

**AVALIAÇÃO DE CONTROLE NEURAL A UM PROCESSO DE
QUATRO TANQUES ACOPLADOS**

JÚLIO CÉSAR PEIXOTO DE OLIVEIRA

**TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**AVALIAÇÃO DE CONTROLE NEURAL A UM PROCESSO
DE QUATRO TANQUES ACOPLADOS**

JÚLIO CÉSAR PEIXOTO DE OLIVEIRA

ORIENTADOR: ADOLFO BAUCHSPIESS

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

**PUBLICAÇÃO: PPGENE.TD-044/09
BRASÍLIA/DF: AGOSTO – 2009**

FICHA CATALOGRÁFICA

OLIVEIRA, JÚLIO CESAR PEIXOTO DE

Avaliação de Controle Neural a um Processo de Quatro Tanques Acoplados.
[Distrito Federal] 2009.

xv, 115p., 210 x 297 mm (ENE/FT/UnB, Doutor, Tese de Doutorado – Universidade de Brasília. Faculdade de Tecnologia, Departamento de Engenharia Elétrica)

- | | |
|----------------------------|----------------------------------|
| 1. Controle de Processos | 2. Sistemas de Nível de Líquidos |
| 3. Sistemas Não-Lineares | 4. Redes Neurais Artificiais |
| 5. Controlador Programável | 6. NARMA-L2 |
| 7. Identificação | |

I. ENE/FT/UnB

II. Título(série)

REFERÊNCIA BIBLIOGRÁFICA

OLIVEIRA., J. C. P.(2009). Avaliação de Controle Neural a um Processo de Quatro Tanques Acoplados. Tese de Doutorado em Engenharia Elétrica, Publicação PPGENE.TD-044/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 115p.

CESSÃO DE DIREITOS

NOME DO AUTOR:

Júlio César Peixoto de Oliveira.

TÍTULO DA TESE DE DOUTORADO:

Avaliação de Controle Neural a um Processo de Quatro Tanques Acoplados.

GRAU: Doutor ANO: 2009

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa tese de doutorado pode ser reproduzida sem autorização por escrito do autor.

Júlio César Peixoto de Oliveira

Agradecimentos

Em primeiro lugar, agradeço a Deus pelo dom da vida, da sabedoria, da perseverança, da saúde e porque compreendeu o meu anseio e me direcionou para atingir o meu objetivo.

Agradeço ao Professor Dr. Adolfo Bauchspiess, pela orientação, colaboração, paciência e apoio brindado ao longo desta tese de doutorado. Agradeço aos Professores da Faculdade de Tecnologia da UNB e ao Professor Peterson Resende da UFMG.

Agradeço a minha família pela compreensão que tiveram enquanto estive realizando esta tese de doutorado.

Agradeço aos meus colegas do laboratório de automação, pelas trocas de informações, sugestões e amizade, construídas durante o curso.

Agradeço aos funcionários do ENE/FT/UnB do Departamento de Engenharia Elétrica, que muito contribuíram para o sucesso deste trabalho.

Aos membros da banca examinadora, pela predisposição em analisar esta tese de doutorado e pelas sugestões recebidas.

Os meus agradecimentos se estendem, também, a todos que, de alguma forma, contribuíram com a realização desta tese de doutorado.

Júlio César Peixoto de Oliveira

Dedicatória

Dedico esta tese de doutorado a todos que de alguma forma colaboraram com a minha formação e para a realização da mesma.

Júlio César Peixoto de Oliveira

RESUMO

AVALIAÇÃO DE CONTROLE NEURAL A UM PROCESSO DE QUATRO TANQUES ACOPLADOS

Autor: Júlio César Peixoto de Oliveira

Orientador: Adolfo Bauchspiess

Programa de Pós-Graduação em Engenharia Elétrica

Brasília, agosto de 2009

O objetivo desta tese de doutorado é a avaliação de controle neural em um processo de quatro tanques acoplados, em escala de laboratório, onde se deseja controlar a altura do nível de água do quarto tanque. A dinâmica desse processo é não linear, uma vez que as vazões nos tanques dependem da raiz quadrada das alturas dos níveis de água. Sendo esse tipo de processo bastante comum em escala industrial principalmente nos ramos químico e petroquímico. Por se tratar de um processo não linear, o desempenho de controladores clássicos depende fortemente da faixa de operação do processo, o que exigiria, assim, ajuste dos seus parâmetros em diferentes pontos de operação. Vários controladores são implementados, objetivando a avaliação, pesquisa, validação e melhoria no desempenho de controladores neurais artificiais.

A resposta característica deste processo não permite um bom desempenho, numa ampla faixa de operação, utilizando-se controladores convencionais, justificando assim o uso dos controladores neurais. Para o projeto, implementação, simulação e controle neural no processo, foram implementados em um computador comercial de uso geral, utilizando o software Matlab/Simulink. Para executar o controlador neural, utilizaram-se módulos eletrônicos de conversão.

A metodologia empregada foi: a identificação, treinamento, simulação e controle para a faixa de operação pretendida com os sinais obtidos da planta. O modelo aproximador NARMA-L2 (*Nonlinear Autoregressive-Moving Average-Norma-L2*), foi inicialmente usado para modelar a planta e posteriormente a RNA foi usada no modelo da planta para calcular a lei de controle. O algoritmo proposto por Narendra, K.S. and Mukhopadhyay, 1997, transforma o sistema não linear em um sistema linear através do cancelamento das não linearidades. A vantagem sobre a engenharia convencional de controle é que a rede neural aprende com dados obtidos do processo. Experimentos e simulações de controles foram realizados nesse processo.

Resultados experimentais e tabelas comparativas são apresentados ao longo desta tese. Mostra-se a viabilidade da abordagem neural de pesos fixos para o controle de plantas não-lineares. Ressalta-se, também, que para esse processo, existem casos onde os controladores neurais utilizados não podiam ser sintonizados para dar um desempenho satisfatório.

A contribuição principal desta tese consiste em mostrar uma avaliação do controlador neural NARMA-L2, em um complexo processo de quarta ordem não linear. Uma avaliação, até o momento, despercebida.

Palavras-chave: controle de processos, sistemas não lineares, redes neurais artificiais, controlador programável, NARMA-L2, identificação, sistema de nível de líquidos.

ABSTRACT

EVALUATION OF NEURAL CONTROL TO A PROCESS OF FOUR INTERCONNECTED TANKS

Author: Júlio César Peixoto de Oliveira

Supervisor: Adolfo Bauchspiess

Programa de Pós-Graduação em Engenharia Elétrica

Brasília, agosto de 2009

The main goal of this doctorate thesis is the evaluation of neural control on a four interconnected tanks process, in laboratory environment, where the controlled variable is the water level height of the fourth tank. The dynamics of this process is non linear, as the outflow from the tanks depends on the square root of corresponding water height. This type of process is quite common in industrial applications as, for example, chemical and petrochemical plants. Because the process is non linear, the performance of conventional control techniques depends strongly on the operation points, thus, on parameters adjustment for each operation point. Many different controllers have been implemented aiming the evaluation, research, validation and improvement of artificial neural controllers.

The characteristic behavior of this process does not allow satisfactory performance on a wide operation range with the use of conventional controllers, what justifies the use of neural controllers. For the design, implementation and simulation of the neural controller on the process, a general purpose commercial computer had been used to run a Matlab/Simulink software environment. To implement the neural controller, electronic interface modules have been used.

The adopted methodology was: identification, training, simulation and control for the proposed operation range using data captured from the plant. The NARMA-L2 structure (Nonlinear Autoregressive-Moving Average-Norm-L2) was initially used to model the plant, while the RNA plant model was used subsequently to calculate the control law. This algorithm, proposed by Narendra, K.S. and Mukhopadhyay, 1997, transforms, in the ideal case, the non linear system into a linear system through the additive and multiplicative cancellation of non linearities. The advantage of neural networks over conventional control engineering is that it neural networks learn from the process data. Many simulations and experiments with neural controllers have been carried out on the process.

Experimental results of neural control and comparative tables are presented throughout this thesis. The viability of neural networks with fixed weights for the control of non linear plants is demonstrated. It should be remarked that, for this process, there are cases for which the neural controller could not be tuned to deliver satisfactory performance.

The main contribution of this thesis is showing an evaluation of NARMA-L2 neural control on a complex fourth order non linear process. So far, an unseen evaluation.

Keywords: process control, non linear systems, artificial neural networks, programmable controller, NARMA-L2, identification, liquid level system.

