuir o *overshoot* reduzindo a temperatura do álcool. Mesmo com a RNA treinada para atuar em faixas de  $T_{alc} > 23^{\circ}C$ , o *overshoot* continuará alto, pois esta é uma característica inerente ao processo e a atuação máxima da bomba está restrita a 100%. O treinamento da RNA para  $T_{alc} > 23^{\circ}C$  servirá apenas para que a rede não faça previsões erradas nesta região e não para conter a elevação da temperatura do meio precipitante. Daí surge uma sugestão: colocar a temperatura do álcool na malha de controle. Aqui caberia uma discussão sobre melhoria do processo e não do controlador, o que foge à proposta deste trabalho. Leite (2007) utilizou em seus experimentos o valor de  $21^{\circ}C$  para o etanol, valor este que poderia ser utilizado como *set-point* na nova malha de controle proposta, dado que os resultados obtidos para esta condição não comprometem a qualidade da enzima obtida.

- o nível do tanque mostrou-se como uma variável indispensável para distinguir vetores de entrada iguais para diferentes valores de saída; sem a variável nível a rede não foi capaz de realizar previsões corretas. A importância desta variável como entrada da rede também pode ser facilmente entendida quando se observa que, uma vez que o processo é realizado em batelada, tem-se uma variação na área de troca térmica ao longo do ensaio e uma variação também no efeito do calor de dissolução do álcool.

- o degrau aplicado na variável manipulada e a temperatura do meio precipitante também são variáveis importantes para a rede neural, pois servem como referência para a previsão da temperatura do meio no instante seguinte de amostragem.

Foi possível comprovar que os dados de ensaios em malha aberta foram suficientes para se realizar o treinamento da rede neural, pois a dinâmica do

processo foi ensinada à rede a partir do fornecimento de dados do comportamento da variável controlada frente a perturbações em degrau na variável manipulada.

Na etapa de validação *online*, ficou claro que um banco de dados representativo do processo tem grande valor para que o modelo neural seja adequado, possibilitando a posterior implementação de uma estratégia de controle.

A implementação da estratégia de controle utilizando o *Solver* do Excel foi bem sucedida, tendo sido possível manter a variável controlada em torno do *set-point*, com pequeno valor de tempo de subida e *overshoot* dentro da faixa permitida para manter a atividade enzimática da bromelina. Note-se que o controlador desenvolvido ainda não está sintonizado.

Um outro ponto relevante é o reconhecimento da limitação desta ferramenta com relação à capacidade de extrapolação. Sabe-se da literatura que o uso de Redes Neurais é mais indicado para interpolações. Por mais que a rede tenha sido treinada com uma ampla faixa de dados do processo, qualquer processo está sujeito a sofrer distúrbios imprevisíveis, o que poderia causar erros de previsão do modelo neural, prejudicando o controle do processo. Sendo assim, a aplicação desta técnica fica sujeita a um chaveamento comandado por uma condição de atuação somente se as condições estiverem dentro do domínio conhecido pela rede.

## **5.2. Uso da planilha Excel**

A utilização da planilha eletrônica do Excel tornou o desenvolvimento do trabalho mais dinâmico e fácil, devido à familiaridade com o *software*. Esta pode ser apontada como uma das principais vantagens para sua utilização, além da rapidez de cálculos observada.

### 5.3. Comparação entre os Controladores

O desempenho do controlador alternativo não foi superior ao do controlador PID, tendo apresentado apenas o parâmetro de consumo energético como mais vantajoso. No entanto, retoma-se aqui o objetivo deste trabalho, que foi o de desenvolver e implementar um controlador alternativo que utilizasse o conceito de redes neurais artificiais, em conjunto com as facilidades oferecidas pelo Excel. Novamente reforça-se que a etapa de sintonia do controlador alternativo não foi realizada; assim, cria-se a expectativa de que se forem encontrados valores ótimos para as restrições do *Solver* o controlador alternativo poderá apresentar melhor desempenho que o convencional PID, que não possui propriedade adaptativa e nem é adequado para processos que possuem relações não lineares entre as variáveis de entrada e de saída.

Outra questão que merece atenção quando da comparação entre os controladores é o esforço dedicado à sintonia de cada um dos controladores. Foi dedicado um tempo muito maior à sintonia do controlador PID do que à sintonia do controlador alternativo. Assim, este também é um fator que deve ser levado em consideração no julgamento do desempenho dos controladores.

### 5.4. Bromelina

Em seu trabalho, Leite (2007) demonstrou que o desempenho do controlador avançado utilizando lógica *fuzzy* foi bem semelhante ao do controlador convencional, quando da análise dos parâmetros ITAE, *overshoot* e tempo de subida. No entanto, ao comparar a atividade enzimática da bromelina obtida com cada um dos controladores, observou-se que o controlador avançado apresentou significativa vantagem. Assim, para um estudo mais completo, poderia ser realizada a análise de atividade enzimática para o controlador alternativo desenvolvido no presente trabalho. Esta etapa não foi realizada pois estaria-se fugindo da proposta inicial, com foco no desenvolvimento do controlador.

## CAPÍTULO 6: SUGESTÕES PARA TRABALHOS FUTUROS

Cada dissertação ou tese apresentada não esgota o tema estudado e em geral constitui apenas uma pequena contribuição num estudo mais amplo. No caso deste trabalho, o estudo de controladores para a recuperação desta enzima foi iniciado no ano de 2006, por Santos, com o controlador PID Adaptativo. Leite (2007) prosseguiu com a pesquisa, introduzindo o controlador inteligente *fuzzy* no leque de estudos do processo em questão. No Laboratório de Controle e Automação de Processos da FEQ/UNICAMP, outros trabalhos com controladores avançados ainda estão sendo desenvolvidos, com aplicação de um sistema neuro-*fuzzy* e aplicação de diferentes estruturas *fuzzy*.

Desta forma, para dar prosseguimento à pesquisa científica, seguem algumas sugestões para trabalhos futuros:

- Utilizar outras funções de ativação para treinamento da rede neural, verificando a melhor função para este processo;
- Estudar a melhor estrutura da rede neural quanto ao número de neurônios na camada intermediária, poupando esforço computacional;
- Inserir na malha de controle a temperatura do álcool, visando mantê-la constante para evitar maiores interferências na precipitação;
- Encontrar condições ótimas para aplicação do controlador neural no processo em estudo (sintonia), isto é, testar outros valores para as restrições do *Solver*, variar os parâmetros internos do *Solver*, encontrados nas opções desta

ferramenta e até mesmo propondo outras restrições, de maneira a encontrar os valores mais adequados para o melhor controle possível;

- Realizar a análise de atividade enzimática da bromelina obtida com o controlador desenvolvido e comparar com a atividade da bromelina obtida com o controlador convencional; para isto, propõe-se o uso do método colorimétrico de biureto, que se baseia na interação entre sulfato de cobre e a proteína. O complexo resultante absorve em 540nm (ZAIA, ZAIA e LICHTIG, 1998);

- Instalar definitivamente um sistema de destilação no Laboratório de Controle e Automação de Processos para reciclar o álcool utilizado nos experimentos, tendo em vista uma preocupação com a questão ambiental;

- Realizar um estudo de atividade enzimática com o álcool destilado, para verificar a viabilidade de utilizá-lo como agente precipitante nos ensaios, reduzindo custos de compra de álcool absoluto;

- Desenvolver e implementar um sistema de controle baseado em sistemas *neuro-fuzzy* para o processo de precipitação de bromelina;

- Desenvolver e implementar um sistema de controle baseado em lógica *fuzzy*, utilizando modelagem de Takagi-Sugeno.