SUMÁRIO

1 - INTRODUÇÃO	1
1.1 - AUTOMAÇÃO E SISTEMAS INDUSTRIAIS	1
1.2 - MOTIVAÇÃO	2
1.3 - A EVOLUÇÃO DOS EXPERIMENTOS NO LAVSI.....	2
1.4 - OBJETIVOS.....	3
1.5 - METODOLOGIA	3
1.6 - ORGANIZAÇÃO DESTE TRABALHO.....	4
1.7 - FUNDAMENTAÇÃO TEÓRICA	5
1.8 - REVISÃO DA LITERATURA	6
1.9 - ARTIGOS PUBLICADOS PELO AUTOR DURANTE O DOUTORADO	8
2 - O PROCESSO DE QUATRO TANQUES ACOPLADOS	10
2.1 - INTRODUÇÃO	10
2.2 - DESCRIÇÃO DA PLANTA	10
2.3 - MODELAGEM MATEMÁTICA.....	11
2.3.1 - PROCESSO DE UM TANQUE	13
2.3.2 - PARÂMETROS DO MODELO.....	17
2.3.3 - SENSORES DE PRESSÃO	18
2.3.4 - MOTO BOMBAS	18
2.3.5 - PARÂMETROS DAS VÁLVULAS	19
2.4 - INSTRUMENTAÇÃO DA PLANTA	20
2.4.1 - SENSORES DE NÍVEL.....	20
2.4.2 - MOTO BOMBA - ELEMENTOS FINAIS DE CONTROLE DA PLANTA.....	21
2.5 - INTERFACES ANALÓGICAS	21
3 - CONTROLE CONVENCIONAL DO PROCESSO DE TANQUES ACOPLADOS	23
3.1 - INTRODUÇÃO	23
3.2 - PROCESSO LINERARIZADO	24
3.3 - CONTROLE PI	25
3.4 - CONTROLE NO ESPAÇO-DE-ESTADOS	25
3.5 - CONTROLE NO ESPAÇO-DE-ESTADOS COM AÇÃO INTEGRAL.....	26
3.6 - SIMULAÇÃO DE CONTROLADORES CONVENCIONAIS.....	27

3.7 - IMPLEMENTAÇÃO DO CONTROLADOR PI NO PROCESSO	31
3.8 - CONTROLE NEURAL <i>GAIN SCHEDULING</i>	32
3.9 - COMENTÁRIOS	35
4 - REDES NEURAIS E CONTROLE NEURAL	36
4.1 - INTRODUÇÃO	36
4.2 - REDES NEURAIS MULTICAMADAS	36
4.3 - ALGORITMO DE TREINAMENTO	37
4.4 - TREINAMENTO DAS REDES NEURAIS COM RETROALIMENTAÇÃO	42
4.5 - CLASSES DOS ALGORITMOS DE TREINAMENTO	43
4.5.1 - MÍNIMOS LOCAIS E GLOBAIS	44
4.6 - CONTROLE NEURAL	51
4.7 - IMPLEMENTAÇÃO DO CONTROLE NEURAL PARA O PROCESSO	51
4.8 - IDENTIFICAÇÃO E CARACTERIZAÇÃO DA REDE NEURAL	55
4.8.1 - CARACTERIZAÇÃO	55
4.8.2 - IDENTIFICAÇÃO	59
4.9 - REDES NEURAIS COMO UMA OPÇÃO ALTERNATIVA	61
4.10 - COMENTÁRIOS	62
5 - CONTROLE NEURAL DO PROCESSO DE NÍVEL DE LÍQUIDOS	63
5.1 - INTRODUÇÃO	63
5.2 - IMPLEMENTAÇÃO DO CONTROLADOR NEURAL NO MATLAB	63
5.3 - RNA CONTROLANDO O MODELO DO PROCESSO	64
5.4 - IMPLEMENTANDO UMA RNA COMPOSTA DE TRÊS CAMADAS	69
5.5 - COMENTÁRIOS	73
6 - CONTROLADOR NARMA-L2	74
6.1 - FERRAMENTA NARMA-L2	74
6.2 - TAXA DE AMOSTRAGEM NO DESEMPENHO DO CONTROLADOR	78
6.3 - METODOLOGIA PARA A IMPLANTAÇÃO DO CONTROLADOR NARMA-L2	78
6.4 - O PROCEDIMENTO DE IDENTIFICAÇÃO UTILIZADO	79
7 - CONTROLE NARMA-L2 NO PROCESSO DE NÍVEL DE LÍQUIDOS	81
7.1 - INTRODUÇÃO	81
7.2 - TÉCNICAS DE CONTROLE	82

7.3 - O CONTROLADOR NARMA-L2 EM OPERAÇÃO.....	83
7.4 - NARMA-L2 CONTROLANDO O MODELO DO PROCESSO	89
7.5 - NARMA-L2 CONTROLANDO O PROCESSO.....	91
7.6 - IDENTIFICAÇÃO DO PROCESSO PELO NARMA-L2	94
7.7 - TREINAMENTO DA RNA UTILIZANDO O CONTROLADOR NARMA-L2.....	97
7.8 - NARMA-L2: SIMULAÇÃO E CONTROLE DA PLANTA.....	99
7.9 - COMENTÁRIOS.....	104
8 - CONCLUSÕES.....	106
REFERÊNCIAS BIBLIOGRÁFICAS	109
APÊNDICE	115

LISTA DE TABELAS

TABELA 2.1 - ESTIMATIVAS DOS PARÂMETROS K_{12} , K_{23} E K_{34}	20
TABELA 3.1 - MODELO LINEARIZADO PARA DIFERENTES PONTOS DE OPERAÇÃO	24
TABELA 5.1 - PARÂMETROS UTILIZADOS NA RNA.....	64
TABELA 7.1- RESPOSTAS PARA VÁRIOS ALGORITMOS UTILIZADOS NOS TANQUES.	85
TABELA 7.2- RESPOSTAS PARA VÁRIAS CONFIGURAÇÕES DOS TANQUES.	88
TABELA 7.3 - ARQUITETURA DE REDE - SIMULAÇÃO.....	90
TABELA 7.4 - DADOS DO TREINAMENTO - SIMULAÇÃO	90
TABELA 7.5 - RESPOSTA OBTIDA PELA CONFIGURAÇÃO DO PROCESSO - SIMULAÇÃO.....	90
TABELA 7.6 - TEMPO DE RESPOSTA DA PLANTA PARA O CONTROLADOR NARMA-L2	90
TABELA 7.7 - ARQUITETURA DE REDE - IDENTIFICAÇÃO	98
TABELA 7.8 - DADOS DO TREINAMENTO - IDENTIFICAÇÃO.....	98
TABELA 7.9 - RESPOSTAS OBTIDAS PELA CONFIGURAÇÃO DO PROCESSO - IDENTIFICAÇÃO....	98
TABELA 7.10 - TEMPO DE RESPOSTA DA SIMULAÇÃO DO CONTROLADOR NARMA-L2	100
TABELA 7.11 - TEMPO DE RESPOSTA DA PLANTA - IDENTIFICADA E TREINADA	102
TABELA 7.12 - RESPOSTAS OBTIDAS PELAS CONFIGURAÇÃO PARA NOVOS PARÂMETROS	102
TABELA 7.13 - TEMPO DE RESPOSTA DA SIMULAÇÃO PARA NOVOS PARÂMETROS.....	103
TABELA 7.14 - TEMPO DE RESPOSTA DA PLANTA PARA NOVOS PARÂMETROS	104

LISTA DE FIGURAS

FIGURA 2.1 – DUAS VISTAS DO PROCESSO DE NÍVEL DE LÍQUIDOS.	10
FIGURA 2.2 – REPRESENTAÇÃO ESQUEMÁTICA DA PLANTA DE TANQUES ACOPLADOS.	12
FIGURA 2.3 – DIAGRAMA DE BLOCOS DO MODELO DE TANQUES ACOPLADOS.	16
FIGURA 2.4 – MOTO BOMBA.	19
FIGURA 2.5 – VÁLVULA IMPLEMENTADA COM RANHURA E ENGASTE.	20
FIGURA 2.6 – MEDIÇÃO DE NÍVEL.	21
FIGURA 2.7 – MÓDULO DE ACIONAMENTO.	21
FIGURA 2.8 – CONTROLADOR PROGRAMÁVEL E OS MÓDULOS ANALÓGICOS.	22
FIGURA 2.9 – VISTA DO CONTROLE E DO PROCESSO.	22
FIGURA 3.1 – SIMULAÇÃO DOS CONTROLES PI, EE E EEP.	23
FIGURA 3.2 – SIMULAÇÃO DO CONTROLADOR EE.	26
FIGURA 3.3 – SIMULAÇÃO DO CONTROLADOR EEP.	26
FIGURA 3.4 – SIMULAÇÃO DOS CONTROLADORES PI, EE E EEI – VARIÁVEIS DE REFERÊNCIA E CONTROLE.	27
FIGURA 3.5 – SIMULAÇÃO DOS CONTROLADORES PI, EE E EEI – VARIÁVEIS DE CONTROLES.	28
FIGURA 3.6 – SIMULAÇÃO DOS CONTROLADORES PI, EE E EEI – DETALHE EM TORNO DO PONTO DE OPERAÇÃO DE 8 CM.	28
FIGURA 3.7 – SIMULAÇÃO DOS CONTROLADORES PI, EE E EEI – VARIÁVEL DE CONTROLE EM TORNO DO PONTO DE OPERAÇÃO DE 8 CM.	29
FIGURA 3.8 – SIMULAÇÃO DO CONTROLADOR PI – VARIÁVEL DE REFERÊNCIA EM TORNO DO PONTO DE OPERAÇÃO DE 3 CM.	29
FIGURA 3.9 – SIMULAÇÃO DO CONTROLE PI.	30
FIGURA 3.10 – CONFIGURAÇÃO DO CONTROLADOR PI IMPLEMENTADO NA PLANTA	31
FIGURA 3.11 – CONTROLE PI NA PLANTA.	31
FIGURA 3.12 – DIAGRAMA DA PLANTA DE TRÊS TANQUES ACOPLADOS.	32
FIGURA 3.13 – CONTROLADOR NEURAL <i>GAIN SCHEDULING</i>	33
FIGURA 3.14 – CONTROLADOR NEURAL <i>GAIN SCHEDULING</i> (REFERÊNCIA: 7,5 CM)	34
FIGURA 3.15 – CONTROLADOR NEURAL <i>GAIN SCHEDULING</i> (REFERÊNCIA: 10 CM)	34
FIGURA 3.16 – CONTROLADOR NEURAL <i>GAIN SCHEDULING</i> (REFERÊNCIA: 12,5 CM)	35
FIGURA 4.1 - DIAGRAMA FUNCIONAL DE UM NEURÔNIO ARTIFICIAL.	36
FIGURA 4.2 – CLASSES DOS ALGORITMOS DE TREINAMENTO	43
FIGURA 4.3 – REPRESENTAÇÃO: MÍNIMO LOCAL E GLOBAL	44

FIGURA 4.4 – MODELO INCREMENTAL PI NO SISTEMA DE TANQUES ACOPLADOS	53
FIGURA 4.5 – ESTRUTURA DA REDE NEURAL ARTIFICIAL USADA PARA O TREINAMENTO	54
FIGURA 4.6 – MODELO DE SIMULAÇÃO DO CONTROLE <i>GAIN SCHEDULING</i>	55
FIGURA 4.7 – REPRESENTAÇÃO ESQUEMÁTICA DA PLANTA SISO - MODELO I	57
FIGURA 4.8 – REPRESENTAÇÃO ESQUEMÁTICA DA PLANTA SISO - MODELO II	57
FIGURA 4.9 – REPRESENTAÇÃO ESQUEMÁTICA DA PLANTA SISO - MODELO III.....	58
FIGURA 4.10 – REPRESENTAÇÃO ESQUEMÁTICA DA PLANTA SISO - MODELO IV	58
FIGURA 4.11 – REPRESENTAÇÃO ESQUEMÁTICA PARA A IDENTIFICAÇÃO PARALELA	59
FIGURA 4.12 – REPRESENTAÇÃO ESQUEMÁTICA PARA A IDENTIFICAÇÃO SÉRIE	60
FIGURA 4.13 – REPRESENTAÇÃO ESQUEMÁTICA PARA A IDENTIFICAÇÃO NÃO LINEAR	61
FIGURA 5.1 – REPRESENTAÇÃO DO CONTROLADOR NEURAL.....	65
FIGURA 5.2 – CONTROLE PID (REFERÊNCIA: 4,6 CM)	65
FIGURA 5.3 – SIMULAÇÃO DO CONTROLADOR NEURAL (REFERÊNCIA: 4,6 CM)	66
FIGURA 5.4 – CONTROLADOR NEURAL (REFERÊNCIA: 4,6 CM)	66
FIGURA 5.5 – SIMULAÇÃO DO CONTROLADOR NEURAL, NOVOS PARÂMETROS DA PLANTA (REFERÊNCIA: 4,6 CM).....	67
FIGURA 5.6 – SIMULAÇÃO DO CONTROLADOR NEURAL, NOVOS PARÂMETROS DA PLANTA (REFERÊNCIA: 7,4 CM).....	68
FIGURA 5.7 – SIMULAÇÃO DO CONTROLADOR NEURAL (REFERÊNCIA: 7,4 CM) E OS RESPECTIVOS NÍVEIS DOS TANQUES H_1, H_2, H_3 E H_4	68
FIGURA 5.8 – CONTROLADOR NEURAL DE TRÊS CAMADAS, CONTROLANDO O MODELO DA PLANTA	70
FIGURA 5.9 – SIMULAÇÃO DO CONTROLADOR NEURAL (REFERÊNCIAS: 2 CM, 4 CM E 2 CM). 71	
FIGURA 5.10 – SIMULAÇÃO DO CONTROLADOR NEURAL (REFERÊNCIAS: 2 CM, 6 CM E 2 CM)71	
FIGURA 5.11 – CONTROLADOR NEURAL DE TRÊS CAMADAS, CONTROLANDO A PLANTA VIA INTERFACES OPC	72
FIGURA 5.12 – CONTROLADOR NEURAL (REFERÊNCIA: 2 CM)	73
FIGURA 6.1 – CONTROLADOR NARMA-L2.....	75
FIGURA 6.2 – DIAGRAMA EM BLOCOS DO MODELO NARMA-L2.....	76
FIGURA 6.3 – DIAGRAMA EM BLOCOS DO CONTROLADOR NARMA-L2	77
FIGURA 6.4 – MODELO DO SISTEMA DE CONTROLE	79
FIGURA 7.1 - MODELO DO SISTEMA DE CONTROLE.....	83
FIGURA 7.2 – (A) SIMULAÇÃO NO MODELO DA PLANTA (REFERÊNCIAS: 5 CM, 3 CM, 5 CM, 2 CM, 4 CM, 6 CM, 8 CM, 10 CM E 4 CM), TESTE 1.....	86

FIGURA 7.2 – (B) SIMULAÇÃO NO MODELO DA PLANTA (REFERÊNCIAS: 5 CM, 3 CM, 5 CM, 2 CM, 4 CM, 6 CM, 8 CM, 10 CM E 4 CM), TESTES 2.....	86
FIGURA 7.2 – (C) SIMULAÇÃO NO MODELO DA PLANTA (REFERÊNCIAS: 5 CM, 3 CM, 5 CM, 2 CM, 4 CM, 6 CM, 8 CM, 10 CM E 4 CM), TESTE 3.....	87
FIGURA 7.3 – (A) SIMULAÇÃO NO MODELO DA PLANTA (REFERÊNCIAS: 5 CM, 15 CM, 5 CM, 12 CM E 10 CM)	87
FIGURA 7.3 – (B) SIMULAÇÃO NO MODELO DA PLANTA (REFERÊNCIA: 10 CM).....	88
FIGURA 7.4 – SIMULAÇÃO DE CONTROLE (REFERÊNCIAS: 2 CM, 4 CM, 6 CM, 4 CM E 2 CM)...	89
FIGURA 7.5 – SISTEMA DE CONTROLE IMPLEMENTADO PARA O CONTROLE DA PLANTA	92
FIGURA 7.6 – DIAGRAMA DE COMUNICAÇÃO ENTRE O COMPUTADOR E O PROCESSO	92
FIGURA 7.7 – CONTROLADOR NARMA-L2, SINAIS DE PROCESSO (REFERÊNCIAS: 2 CM, 4 CM, 6 CM, 4 CM E 2 CM).....	93
FIGURA 7.8 – CONTROLADOR PID, SINAIS DE PROCESSO (REFERÊNCIAS: 2 CM, 4 CM, 6 CM, 4 CM E 2 CM)	93
FIGURA 7.9 - SISTEMA DE CONTROLE IMPLEMENTADO NO PROCESSO.	94
FIGURA 7.10 - ILUSTRAÇÃO DOS TANQUES ACOPLADOS.....	95
FIGURA 7.11 - SINAL GERADO PELO NARMA-L2, PARA A EXCITAÇÃO DA PLANTA.	96
FIGURA 7.12 - SISTEMA IMPLEMENTADO PARA A IDENTIFICAÇÃO DA PLANTA.	96
FIGURA 7.13 - SINAL IDENTIFICADO NA PLANTA	97
FIGURA 7.14 - VISÃO DOS GRÁFICOS GERADOS NO TREINAMENTO DA REDE NEURAL.....	98
FIGURA 7.15 - CONTROLADOR NARMA-L2 (COM A REDE NEURAL TREINADA) IMPLEMENTADO NO MODELO DA PLANTA.	99
FIGURA 7.16 - SIMULAÇÃO DO CONTROLADOR NARMA-L2 TREINADO NA PLANTA (REFERÊNCIAS: 3 CM, 4 CM E 2,5 CM)	100
FIGURA 7.17 - ESQUEMA DE MONTAGEM PARA O CONTROLADOR NARMA-L2, PARA A VALIDAÇÃO NA PLANTA DE TANQUES.....	101
FIGURA 7.18 - SINAIS OBTIDOS NA PLANTA (REFERÊNCIAS: 4 CM, 5 CM E 4 CM).	101
FIGURA 7.19 - NARMA-L2, TREINADO NA PLANTA E IMPLEMENTADO NO MODELO DA PLANTA (REFERÊNCIAS: 4,6 CM, 6,6 CM E 4,6 CM).	103
FIGURA 7.20 - SINAIS OBTIDOS NA PLANTA (REFERÊNCIAS: 5 CM, 6 CM E 5 CM).	104
FIGURA 8.1 - PROCESSO DE QUATRO TANQUES ACOPLADOS: (A) PROCESSO ATUAL; (B) PROCESSO SUGERIDO.	108

LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIACÕES

SÍMBOLOS

A_n	Área da secção transversal do tanque	[cm ²]
C_n	Capacitância do tanque n (secção transversal)	[cm ²]
k_{nm}	Resistência do tanque n para o tanque m	[s/cm ²]
h_n	Nível do tanque n	[cm]
q_{nm}	Vazão do tanque n para o tanque m	[cm ³ /s]
Q_{nm}	Vazão do tanque n para o tanque m em regime permanente	[cm ³ /s]
q_{in}	Vazão de entrada no tanque n	[cm ³ /s]
q_{on}	Vazão de saída no tanque n	[cm ³ /s]