Com as propostas sugeridas, espera-se que outros pesquisadores tenham a oportunidade de desenvolver projetos nesta área, contribuindo para a pesquisa científica.



## CAPÍTULO 7: REFERÊNCIAS BIBLIOGRÁFICAS

BERVEVELLO, LUIZ CARLOS. *Estudo do processo de Recuperação e Separação de Bromelina utilizando Sistema de Duas Fases Aquosas em Micro-Coluna de Extração*. Campinas: Faculdade de Engenharia Química, Universidade Estadual de Campinas, 2001. 105p. Tese (Doutorado).

CAMPOS, M. M., SAITO, K. *Sistemas inteligentes em controle e automação de processos*. Rio de Janeiro: Editora Ciência Moderna Ltda., 2004. Capítulo 4: Sistemas Inteligentes baseados em Redes Neurais, p. 137-203.

CESAR, ANA CLAUDIA WABISZCZEWICZ. *Análise de Viabilidade Econômica de um Processo de Extração e Purificação da Bromelina do Abacaxi*. Campinas: Faculdade de Engenharia Química, Universidade Estadual de Campinas, 2005. 99 p. Tese (Doutorado).

CESAR, A. C. W., SILVA, R., LUCARINI, A. C. Recuperação de enzimas proteolíticas presentes na casca e talo do abacaxi. RIC, 01, 47-54, São Carlos, 1999.

CUNHA, GETÚLIO AUGUSTO PINTO. Informativo Mensal da Equipe Técnica de Abacaxi. *Embrapa Mandioca e Fruticultura Tropical*, v. 5, número 1, janeiro a abril de 2007.

CYBENKO, G. Approximation by superposition of a sigmoidal function, *Mathematics of Control, Signals and Systems*, v. 2, p. 303-314, 1989.

DRAEGER, A., ENGELL, S., RANKE, H. Model predictive control using neural networks. *IEEE Control Systems*, v. 15, p. 61-66, 1995.

EDGAR, T. F., HIMMELBLAU, D. M., LASDON, L. S. *Optimization of Chemical Processes*. 2<sup>nd</sup> ed. New York: McGraw Hill, 2001, 651 p.

FAUSETT, L. *Fundamentals of Neural Networks - Architectures, Algorithms and Applications*. New Jersey: Prentice Hall, 1994, 461 p.

FERRAREZI, LUIZ FELIPE NOGUEIRA. *Avaliação experimental do desempenho de diferentes controladores fuzzy aplicados em um processo em batelada*. Campinas: Faculdade de Engenharia Química, Universidade Estadual de Campinas, 2008. 26 p. Exame de Qualificação (Mestrado).

FILETI, A. M. F., PACIANOTTO, T. A., CUNHA, A. P. Neural modeling helps the BOS process to achieve aimed end-point conditions in liquid steel. *Engineering Applications of Artificial Intelligence*, v. 19, p. 9-17, 2006.

FYLSTRA, D., LASDON, L., WATSON, J., WAREN, A. Design and Use of the Microsoft Excel Solver. *Interfaces* 28 (5), p. 29-55, September-October 1998.

GADKAR, K. G., MEHRA, S., GOMES, J. Online adaptation of neural networks for bioprocess control. *Computers and Chemical Engineering*, v.29, p. 1047-1057, January 2005.

GLASSEY, J., IGNOVA, M., WARD, A. C., MONTAGUE, G. A., MORRIS, A. J. Bioprocess supervision: neural networks and knowledge based systems. *Journal of Biotechnology*, v. 52, p. 201-205, 1997.

GRANADA, G. G., ZAMBIAZI, R. C., MENDONÇA, C. R. B. Abacaxi: Produção, Mercado e Subprodutos. *B. CEPPA*, v.22, n.2, p.405-422, jul/dez 2004.

GONZAGA, J. C. B., MELEIRO, L. A. C., KIANG, C., MACIEL FILHO, R. ANN-based soft-sensor for real-time process monitoring and control of the industrial polymerization process. *Computers and Chemical Engineering*, v. 33, p. 43-49, 2009.

HERNÁNDEZ, J. A., BASSAM, A., SIQUEIROS, J., JUÁREZ-ROMERO, D. Optimum operating conditions for a water purification process integrated to a heat transformer with energy recycling using neural networks inverse. *Renewable Energy*, v. 34, p. 1084-1091, 2009.

HUSSAIN, M. A. Review of the applications of neural networks in chemical process control – simulation and online implementation. *Artificial Intelligence in Engineering*, v. 13, p. 55-68, 1999.

KHALID, M., OMATU, S., Yusof, R. MIMO furnace control with neural networks. *IEEE Transactions on Control Systems Technology*, v. 1, p. 238-245, 1993.

LEITE, MANUELA SOUZA. *Estudo comparativo do desempenho de controladores fuzzy e convencional aplicados em um bioprocesso*. Campinas: Faculdade de Engenharia Química, Universidade Estadual de Campinas, 2007. 142 p. Dissertação (Mestrado).

LEITE, M. S., FUJIKI, T. L., SILVA, F. V., FILETI, A. M. F. Determinação de condições operacionais de um processo de precipitação da bromelina via planejamento experimental. In: XVII Congresso Brasileiro de Engenharia Química, setembro 2008, Recife.

LENNOX, B., MONTAGUE, G. A., FRITH, A. M., GENT, C., BEVAN, V. Industrial application of neural networks – an investigation. *Journal of Process Control*, v. 11, p. 497-507, 2001.

LOPES, FRANCISCO LUIZ GUMES; SEVERO JÚNIOR, JOÃO BAPTISTA; SANTANA, JOSÉ CARLOS CURVELO; SOUZA, ROBERTO RODRIGUES DE; TAMBOURGI, ELIAS BASILE. Utilização de membranas planas na concentração de enzimas bromelinas da polpa de abacaxi (*Ananas comosus* L.). *Revista Brasileira de Produtos Agroindustriais*, Campina Grande, v. 7, n. 1, p. 33-38, 2005.

MARTINS, F. G., COELHO, M. A. N. Application of feedforward artificial neural networks to improve process control of PID-based control algorithms. *Computers and Chemical Engineering*, v.24, p.853-858, 2000.

MATTOS, P. E. O. Validação Clínica da Suplementação de Bromelaína para Atletas, *Projeto de Pesquisa*, Instituto de Farmacologia e Biologia Molecular, UNIFESP, São Paulo, 2005.

MURACHI, T. Bromelain Enzymes. *Methods in Enzymology*, v.XLV, p.475-485, 1976.

NAGY, Z. K. Model based control of a yeast fermentation bioreactor using optimally designed artificial neural networks. *Chemical Engineering Journal*, v.127, p.95-109, October 2006.

PADMAVATHI, G., MANDAN, M. G., MITRA, S. P., CHAUDHURI, K. K. Neural modelling of Mooney viscosity of polybutadiene rubber. *Computers and Chemical Engineering*, v. 29, p. 1677-1685, 2005.

REZENDE, SOLANGE OLIVEIRA. *Sistemas Inteligentes: fundamentos e aplicações*. 1.ed. Barueri: Editora Manole Ltda, 2003. Capítulo 6: Redes Neurais Artificiais, p. 141-168.

SANTOS, REGINA LÚCIA DE ANDRADE DOS. *Controle e Monitoramento, em Tempo Real, de um Processo de Precipitação de Bromelina utilizando*

*Comunicação Digital Fieldbus*. Campinas: Faculdade de Engenharia Química, Universidade Estadual de Campinas, 2006. 86 p. Dissertação (Mestrado).

SCOPES, R. K. *Protein Purification: Principles and Practical*. New York: Springer Verlag, 1994, 329 p.