NOMENCLATURA

$f(.)$	função escalar genérica
t	tempo
kp	ganho proporcional de um regulador PID
Td	tempo derivativo de um regulador PID
Ti	tempo integral de um regulador PID
Ts	período de amostragem
u	sinal de controle
y^*	valor da variável y em um ponto de operação do sistema
w	conjunto total de pesos de uma rede neural
η	parâmetro de aprendizagem da rede <i>backpropagation</i>
ρ	densidade do líquido
$\Sigma(x,y)$	sistema dinâmico genérico com parâmetros x e y
σ	função de ativação sigmoidal
$[.]^T$	transposição de um vetor ou matriz
δ	operador diferença em relação ao valor no ponto operação

ABREVIACÕES

<i>CLP</i>	controlador lógico programável
<i>PID</i>	controlador proporcional+integral+derivativo
<i>RNA</i>	rede neural artificial
<i>NARMA-L2</i>	non-linear auto-regressive moving average – L2 norm
<i>SP</i>	valor desejado para a variável de processo
<i>PV</i>	variável de processo
<i>MV</i>	variável de controle
<i>TDL</i>	time delay line
<i>GUI</i>	graphics user interface
<i>EE</i>	Espaço-de-Estados
<i>EI</i>	Espaço-de-Estados com ação Integral
<i>SISO</i>	single-input single-output

1 - INTRODUÇÃO

1.1 - AUTOMAÇÃO E SISTEMAS INDUSTRIAIS

É visível o crescente desenvolvimento das atividades industriais relacionadas com controle de processos industriais. A solução tradicional para os problemas de controle tem sido o uso do controlador PID sintonizado manualmente. Bons resultados são normalmente alcançáveis com algoritmo PID, embora os procedimentos de sintonia possam consumir tempo, além de ser necessária nova sintonia nos casos de mudança nos parâmetros do processo.

A pesquisa em redes neurais artificiais teve um grande interesse a partir dos anos 80. Esta área tem seus princípios assentados nos trabalhos de modelagem biológica de processos neurofisiológicos, cognitivos e comportamentais.

Tem-se observado um intenso esforço acadêmico nas últimas décadas no sentido de se fundamentar teórica e experimentalmente diversos esquemas de controle. Neste contexto, as redes neurais artificiais são uma abordagem que vem sendo utilizada para diversos problemas de controle de sistemas dinâmicos não-lineares. Isto se deve a suas características de aproximadores funcionais universais (Cybenko, 1989) e sua capacidade de aprender mapeamentos complexos por meio de exemplos.

As redes neurais artificiais (RNA) são importantes instrumentos para a aplicação na área de controle de sistemas não-lineares dadas suas características de mapeadores universais e capacidade de aprender por treinamento (Lippmann, 1987). Aplicações envolvendo controle adaptativo através da linearização do sinal de controle para sistemas de uma entrada e uma saída (Chen & Khalil, 1995), modelagem de sistemas NARMAX (Wang et al., 1994) e Controle por Modelo Interno (Hunt & Sbarbaro, 1991) são alguns dos muitos exemplos de utilização de RNA em controle de processos. As RNAs têm se mostrado superiores aos controladores adaptativos convencionais lineares e não-lineares quanto à precisão e à robustez do sistema em malha fechada (Narendra, 1995).

Sistemas dinâmicos não-lineares são de grande interesse para a engenharia de controle, dada a sua importância na representação matemática de inúmeros sistemas físicos. Devido à complexidade de tais sistemas é difícil desenvolver técnicas abrangentes para o projeto

de controladores. Atualmente diversas técnicas são utilizadas em controle não-linear. As técnicas usuais baseiam-se no cancelamento das não-linearidades (Slotine & Weiping, 1991), que apresenta como principal limitação o fato de necessitarem de um modelo perfeito da planta e as mesmas possuem dinâmica inversa estável. Outro método importante, bastante pesquisado em controle de processos é denominado controle interno (Garcia e Morari, 1982). E por fim, a técnica denominada modelo aproximado NARMA-L2 (Nonlinear Autoregressive-Moving Average) representa o sistema proposto por Narendra e Mukhopadhyay, 1997, a qual foi utilizada neste trabalho para implementar o controle neural do processo de quatro tanques acoplados do Laboratório de Automação e Visão de Sistemas Inteligentes – LAVSI, em simulação e no processo.

1.2 - MOTIVAÇÃO

O trabalho experimental desenvolvido nesta tese foi realizado na planta de quatro tanques acoplados do Laboratório de automação e Visão Inteligente – LAVSI. Nesta planta o problema de controle consiste em regular o nível de um tanque atuando na vazão de entrada de outro tanque. O sistema tem uma dinâmica nitidamente não-linear, pois as vazões nos tanques dependem aproximadamente da raiz quadrada das alturas dos níveis de água. Estas não-linearidades fazem com que o desempenho dos controladores clássicos PID dependa fortemente da faixa de operação do processo, exigindo, assim, ajustes dos seus parâmetros para cada ponto de operação.

Os controladores neurais são projetados em microcomputadores e os resultados são transferidos para o dispositivo final de controle. A escassez de trabalhos na área, utilizando o controlador NARMA-L2, experimentados juntamente com a planta foi a motivação principal deste trabalho, já que microcomputadores são largamente usados em ambientes acadêmicos, comerciais e industriais.

1.3 - EVOLUÇÃO DOS EXPERIMENTOS NO LAVSI

O processo de nível de líquidos utilizado atualmente para experimentos no Laboratório de Automação, Visão e Sistemas Inteligentes é resultado de uma evolução a partir dos sistemas desenvolvidos em projetos anteriores.

O primeiro projeto utilizado no LAVSI (Laboratório de Automação, Visão e Sistemas Inteligentes) consistia de três tanques interconectados por válvulas e alimentados por duas

moto-bombas proporcionais, que por sua vez são acionadas por um *driver* de potência comandado por um sinal de tensão. O nível de líquido é monitorado por sensores elétricos (potenciômetros) acoplados a bóias por meio de uma haste metálica. (Gosmann, 2002). O segundo projeto, de dois tanques, utilizado nos experimentos de dois tanques interconectados por válvulas e alimentado por moto-bomba DC (Tamayo e Mendes 2004, Ramos e Wense 2008). O terceiro projeto de sistema utilizado nos experimentos consiste de quatro tanques interconectados por válvulas e alimentados por quatro moto-bombas proporcionais, que por sua vez são acionadas por um *driver* de potência comandado por um sinal de tensão. O nível de líquido é monitorado por sensores elétricos de pressão.

1.4 - OBJETIVOS

O primeiro objetivo deste trabalho é pesquisar, implementar e avaliar a técnicas de controle não linear NARMA-L2 via redes neurais artificiais em processos de nível de líquido acoplados.

A principal contribuição deste trabalho é estudo de avaliação de controle neural de um processo não linear, onde a rede neural é treinada tanto com dados de simulação quanto com dados do processo. O processo simulado permite avaliar a região de operação e treinar a rede neural com um “modelo nominal”. Este passo intermediário aumenta a chance de sucesso ao se treinar a rede neural com dados do processo. O modelo de simulação é sempre uma simplificação de realidade e assim se justifica o uso de redes neurais, pois estas procuram capturar a real dinâmica do processo.

O segundo objetivo é a validação experimental da metodologia proposta utilizando como aparato experimental o processo de nível de líquidos de quatro tanques acoplados configurável do LAVSI. A abordagem foi implementar o controlador em computador comercial de uso geral em ambiente Matlab/Simulink.

1.5 - METODOLOGIA

Como foi mencionado, o processo estudado é não-linear, apresentando quatro variáveis de estados. Vários controladores são implementados (PID, Neural, *Gain-Scheduling* e NARMA-L2). Para tanto foi proposta a seguinte metodologia:

1. Obter analiticamente um modelo não-linear do processo. Os parâmetros deste serão obtidos experimentalmente;
2. Definir as especificações de desempenho do controlador;
3. Projetar os diferentes controladores para o processo de quatro tanques acoplados;
4. Implementar um controlador NN-NARMA-L2 baseado no aprendizado do processo simulado;
5. Para cada controlador projetado, realizar testes no processo, semelhantes aos experimentos simulados;
6. Implementar o controlador NARMA-L2 com dados do processo;
7. Comparação crítica dos resultados.

1.6 - ORGANIZAÇÃO DESTE TRABALHO

O segundo capítulo descreve a planta de quatro tanques acoplados do Laboratório de Automação, Visão e Sistemas Inteligentes, para estudos de diferentes estratégias de controle de um processo multivariável não-linear, discutindo os aspectos do hardware envolvido para o controle.

O terceiro capítulo descreve o controle convencional no processo de quatro tanques acoplados apresentado no segundo capítulo.

O quarto capítulo descreve as redes neurais e o controlador neural para a aplicação em controle de processos. Uma breve descrição é dada sobre os suportes teóricos dos sistemas neurais e sobre os algoritmos utilizados.

O quinto capítulo descreve a pesquisa e projeto de treinamento e implementação do controle neural do processo de nível de líquidos. O controlador neural foi implementado em um computador comercial de uso geral.

O sexto capítulo descreve o controlador NN-NARMA-L2 (*Nonlinear Autoregressive-Moving Average*), um arranjo da rede neural através do modelo da planta, que é treinado *offline*. O controlador procura transformar a dinâmica não-linear do sistema em uma dinâmica linear no cancelamento das não linearidades (Narendra,1995).

O sétimo capítulo descreve a implementação, identificação, treinamento e o controle da planta por meio do controlador NN-NARMA-L2 do processo de nível de líquido e resultados. Implementações do controle neural no modelo para acelerar o processo da pesquisa, identificando o modelo, e posteriormente aplicado na planta, identificando a planta, por meio de computador comercial de uso geral, utilizando o software Matlab/Simulink.

O oitavo capítulo apresenta conclusões e sugestões para a implementação de controle neural em processo de tanques acoplados.

1.7 - FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica desta tese e a revisão bibliográfica são pautadas em literatura recente, onde são abordados o estado da arte, trabalhos e métodos anteriormente descritos para o controle de processo baseado em controlador neural e controlador NN-NARMA-L2 em um processo tanques de água acoplados, mostrando contribuição significativa e nova, na literatura acadêmica.

Na revisão bibliográfica realizada, nos artigos relacionados ao controlador neural e o controlador NARMA-L2 para a identificação de sistemas dinâmicos nota-se grande interesse de pesquisadores, principalmente para o controle de sistemas não-lineares. Técnicas de controle clássicas são baseadas em modelos linearizados dos sistemas físicos, o que representa perda de informações, que muitas vezes são importantes para o funcionamento do sistema com altos níveis de exigência. Atualmente, a utilização de diferentes técnicas de controle inteligente tem aberto uma nova perspectiva no tratamento de sistemas não-lineares e no projeto de seus controladores. Diversos artigos foram estudados mostrando excelentes resultados utilizando as redes neurais no processo de identificação de não linearidades para controle. Porém, nota-se que a maioria dos controles NARMA-L2, encontrados nessa revisão bibliográfica se limitam a simulações e não ao controle real do processo físico, como realizado nesse trabalho.

Apresentam-se neste trabalho as pesquisas e implementação da identificação utilizando o controle NN-NARMA-L2, com diferentes algoritmos, que se mostraram serem eficazes para identificar as não linearidades presentes no processo de tanques acoplados. Assim, pode-se dizer que cada estrutura de controle tem seu campo de aplicação em função do requerimento de controle da planta e em função das características da planta a controlar.

Assim, na análise dos resultados obtidos, utilizaram-se o algoritmo de retropropagação e o controlador neural NN-NARMA L2. As respostas obtidas são apresentadas e discutidas.

1.8 - REVISÃO DA LITERATURA

Apresenta-se aqui uma revisão bibliográfica do conjunto de técnicas utilizadas no controle de processos de nível, bem como comentários e o estado da arte em relação ao controle de processos de tanques acoplados.

O trabalho de Mokri et al. (2008) são discutidas a identificação do modelo de um manipulador e o controle desse manipulador utilizando o controlador NARMA-L2. O algoritmo de controle foi implementado em Matlab/Simulink, utilizando o toolbox *xPC*, via protocolo de comunicação RS232.

O trabalho de Freitas (2007) apresenta aplicações de técnicas de controle preditivo linear em sistemas não lineares multivariáveis, baseada na linearização exata adaptativa. Aplicaram-se técnicas de controle linear como PID, Espaço-de-Estados, Controle Ótimo e Controle Preditivo, em um sistema multivariável de nível de líquidos, um processo de quatro tanques, composto por duas entradas e duas saídas.

O trabalho de Lopes et al. (2006) é discutida a implementação de uma estrutura de controle neural que identifica e controla um sistema hidráulico (um controle neural aplicado a um sistema de tanques acoplados utilizando controlador lógico programável). Foi utilizada uma estrutura de rede neural convencional implementada em linguagem C++ em computador comercial e os sinais de controle transferidos para um controlador lógico programável, o qual controla esse sistema hidráulico. Foi utilizado o sistema supervisório Intouch e a programação na linguagem *ladder logic* do CLP e o RSLogix. Os resultados obtidos com o sistema de controle neural dos tanques acoplados indicaram bom desempenho, o desempenho do controlador neural foi comparado com o desempenho de um controlador PID sintonizado na faixa de operação.

O trabalho de Prapong et al. (2006) são discutidas técnicas de transferência de dados entre o Matlab instalado em computador comercial e o controlador lógico programável, para o controle de processo de radiação tipo raio-x. Foi utilizado o esquema de registro de dados OPC (*OLE for Process Control*) do Matlab conectado com o CLP via RS232C.

O trabalho de Sekhon et al. (2006) são discutidas técnicas de transferência de dados entre o Matlab instalado em computador comercial e o controlador lógico programável, para o controle de processo de uma turbina, envolvendo motor e gerador como carga. Foi utilizado o esquema de registro de dados *OPC server connection* do toolbox do Matlab/Simulink conectado com o CLP Alen Bradley via RSLinx da Alen Bradley.

O trabalho de Mohammad e Farrokhi (2006) é discutido técnicas de controle utilizando o modelo NARMA-L2 para identificar entradas e saídas de modelos de sistemas não lineares.

O trabalho de Lopes et al. (2006) é discutida a implementação do projeto de um sistema para a geração de pontos de operação aplicado a um processo não-linear, utilizando um controle implementado no Controlador Programável interfaceado com um computador. A comunicação entre o Controlador Programável, onde bombas e sensores estão ligados, e o sistema supervisor é realizada através de um canal de comunicação RS232.

O trabalho de Lucena (2005) apresenta-se uma análise da lei de controle baseada em um esquema híbrido indireto usando rede neural, para uma planta de nível de segunda ordem, utilizando dois tanques acoplados. Foram feitas análises matemáticas de estabilidade e diversas implementações, com plantas simuladas e com plantas reais, com a finalidade de se analisar o problema.

O trabalho de Pukrittayakamee et al. (2002) são discutidas técnicas teóricas de controle utilizando o modelo controlador NARMA-L2, para a identificação e controle do modelo do processo levitação magnética (MagLev).

O trabalho de Grega (1999) são discutidas técnicas de implementação e controle utilizando ambiente Matlab/Simulink para o projeto e compilando os comandos obtidos para a implementação em controlador lógico programável, o qual controlará o processo de um sistema de ar condicionado e ventilação. Também se discutem as técnicas de controle utilizando o modelo RTW (Real-Time Windows).

O trabalho de Soares e Nadal (1999) é discutido técnicas de controle aplicadas na área biomédica empregando o controlador NARMA-L2 utilizando o algoritmo Levenberg-Marquardt, o qual mostrou-se mais eficiente para esse trabalho, durante o treinamento da rede neural, necessitando de pouco tempo de processamento.

Nesses artigos publicados, não se encontraram artigos que abordam sobre o tema proposto nesta tese envolvendo juntamente identificação e controle de um processo de quatro tanques acoplados utilizando o controlador neural, o controlador NARMA-L2, e o software Matlab/Simulink, com transferência de dados via toolbox OPC a esse processo. Foram encontrados poucos trabalhos relacionados com o controlador NARMA-L2, porém não abordaram técnicas neurais de controle utilizando vários tanques acoplados. Encontraram-se vários artigos que abordam as técnicas de controle em separado como simulações da planta e controle em plantas de dois tanques acoplados. Assim, o estudo de avaliação de controle neural de um processo não linear composto de quatro tanques de água acoplados entre si, alavanca uma contribuição teórica e prática para o meio acadêmico e profissional.

1.9 - ARTIGOS PUBLICADOS PELO AUTOR DURANTE DO DOUTORADO

Os trabalhos experimentais, pesquisas e estudos iniciaram-se em uma planta de três tanques acoplados, utilizado por Hugo Gosmann (Gosmann, 2002), que resultou-se no artigo “Neural Control of a Non Linear Process With Gain Scheduling” (Oliveira & Bauchspiess, SBRN 2004). Este artigo apresenta a aplicação de controle neural com pesos selecionáveis pelo ponto de operação. Como processo crítico, que eventualmente deve operar por meses sem falhas, é necessário utilizar equipamento robusto. Assim o controlador neural foi implementado em controlador lógico programável (CLP), na linguagem *ladder logic* como ambiente de programação. Utilizando um controlador PID sintonizado em cada um dos quatro pontos de operação (30%, 40%, 50% e 60%), foram coletados dados para treinar uma rede neural, com uma estrutura definida, porém com diferentes conjuntos de pesos. O artigo referenciado apresenta resultados do controle neural e são discutidas as vantagens e restrições da implementação de redes neurais em controladores lógico programáveis.