SEBORG, D. E., Experience with nonlinear control and identification strategies. In INTERNATIONAL CONFERENCE ON CONTROL, 1994, London, p. 879-886.

SEBORG, D. E., EDGAR, T. F., MELLICHAMP, D. A. *Process Dynamics and Control*. New York: John Wiley and Sons, Inc, 1989, 717 p.

Smar Equipamentos Industriais Ltda. Departamento de Engenharia de Aplicações. *Apostila: Como implementar projetos com fieldbus foundation*. Sertãozinho: 1998. v. 1, 236 p.

VAN CAN, H. J. L., BRAAKE, H. A. B. T., HELLINGA, C., KRIJGSMAN, A. J., VERBRUGGEN, H. B., LUYBEN, K. CH. A. M., HEIJNEN, J. J. Design and real time testing of a neural model predictive controller for a nonlinear system. *Chemical Engineering Science*, v. 50, p. 2419-2430, 1995.

ZAIA, D. A. M., ZAIA, C. T. B. V., LICHTIG, J. Determinação de Proteínas Totais via Espectrofotometria: Vantagens e Desvantagens dos Métodos Existentes. *Química Nova*, v.21, n.6, p.787-793, junho 1998.

## APÊNDICE A: CONFIGURAÇÃO DO DRIVER ODBC

O primeiro passo para a configuração do *driver* ODBC é a criação de uma nova fonte de dados no Windows. Para isto, os procedimentos seguintes devem ser executados:

1. Clicar em 'Iniciar' → 'Painel de Controle'

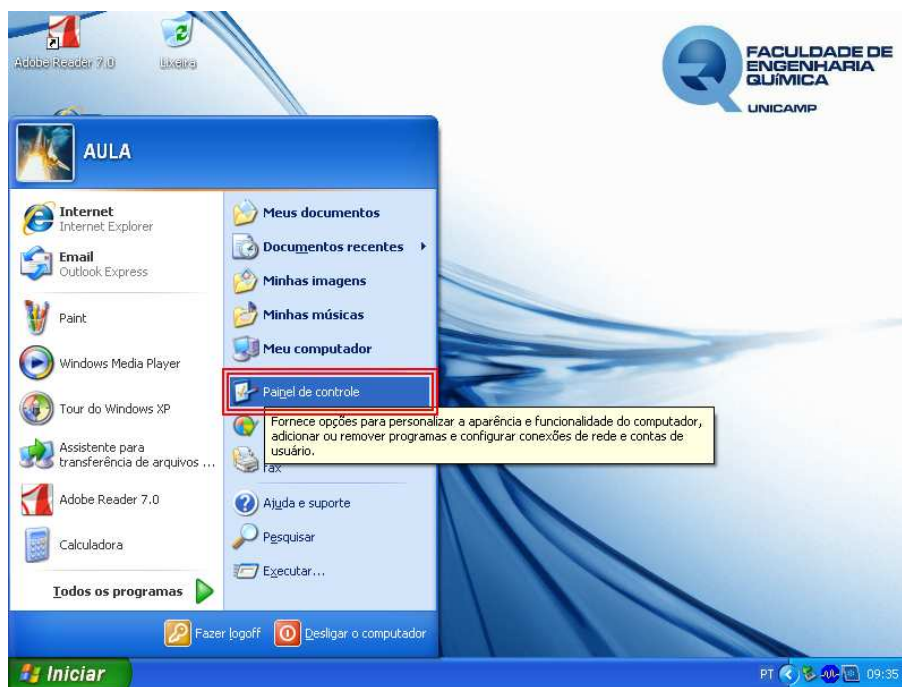


Figura A.1: Primeiro passo para configuração do *driver* ODBC

2. Dar um clique duplo em 'Ferramentas Administrativas'

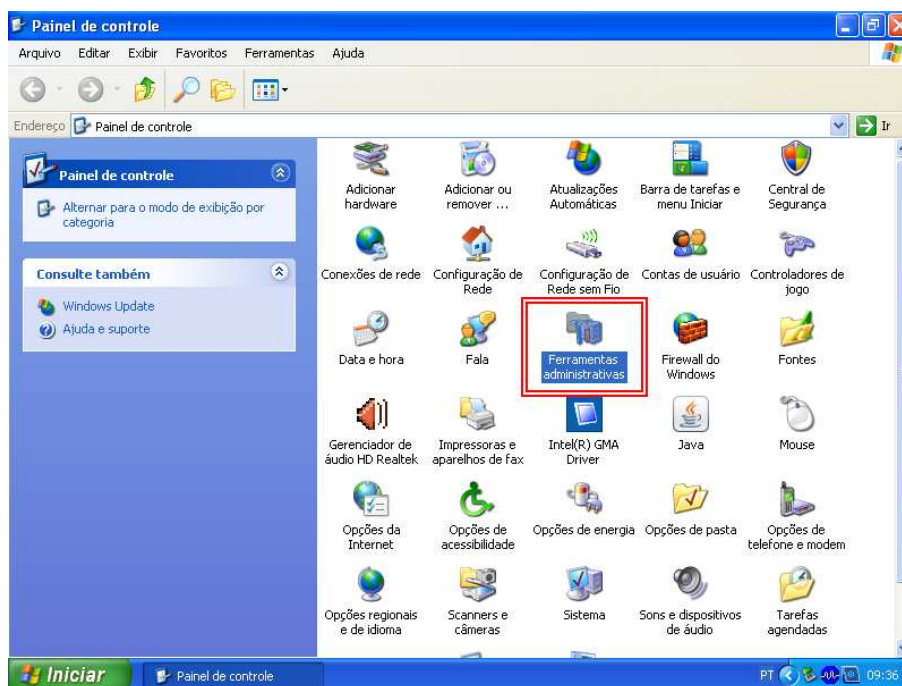


Figura A.2: Segundo passo para configuração do *driver* ODBC

### 3. Dar um clique duplo em 'Fontes de dados (ODBC)'

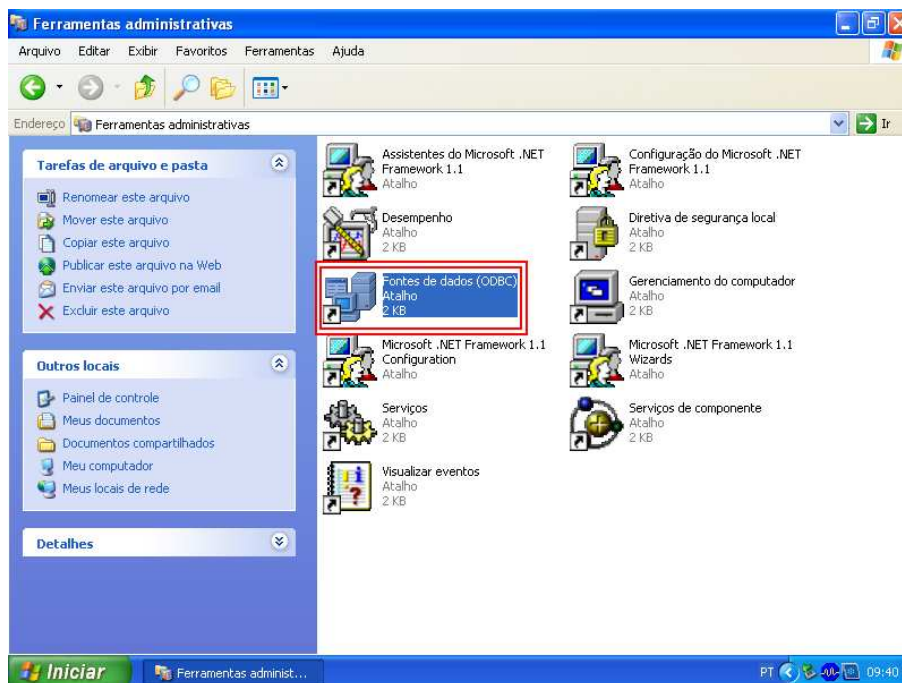


Figura A.3: Terceiro passo para configuração do *driver* ODBC

4. Na guia 'Fonte de dados de usuário', clicar em 'Adicionar'

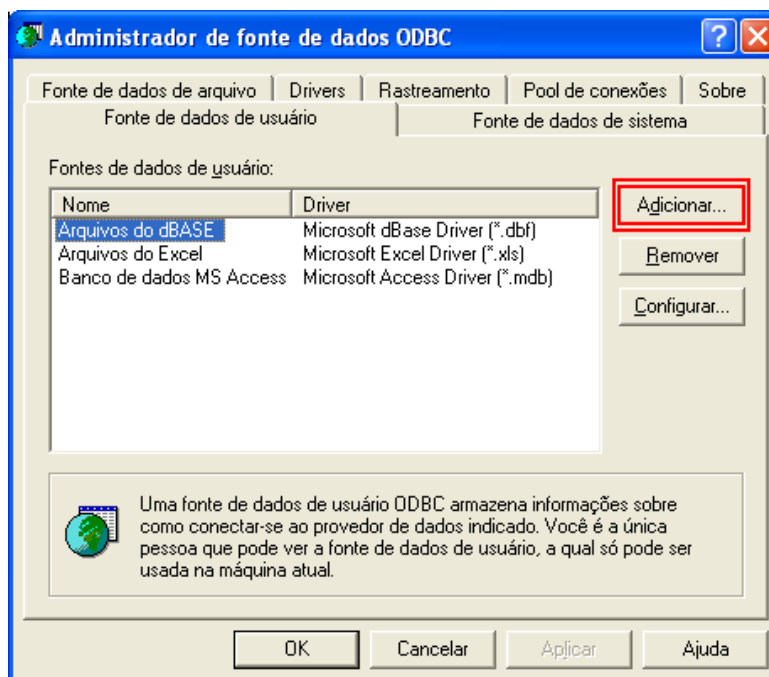


Figura A.4: Quarto passo para configuração do *driver* ODBC

5. Dar um clique duplo em 'Driver do Microsoft Excel(\*.xls)'

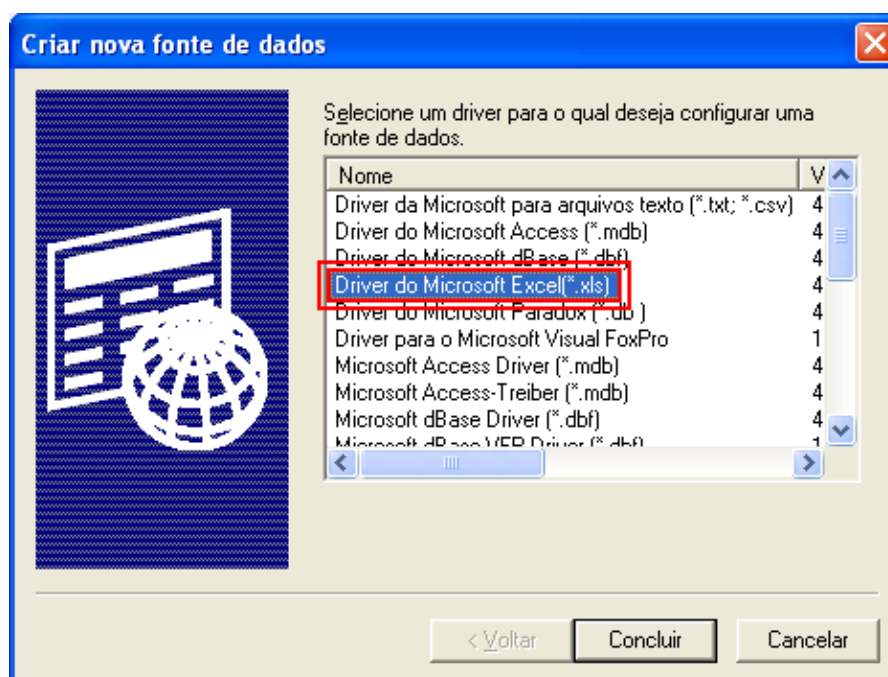


Figura A.5: Quinto passo para configuração do *driver* ODBC



6. Preencher o nome da fonte de dados (1), descrição (2) e clicar em ‘Selecionar pasta de trabalho...’ (3)

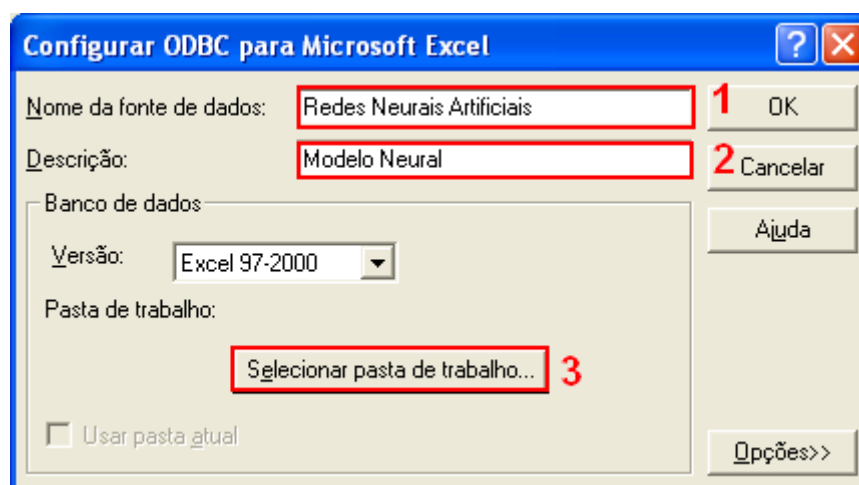


Figura A.6: Sexto passo para configuração do *driver* ODBC

7. Selecionar o arquivo que contém a planilha a ser utilizada na aplicação

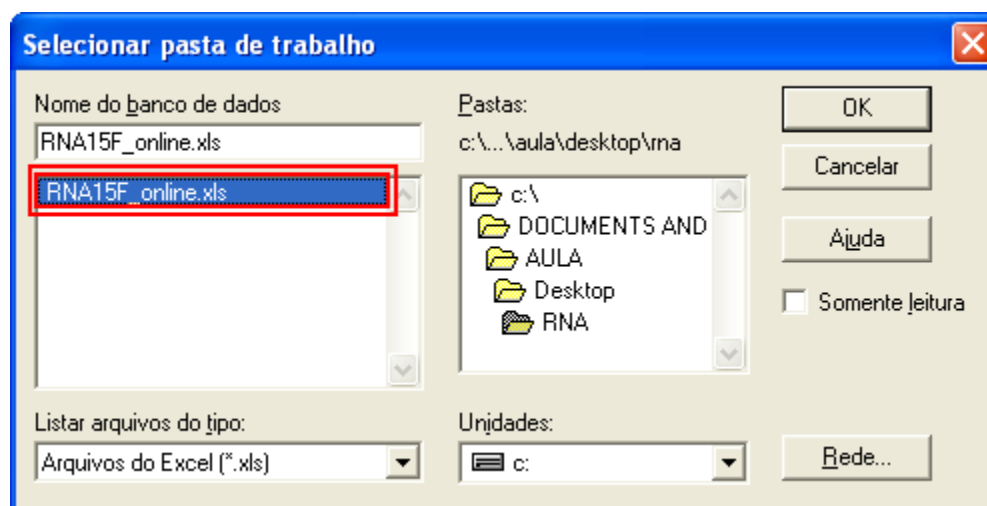


Figura A.7: Sétimo passo para configuração do *driver* ODBC

8. Conferir se o arquivo foi selecionado corretamente (1), clicar em ‘Opções’ (2) e retirar a seleção da opção ‘Somente leitura’ (3)

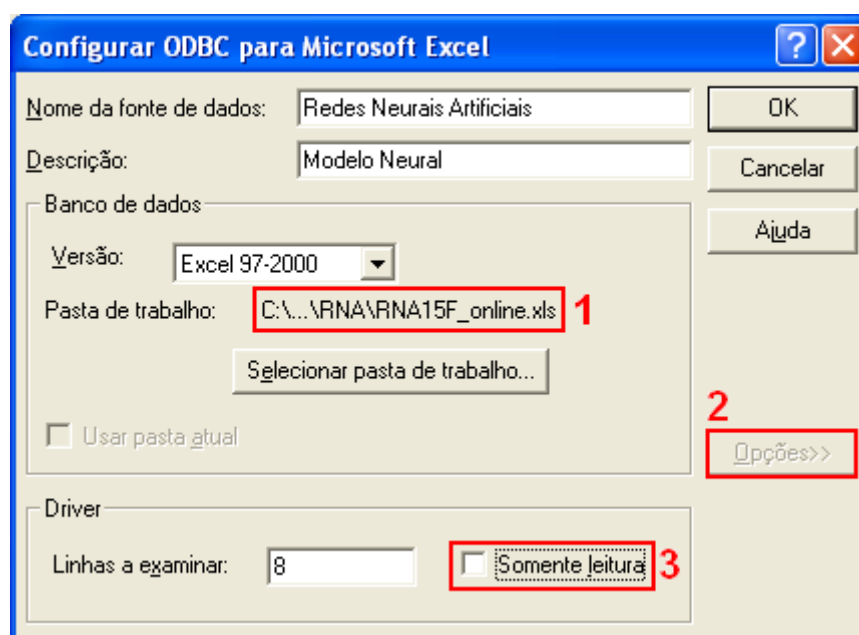


Figura A.8: Oitavo passo para configuração do *driver* ODBC

9. Clicar em 'Ok'

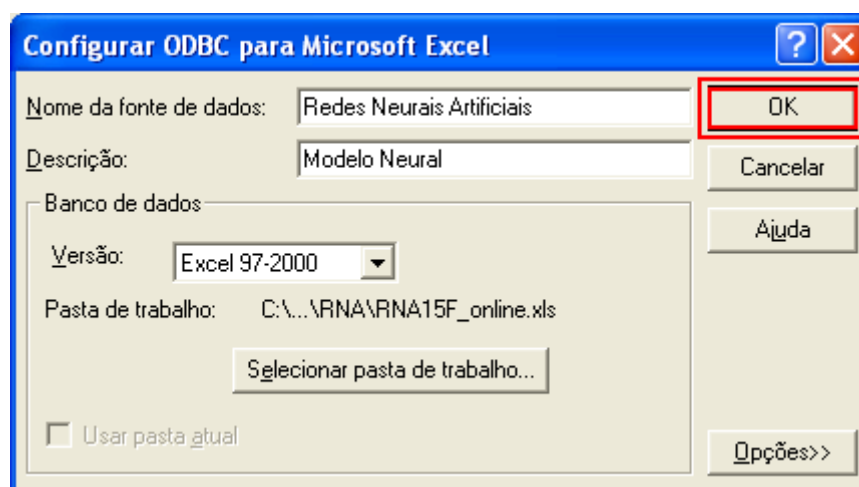


Figura A.9: Nono passo para configuração do *driver* ODBC

10. Clicar em 'Ok'

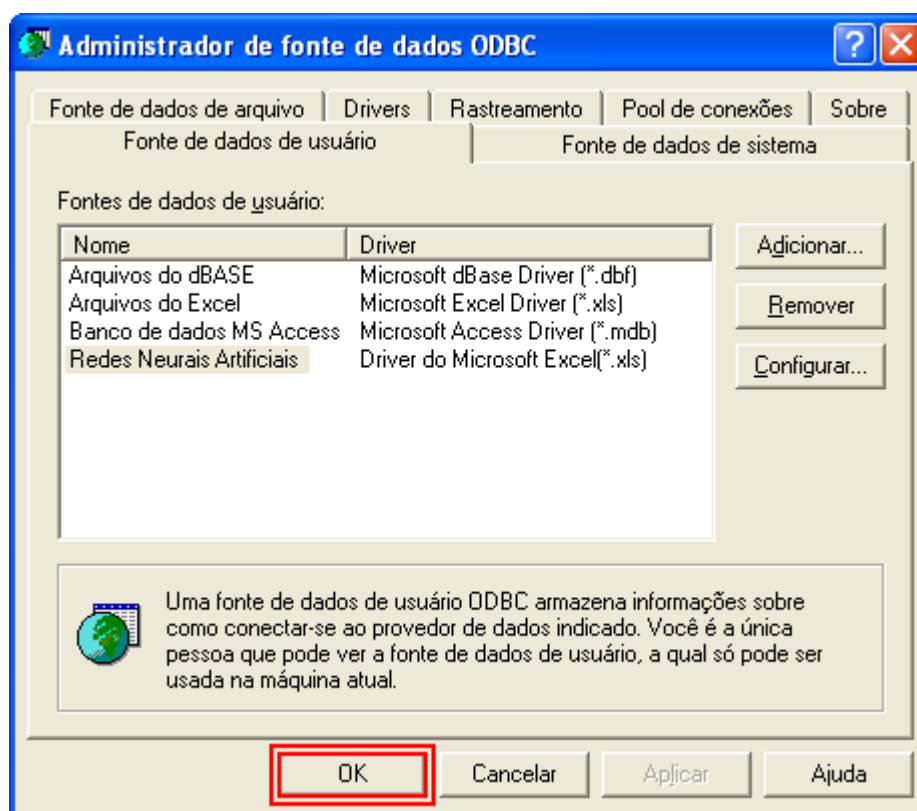


Figura A.10: Décimo passo para configuração do *driver* ODBC

Após esta configuração inicial, deve-se proceder com a configuração da planilha ODBC no Indusoft. Optou-se pela criação de duas planilhas: uma para escrita dos dados do Indusoft para o Excel (planilha “escrita”) e outra para leitura dos valores calculados no Excel para o Indusoft (planilha “leitura”). Na Figura A.11, é apresentada a planilha “escrita”.