Outros artigos foram escritos ao longo do doutorado, utilizando a planta de quatro tanques acoplados, para estudos de diferentes estratégias de controle de um processo não-linear, como o artigo “Controle Neural com Pesos Selecionáveis pela Faixa de Operação Aplicado a um Processo de Nível de Líquidos de 4ª Ordem” (Oliveira & Bauchspiess, XVI CBA 2006), que apresenta duas abordagens para o controle neural de 4ª ordem de um conjunto de quatro tanques de água acoplados. O uso de um conjunto de pesos fixos,

obtido para toda faixa de operação é comparado com o escalonamento de pesos obtidos para faixas de operação. Como um processo crítico que deve operar por meses sem falhas, é necessário usar equipamento robusto. O controlador neural foi implementado em um controlador lógico programável (PLC) usando a programação em linguagem de contatos. Usando um controlador PID, cinco pontos de operação foram sintonizados (10%, 20%, 30%, 40% e 50%), os dados foram coletados para treinar a rede neural. A estrutura da rede neural foi projetada e cinco conjuntos de pesos diferentes foram obtidos do treinamento, para cada ponto de operação. Os resultados dos controladores neurais implementados e as vantagens e restrições da implementação de redes neurais em CLP's são discutidos.

O artigo "Controle Neural de um Processo de Nível de 4ª Ordem" (Oliveira & Bauchspiess, VIII SBAI 2007), apresenta o controle neural NN-NARMA-L2 desse processo de 4ª ordem não-linear. Este processo não linear pode ser configurado de diferentes formas produzindo sistemas com tempos de resposta muito diferentes, dependendo do ponto de operação, sendo uma plataforma de teste para algoritmos de controle avançados. Neste artigo o treinamento foi realizado com os seguintes algoritmos: "*Conjugate Gradient Backpropagation with Powell-Beale Restarts*", "*Conjugate Gradient Backpropagation with Fletcher-Reeves Updates*" e "*Conjugate Gradient Backpropagation with Polak-Ribiere Updates*". A escolha do número de neurônios por camada e os objetivos de erro são discutidos. Os resultados experimentais para as diferentes condições de treinamento são apresentados e avaliadas.

Já o artigo "Neural Control Of a 4rd Order Level Process" (Oliveira & Bauchspiess, 19º COBEM 2007), apresenta um controle neural NARMA-L2, também, desse processo de 4ª ordem não-linear. O desenvolvimento e a execução do controlador proposto foram feitos usando Matlab® e Simulink®. O controlador neural foi executado usando o NN-NARMA-L2. A transferência dos sinais do controle e da medida entre computador (Matlab/Simulink) para os módulos analógicos de entrada e de saída foi realizada pelo toolbox OPC, através de um interfaceador de comunicação instalado no computador, via protocolo de comunicação RS232, entre o microcomputador e o controlador programável, responsável pelo funcionamento dos módulos analógicos. Este artigo foi selecionado pela Associação Brasileira de Engenharia e Ciências Mecânicas (ABCM) e publicado na ABCM Symposium Series in Mechatronics - Volume 3, em 2008.

2 - O PROCESSO DE QUATRO TANQUES ACOPLADOS

2.1 - INTRODUÇÃO

Em todo o trabalho experimental, desenvolvido nesta tese de doutorado, utilizou-se a planta de quatro tanques acoplados do Laboratório de Automação, Visão e Sistemas Inteligentes (LAVSI), para estudos de diferentes estratégias de controle de um processo não-linear. A seguir são discutidos os aspectos do *hardware* envolvido que é importante para as várias fases do desenvolvimento e simulações dos controles neurais propostos nesse trabalho.

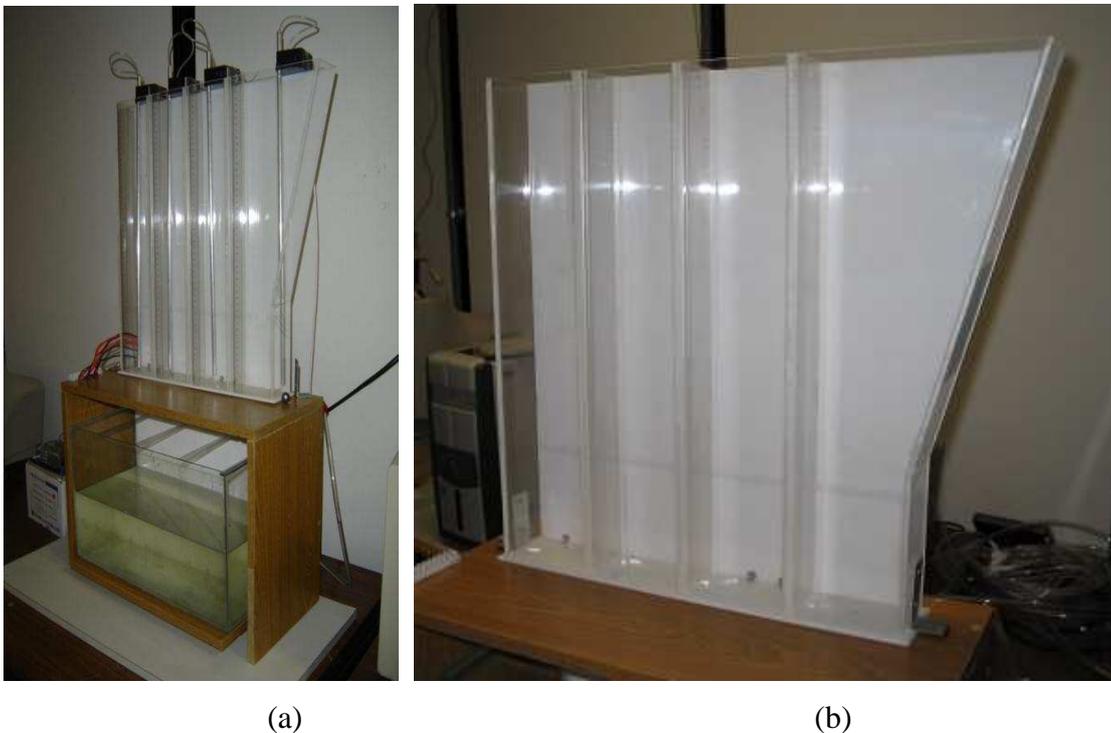


Figura 2.1 - Duas vistas do processo de nível de líquidos.

2.2 - DESCRIÇÃO DA PLANTA

A planta de tanques acoplados em estudo consiste de quatro tanques de água, como ilustram as Figuras 2.1.(a) e 2.1.(b)

Três desses tanques possuem dimensões de 49,5x10x6 cm e o quarto tanque tem secção variável. As secções transversais dos tanques 1, 2 e 3 são de 60 cm². A largura do tanque 4 a partir de 14,7 cm de altura, e que representa 29% do nível de líquido, passa a aumentar linearmente até 20,2 cm de largura (31,65°) com área da secção transversal passando de 60 cm² para 121,2 cm².

O tanque 1 possui entrada de água, que é bombeada do reservatório para este tanque por meio de uma moto-bomba. O tanque 2 possui uma saída de água, q_{o2} , através de um furo de 4 mm de diâmetro, situado no fundo desse tanque para o reservatório. O tanque 4 possui uma saída de água, q_{o4} , através de um furo de 4 mm, situado no fundo desse tanque para o reservatório.

Entre os compartimentos existe uma ranhura de 2 mm de largura com altura ajustável, que determina o parâmetro k (ver eq. 2.2 adiante), de interconexão entre os tanques. A moto-bomba é acionada por meio de um sistema de potência comandado a partir de uma tensão entre 0 e 10 Vcc. Cada tanque é dotado de um sensor de nível (pressão), figura 2.6, para a medição da coluna de líquido. A medição do nível por meio de sensores de pressão específicos de nível de líquido, aumenta a confiabilidade e a precisão em relação à medição anterior de nível feita com potenciômetro. A dinâmica do processo é nitidamente não-linear, uma vez que as vazões nos tanques dependem da raiz quadrada das alturas dos níveis de água.

Toda a estrutura física do sistema (tanques, sensores e atuadores), para este trabalho, foram montadas no ambiente do Laboratório de Automação, Visão e Sistemas Inteligentes - LAVSI. Foram desenvolvidos também os projetos da eletrônica de condicionamento de sinal dos sensores e da eletrônica de potência para acionamento dos atuadores. Os controladores serão implementados por meio de uma ferramenta computacional, em ambiente Matlab/Simulink.

2.3 - MODELAGEM MATEMÁTICA

Através da vazão de entrada no tanque 1 via coeficientes de aberturas das respectivas válvulas k_{12} , k_{23} e k_{34} , podemos controlar a vazão entre os tanques 1, 2, 3 e 4, respectivamente.

O modelo teórico deste processo pode ser obtido utilizando princípios de conservação (Aguirre, 2000): A vazão volumétrica de água que entra subtraída da vazão que sai é igual à variação acumulada no tanque.