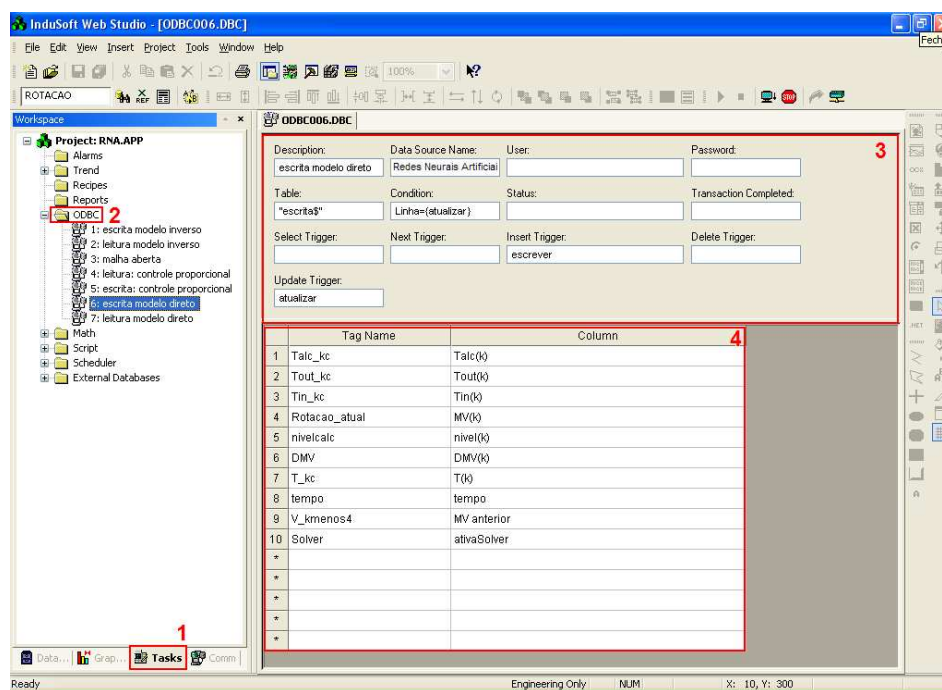


Figura A.11: Configuração da planilha ODBC no Indusoft, para escrever valores do Indusoft no Excel

Na Aba 'Tasks' (1), deve-se inserir uma nova planilha na pasta 'ODBC' (2), clicando com o botão direito em 'ODBC'. Na área marcada com o número 3, os campos devem ser preenchidos conforme descrito a seguir:

- *Description*: descrição da planilha
- *Data Source Name*: nome da fonte de dados; deve ser o mesmo nome preenchido no campo 1 do passo número 6 da configuração do *driver* ODBC no Windows
- *Table*: nome da planilha do Excel que receberá os dados do Indusoft; deve ter a forma "nome\$"
- *Condition*: deve ser preenchido caso se deseje que os novos dados sejam escritos por cima dos anteriores. Quando não preenchido, os novos dados serão inseridos na linha seguinte da planilha Excel
- *Insert Trigger*: deve ser preenchido com o *tag* do tipo Booleano que deverá ter seu valor alternado entre 0 e 1 para iniciar a escrita dos valores

- *Update Trigger*: deve ser preenchido com o mesmo *tag* utilizado no campo *Condition*. Também deve ser do tipo Booleano e deve ser alternado entre 0 e 1 para atualizar os dados no Excel.

Na área marcada com o número 4 na Figura A.11, colocam-se os *tags* cujos valores serão escritos na planilha Excel (coluna '*Tag Name*') e os nomes das colunas do Excel onde devem ser escritos os dados correspondentes (coluna '*Column*').

A configuração da planilha ODBC "leitura" no Indusoft é similar, como mostrado na Figura A.12.

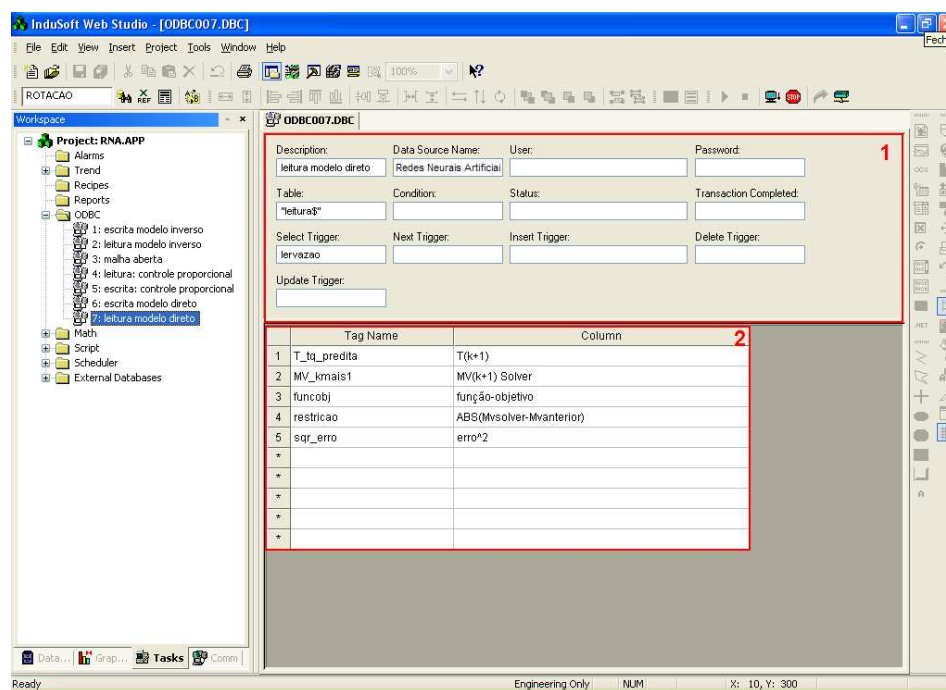


Figura A.12: Configuração da planilha ODBC no Indusoft, para ler valores do Excel, atribuindo-os a *tags* no Indusoft

Na área marcada com o número 1, os campos '*Description*', '*Data Source Name*' e '*Table*' devem ser preenchidos como no caso da planilha "escrita". No

campo '*Select Trigger*', deve ser colocado um *tag* Booleano que deve ser alternado entre 0 e 1 de acordo com o intervalo de amostragem desejado.

Na área marcada com o número 2, colocam-se os *tags* que receberão os valores calculados pelo Excel (coluna '*Tag Name*') e os nomes das colunas do Excel a partir de onde serão lidos os dados correspondentes (coluna '*Column*').

## APÊNDICE B: *SCRIPTS* DE PROGRAMAÇÃO UTILIZADOS NO DESENVOLVIMENTO DO PROJETO

### *Script* de treinamento da RNA

```
clear all; clc; %limpa as variáveis do histórico e limpa o prompt%
M = load('treinal5F.dat'); %carrega o arquivo de dados%
M = M'; %calcula a matriz transposta%
entrada = M(1:7,:); %define os dados de entrada da rede%
saida = M(8,:); %define os dados de saída da rede%
[entradan,minentrada,maxentrada,saidan,minsaida,maxsaida]=premnmx(
entrada,saida);
%define os parâmetros máximos e mínimos das matrizes de entrada e
saída
%e faz normalização%

net.numinputs = size(entrada(:,,:),7); %nº variáveis de entrada%
net.numlayers = 2; %nº de camadas sem a camada de entrada%
net = newff(minmax(entradan(:,:)), [14
1], {'tansig', 'purelin'}, 'trainlm');
%cria a rede, definindo número de neuronios e função de ativação
das camadas%
%intermediária e de saída%
net.trainParam.epochs = 2000; %nº de passos%
net.trainParam.goal = 1e-4; %convergência desejada%
net.initFcn = 'initlay'; %função que inicia os pesos e bias%
net.performFcn = 'sse'; %função objetivo a ser minimizada%
net.trainParam.min_grad = 1e-100; %mínimo gradiente%
net.trainParam.mu_max = 1e+400; %máximo MU%
net = init(net);

[net,tr] = train(net,entradan(:,:),saidan(:,:)); %realiza o
treinamento da rede%
%pesos e bias da rede determinados e guardados em 'net'%

Y = sim(net,entradan(:,:)); %simula com os dados de entrada do
```

```

arquivo de treinamento%
X = postmmx(Y,minsaida,maxsaida); %desnormaliza dados de saida%

figure(1);
[m,b,r]=postreg(X(1,:),saida(1,:)); %gráfico da saída real versus
calculada%

figure(2);
plot(saida(1:,:),'-k') %grafica a saida real do arquivo de
treinamento%
hold on
plot(X(1:),'or'); %grafica a saida calculada pela rede%
xlabel('amostragem','FontSize',20); %rótulo eixo x, fonte tamanho
20%
ylabel('PV (%)','FontSize',20); %rótulo eixo y, fonte tamanho 20%
legend('saída real','saída calculada',1); %legenda no canto
superior direito%
hold off

save 'rna15F' net; %salva a rede criada%

```

### *Script de teste da RNA*

```

clear all; clc; %limpa as variáveis do histórico e limpa o prompt%
load rna15F; %carrega a rede gravada
M = load('treina15F.dat'); %carrega arquivo de dados para
treinamento, para pegar a mesma normalização no treinamento e no
teste%
M = M'; %transposta%
intre = M(1:7,:); %dados de entrada da rede%
outtre = M(8,:); %dados de saída da rede%
[intren,minintre,maxintre,outtren,minouttre,maxouttre]=premmx(int
re,outtre); %normaliza dados%

N = load('ensaio137.dat'); %carrega arquivo de dados para teste%
N = N';
in = N(1:7,:); %dados de entrada da rede%
out = N(8,:); %dados de saída da rede%
[inn] = trammmx(in,minintre,maxintre); %normaliza dados%

```



```
Y = sim(net,inn(:,:)); %simula com os dados de entrada do arquivo
de teste%
X = postmmx(Y,minouttre,maxouttre); %desnormaliza dados de saida%

figure(2);
plot(out(1,:),'-k') %grafica a saida real do arquivo de teste%
hold on
plot(X(1:,:),'om'); %grafica a saida calculada pela rede%
xlabel('amostragem','FontSize',20); %rótulo eixo x, fonte tamanho
20%
ylabel('PV (°C)','FontSize',20); %rótulo eixo y, fonte tamanho 20%
legend('saída real','saída calculada',1); %legenda no canto
superior direito%
colordef white;
hold off

figure(3);
[m,b,r]=postreg(X(1:,:),out(1:,:)); %grafico da saida real versus
calculada%
xlabel('T','FontSize',20); %rótulo eixo x, fonte tamanho 20%
ylabel('A','FontSize',20); %rótulo eixo y, fonte tamanho 20%
whitebg([1 1 1]);
colordef white;
```

### Script utilizado no desenvolvimento do otimizador

```
'ROTINA PARA CÁLCULO DA ROTAÇÃO DA BOMBA DE PROPILENOGLICOL
'NUM SISTEMA DE PRECIPITAÇÃO DE BROMELINA
'Autora: Engenheira Química Tatiana Lie Fujiki
'Mestranda - Engenharia Química
'Orientadora: Profª Drª Ana Maria Frattini Fileti
'Área de Concentração: Sistemas de Processos Químicos e
Informática
'Faculdade de Engenharia Química - UNICAMP

'A cada alteração na planilha "escrita", a sub-rotina "resolva" é
executada
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target
As Range)
If Sh.Name = "escrita" Then Call resolva
End Sub

Sub resolva()
'cálculo via Solver
'redefinir tudo
SolverReset
'parâmetros do Solver
SolverOK
SetCell:=Sheets("leitura").Range("B2"),MaxMinVal:=2,
ByChange:=Sheets("leitura").Range("C2")
'restrições
SolverAdd CellRef:=Sheets("leitura").Range("C2"),
Relation:=1, FormulaText:="100"
SolverAdd CellRef:=Sheets("leitura").Range("C2"),
Relation:=3, FormulaText:="0"
SolverAdd CellRef:=Sheets("leitura").Range("J2"),
Relation:=1, FormulaText:="35"
'finaliza cálculo
SolverSolve UserFinish:=True
'manter solução do Solver
SolverFinish KeepFinal:=1
End Sub
```

# APÊNDICE C: CURVAS DE CALIBRAÇÃO DE BOMBAS E SENSORES

## C.1. Calibração da bomba de deslocamento positivo

A calibração da bomba de deslocamento positivo foi realizada com a solução de propilenoglicol a 50% v/v, à mesma temperatura de realização dos ensaios, que era de 0°C. A curva de calibração obtida é mostrada na Figura C.1:

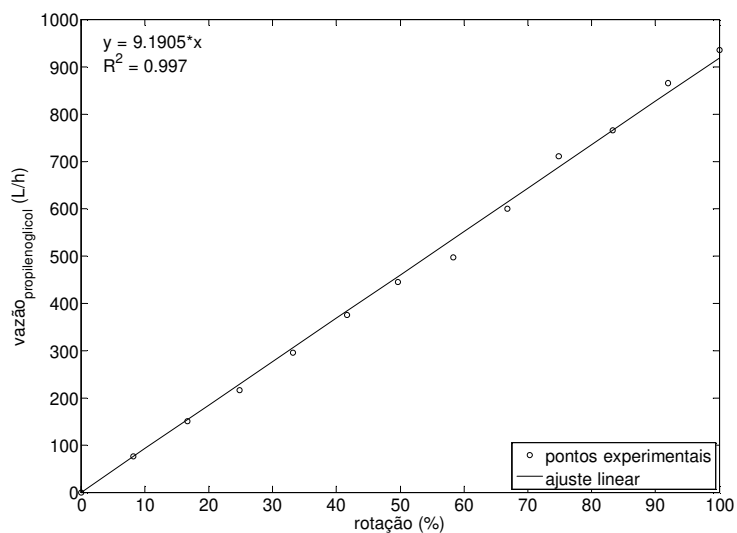


Figura C.1: Curva de calibração da bomba de propilenoglicol

## C.2. Calibração da bomba peristáltica

A curva de calibração da bomba de álcool é apresentada na Figura C.2:

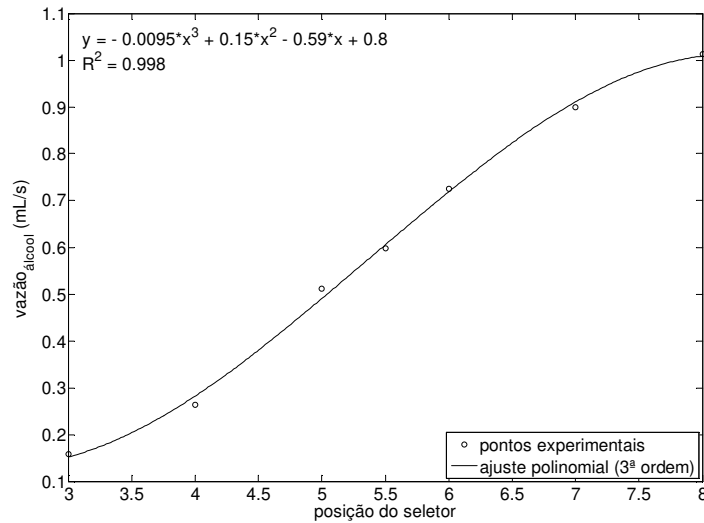


Figura C.2: Curva de calibração da bomba peristáltica

### C.3. Calibração das termorresistências

Os sensores de temperatura foram calibrados segundo os gráficos mostrados na Figura C.3 à Figura C.6:

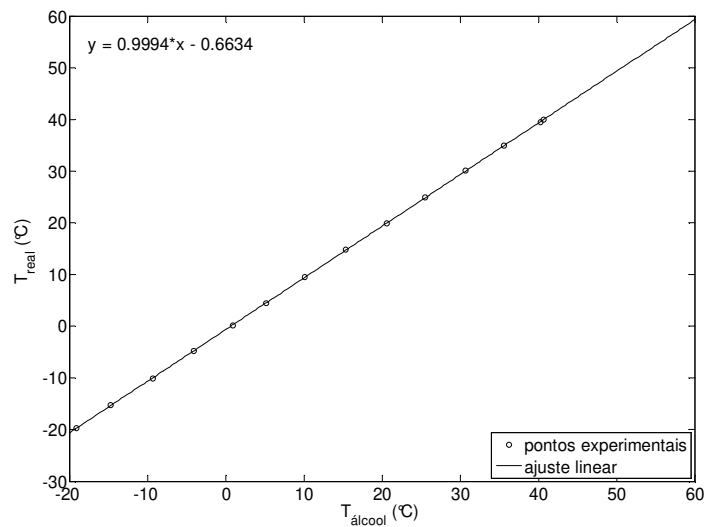


Figura C.3: Curva de calibração do sensor de temperatura do álcool

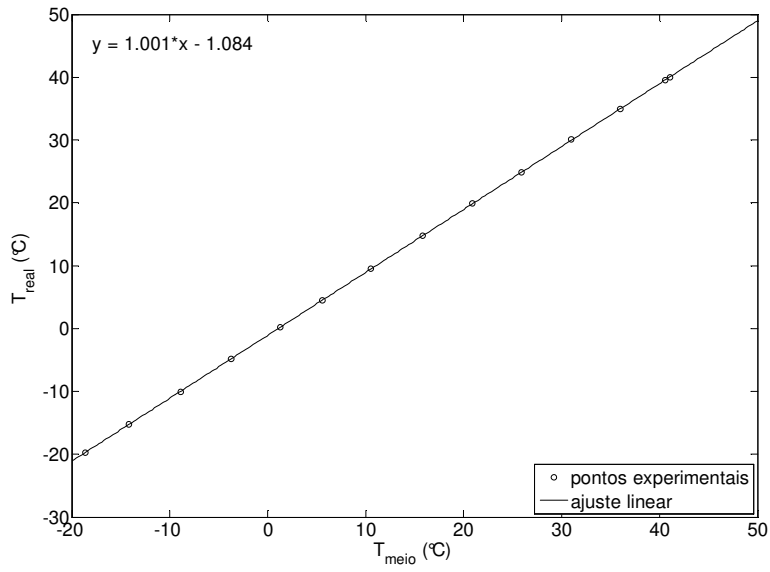


Figura C.4: Curva de calibração do sensor de temperatura do meio precipitante

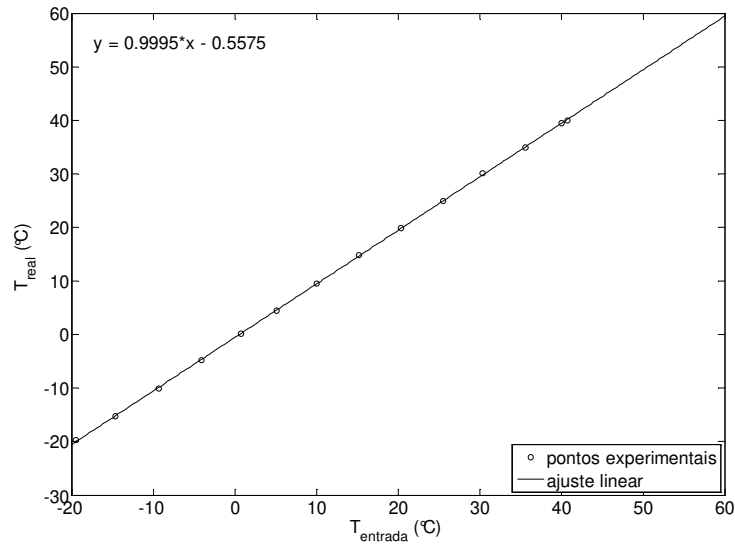


Figura C.5: Curva de calibração do sensor de temperatura de entrada do propilenoglicol

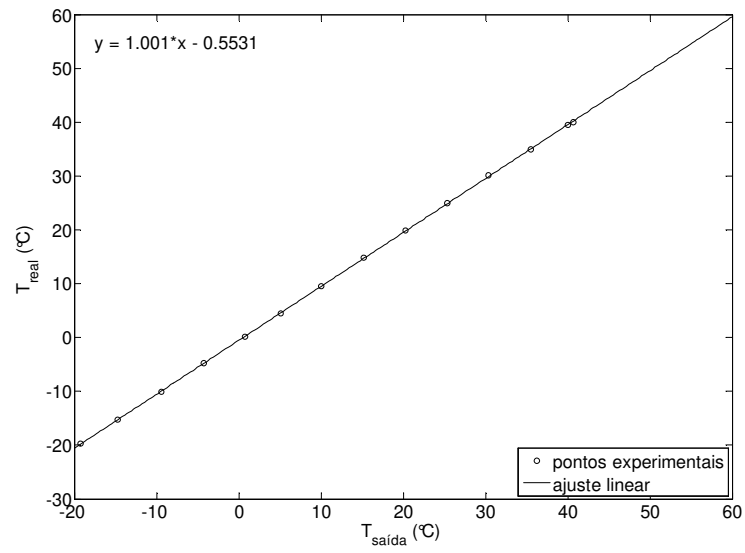


Figura C.6: Curva de calibração do sensor de temperatura de saída do propilenoglicol

## APÊNDICE D: TRATAMENTO DE RESÍDUOS

O tratamento de resíduos foi realizado numa coluna de destilação em batelada, utilizando uma razão de refluxo de 0,1.

A curva de calibração do refratômetro utilizado para determinar a concentração de álcool na mistura é apresentada na Figura D.1:

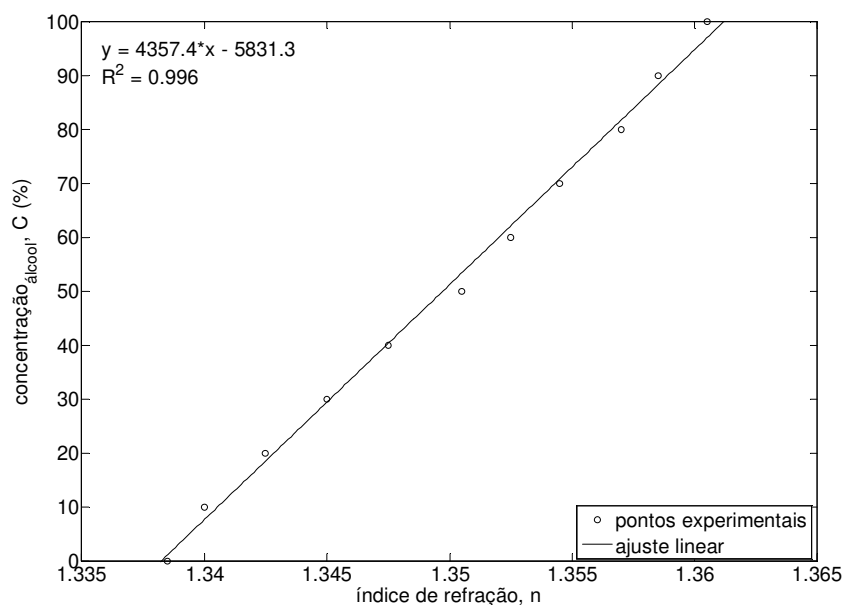


Figura D.1: Curva de calibração do refratômetro para análise das amostras de resíduo da precipitação de bromelina

Para uma amostra de 2 L de resíduo, obtiveram-se os seguintes dados:

$$V_{\text{álcool}} = 890\text{mL} \text{ e } C_{\text{álcool}} = 92\% .$$

Este álcool obtido pode ser destinado à realização de novos ensaios de precipitação, desde que constatada a sua adequação para este fim (através da análise de atividade da enzima obtida, por exemplo).