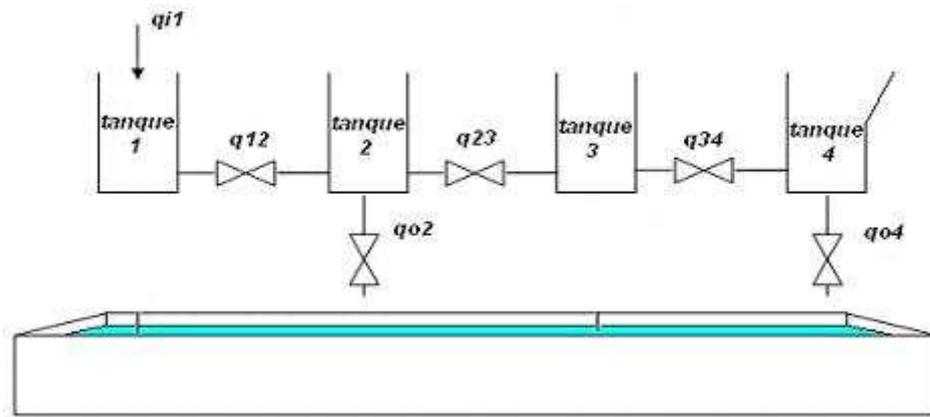


Figura 2.2 - Representação esquemática da planta de tanques acoplados.

No processo de tanques acoplados mostrado na figura 2.2, as vazões e as alturas foram assim definidas:

- q_{i1} = entrada no tanque 1 em $[\text{cm}^3/\text{s}]$;
- q_{12} , q_{23} e q_{34} = vazões entre os tanques 1-2, 2-3 e 3-4 em $[\text{cm}^3/\text{s}]$;
- q_{o2} e q_{o4} = vazões de saída nos tanques 2 e 4, através dos furos de 4 mm em $[\text{cm}^3/\text{s}]$;
- h_1 , h_2 , h_3 e h_4 = altura do nível de água nos tanques 1, 2, 3 e 4 em $[\text{cm}]$;
- A_1 , A_2 , A_3 e $A_4(h_4)$ = são as áreas das seções transversais dos tanques 1, 2, 3 e 4 respectivamente em $[\text{cm}^2]$;
- k_{12} , k_{23} e k_{34} , são os parâmetros das válvulas entre os tanques em $[\text{cm}^{2,5}/\text{s}]$;
- k_2 , e k_4 , são os parâmetros da saída pelos furos em $[\text{cm}^{2,5}/\text{s}]$

A modelagem de um sistema de nível de líquido deve ser realizada utilizando algumas simplificações. Dentre elas pode-se considerar que a água é incompressível e que a perda de carga nos dutos que transportam a água é nula. Estas simplificações são também aplicadas na literatura, como visto em (Ogata, 2003).

Supondo que há conservação da massa, pois todas as entradas e saídas de água nos tanques fazem parte do modelo, pode-se dizer que a variação no volume de um tanque em certo intervalo de tempo é igual à soma das vazões de entrada menos as vazões de saída, podendo ser representado pela expressão da equação 2.1, considerando a área constante (Eq. 2.8):

$$\frac{dv}{dt} = A \frac{dh}{dt} = \sum q_i - \sum q_o \quad (2.1)$$

Em que v é o volume do tanque, A é uma constante, representando a área da secção transversal do líquido, h é a altura do nível [cm], q_i [cm³/s] representa a vazão de entrada de líquido e q_o [cm³/s] as vazões de saída.

2.3.1 - PROCESSO DE UM TANQUE

Outra importante relação a ser observada para fluxo turbulento, que vem da equação de Bernoulli (Ogata, 2003), determina a vazão instantânea de saída de líquido, dada pela seguinte expressão:

$$q = k \cdot \sqrt{h} . \quad (2.2)$$

Em que q é a vazão de saída de líquido [cm³/s], o coeficiente k [cm^{2,5}/s] depende de diversos fatores, como a aceleração da gravidade e da válvula de passagem e h é a altura no nível do tanque com relação à válvula.

A natureza de um escoamento, isto é, se laminar ou turbulento e sua posição relativa numa escala de turbulência é indicada pelo número de Reynolds (Re). Para o número de Reynolds maior que 4000 aproximadamente, a condição de fluxo torna-se turbulenta. Para o escoamento nos quatro tanques acoplados, verificou-se turbulento.

Formulando-se o balanço, para vazão volumétrica de todos os tanques chega-se às seguintes equações:

$$A_1 \frac{dh_1}{dt} = q_{i1} - q_{12} \cdot \text{sign}(h_1 - h_2), \quad (2.3)$$

$$A_2 \frac{dh_2}{dt} = q_{12} \cdot \text{sign}(h_1 - h_2) - q_{o2} - q_{23} \cdot \text{sign}(h_2 - h_3), \quad (2.4)$$

$$A_3 \frac{dh_3}{dt} = q_{23} \cdot \text{sign}(h_2 - h_3) - q_{34} \cdot \text{sign}(h_3 - h_4), \quad (2.5)$$

$$A_4 \frac{dh_4}{dt} = q_{34} \cdot \text{sign}(h_3 - h_4) - q_{o4} . \quad (h_4 \leq 14,7\text{cm}) \quad (2.6)$$

A função $sign(x)$ é definida da seguinte forma:

$$sign(x) = \begin{cases} 1, & \text{se } x > 0 \\ -1, & \text{se } x < 0 \\ 0, & \text{se } x = 0 \end{cases} \quad (2.7)$$

Esta função é utilizada para determinar o sentido das vazões entre os tanques de maior coluna de líquido para a menor. A área A_4 depende do nível h_4 de acordo com a seguinte relação:

$$A_4(h_4) = \begin{cases} 60 \text{ cm}^2, & \text{se } h_4 < 14,7 \text{ cm} \\ \left((h_4 - 14,7) \times \frac{20,2 - 10}{49,5 - 14,7} + 10 \right) \cdot 6 \text{ cm}^2, & \text{se } h_4 \geq 14,7 \text{ cm} \end{cases} \quad (2.8)$$

Como pode ser observado na expressão anterior, com h_4 valendo 14,7 cm, tem-se uma área de 60 cm², e com h_4 igual a 49,5 cm, tem-se uma área de 121,2 cm², como era desejado. Neste trabalho será considerado apenas h_4 menor que 14,7 cm.

A seção transversal dos tubos que são conectados aos sensores de pressão pode ser desprezada por ser inferior a 0,5 cm², não representando uma alteração significativa na área do tanque.

O sinal q_{il} é a vazão fornecida pela bomba que é utilizada para o controle do sistema. Foram realizados experimentos para a calibração da bomba variando a tensão aplicada sobre a mesma, obtendo, desta forma, um modelo de calibração que é utilizado para converter o sinal de controle de vazão para um valor em tensão que será aplicado sobre a bomba.

Para o sistema descrito neste trabalho, as equações que determinam as vazões são as seguintes:

$$q_{o2}(t) = k_{o2} \cdot \sqrt{h_2(t)}, \quad (2.9)$$

$$q_{o4}(t) = k_{o4} \cdot \sqrt{h_4(t)}, \quad (2.10)$$

$$q_{12}(t) = \text{sign}(h_1(t) - h_2(t))k_{12} \cdot \sqrt{|h_1(t) - h_2(t)|}, \quad (2.11)$$

$$q_{23}(t) = \text{sign}(h_2(t) - h_3(t))k_{23} \cdot \sqrt{|h_2(t) - h_3(t)|}, \text{ e} \quad (2.12)$$

$$q_{34}(t) = \text{sign}(h_3(t) - h_4(t))k_{34} \cdot \sqrt{|h_3(t) - h_4(t)|}. \quad (2.13)$$

Substituindo as equações das vazões (2.9 a 2.13) nas expressões de equilíbrio de massa do sistema, obtêm-se o modelo dinâmico completo do processo de tanques acoplados do LAVSI, representado pelas equações:

$$A_1 \frac{dh_1}{dt} = q_{i1} - \text{sign}(h_1 - h_2) \cdot k_{12} \cdot \sqrt{|h_1 - h_2|}, \quad (2.14)$$

$$A_2 \frac{dh_2}{dt} = \text{sign}(h_1 - h_2) \cdot k_{12} \cdot \sqrt{|h_1 - h_2|} - k_{o2} \cdot \sqrt{h_2} - \text{sign}(h_2 - h_3) \cdot k_{23} \cdot \sqrt{|h_2 - h_3|}, \quad (2.15)$$

$$A_3 \frac{dh_3}{dt} = \text{sign}(h_2 - h_3) \cdot k_{23} \cdot \sqrt{|h_2 - h_3|} - \text{sign}(h_3 - h_4) \cdot k_{34} \cdot \sqrt{|h_3 - h_4|}, \text{ e} \quad (2.16)$$

$$A_4 \frac{dh_4}{dt} = \text{sign}(h_3 - h_4) \cdot k_{34} \cdot \sqrt{|h_3 - h_4|} - k_{o4} \cdot \sqrt{h_4}. \quad (2.17)$$

Modelo válido para $h_4 \leq 14,7\text{cm}$.

O diagrama de blocos do processo de tanques acoplados implementado no Simulink, é apresentado na Figura 2.3.

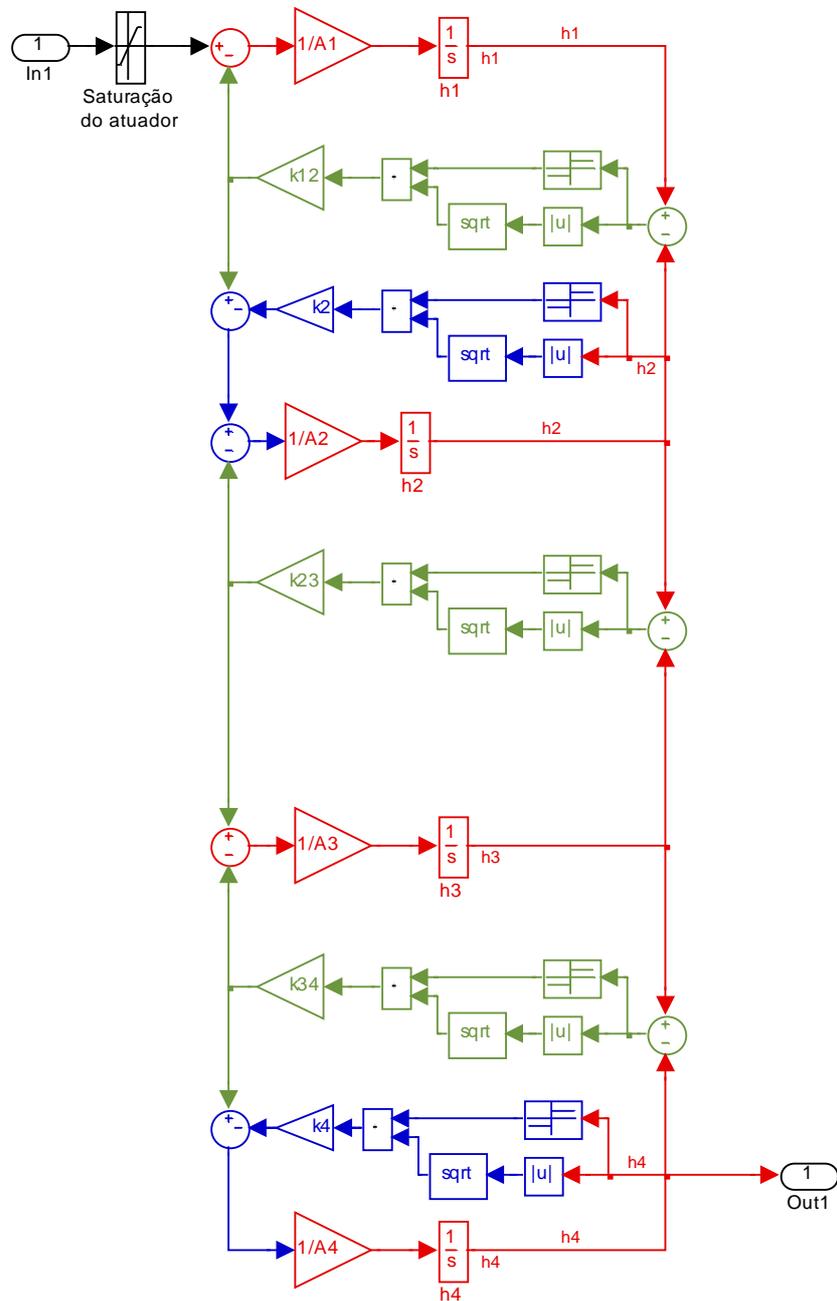


Figura 2.3 - Diagrama de blocos do modelo de tanques acoplados.

2.3.2 - PARÂMETROS DO MODELO

A seguir serão descritos os experimentos realizados para a determinação dos parâmetros que relacionam a vazão com a altura de cada nível, os parâmetros k_{12} , k_{23} , k_{34} , k_{02} e k_{04} descritos anteriormente no modelo do sistema.

Inicialmente foram identificados os parâmetros k_{02} e k_{04} . Para tal, controlaram-se os níveis dos tanques 2 e 4 em aproximadamente 46 cm, mantendo as válvulas entre os tanques 1-2 e 3-4 abertas, com aquela entre os tanques 2-3 fechada. Em seguida estas válvulas foram fechadas permitindo o esvaziamento dos tanques através das válvulas k_{02} e k_{04} . Com estes dados foi possível obter uma aproximação da vazão a cada amostra do tempo.

Relembrando a equação que relaciona a vazão turbulenta do tanque com a sua altura, tem-se:

$$q = k \cdot \sqrt{h}, \quad (2.18)$$

onde q é a vazão de saída, h é a altura de coluna d'água e k é a constante turbulenta.

A equação anterior pode ser reescrita para fornecer k como função de q e h :

$$k = \frac{q}{\sqrt{h}}. \quad (2.19)$$

Outros parâmetros de interesse são os coeficientes que determinam a vazão entre os quatro tanques do sistema. Como estes consistem em válvulas variáveis, foi necessária a calibração dos mesmos em diversas configurações. Os parâmetros de interesse são denominados: k_{12} , k_{23} e k_{34} .

Para a determinação dos mesmos foi utilizada a expressão que determina a vazão entre dois tanques, como mostrado na equação 2.20, para os tanques 1 e 2:

$$q_{12}(t) = k_{12} \cdot \sqrt{|h_1(t) - h_2(t)|}. \quad (2.20)$$

No experimento realizado foi controlado o nível do tanque 1 em um valor, e medida a diferença da altura entre os níveis dos tanques 1 e 2. Foi necessário fechar a saída de água

do tanque 1 (q_{o1}) para que toda a vazão fornecida pela bomba passasse entre os tanques e saísse pelo tanque 2 (q_{o2}). Desta forma, tem-se:

$$q_{i1} = q_{12} = q_{o2}. \quad (2.21)$$

Assim, utilizando a identificação realizada para a vazão da moto-bomba, pode-se obter o parâmetro desejado através da seguinte expressão:

$$k_{12} = \frac{q_{12}(t)}{\sqrt{|h_1(t) - h_2(t)|}}. \quad (2.22)$$

Resta determinar os parâmetros que relacionam a vazão de saída de água nos tanques 1 e 3. A vazão depende da diferença entre a altura do nível do tanque e a altura do nível do reservatório de água.

Um ponto positivo dos experimentos utilizados para a identificação dos parâmetros do sistema é que foi necessária apenas a identificação dos motores para a realização da identificação dos outros parâmetros, evitando, desta forma, a propagação de possíveis erros na estimação dos parâmetros da planta, caso o método utilizado para determinar um parâmetro dependesse dos outros parâmetros identificados.

2.3.3 - SENSORES DE PRESSÃO

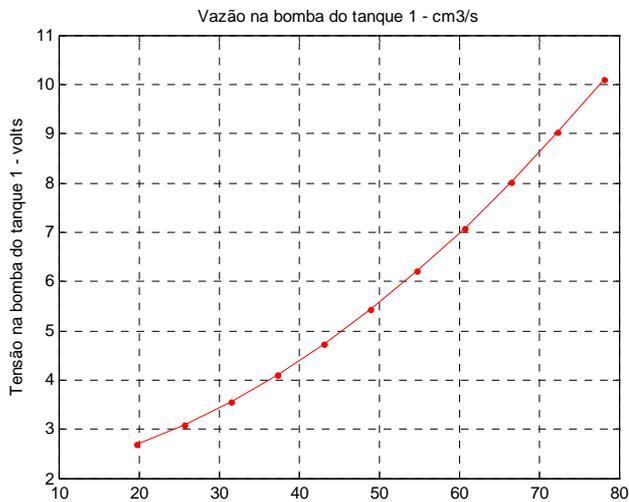
Através da calibração dos sensores de pressão obteve-se o valor do ganho, com *offset* zerado. Testes mostraram o correto funcionamento do sistema de medição (Melo & Bernardes, 2006).

2.3.4 - MOTO BOMBAS

Na figura 2.4 é apresentado o resultado da identificação da moto bomba do tanque 1, realizado experimentalmente indicando os pontos experimentais. A curva obtida foi aproximada por equações do segundo grau que aproximaram satisfatoriamente as vazões. Foi observado um comportamento de zona-morta das saídas para valores de tensão inferiores aproximadamente a 2 Volts e de saturação para valores superiores aproximadamente a 11 Volts e, por isso, estes pontos foram excluídos das curvas para uma melhor aproximação, mas foram utilizados no *software* para converter o sinal de controle de vazão para um valor em tensão que foi aplicado sobre a bomba.

A expressão obtida que relaciona a tensão aplicada sobre a bomba à sua vazão é mostrada na equação 2.23:

$$V_1 = 0,001132q_1^2 - 0,01609q_1 + 1,935 \quad (2.23)$$



a)



b)

Figura 2.4 – Moto bomba: a) Curva de calibração da moto bomba 1, b) moto bomba

2.3.5 - PARÂMETROS DAS VÁLVULAS

Para os parâmetros k_{02} e k_{04} , o experimento consistiu na manutenção dos tanques em um nível alto (46 cm), e em seguida foi permitido que os tanques esvaziassem independentemente dos outros, obtendo-se os valores médios:

$$k_{o2} = 4,31 \text{ cm}^{2,5} / \text{s} , \text{ e} \quad (2.24)$$

$$k_{o4} = 4,96 \text{ cm}^{2,5} / \text{s} . \quad (2.25)$$

Realizados outros experimentos de controle variando a posição das válvulas, obtiveram-se os valores médios para os parâmetros mostrados na Tabela 2.1. Na Figura 2.5 apresenta a válvula implementada com ranhura e engaste.

Tabela 2.1 - Estimativas dos parâmetros k_{12} , k_{23} e k_{34} .

Posição da válvula	$k_{12} [cm^{2,5}/s]$	$k_{23} [cm^{2,5}/s]$	$k_{34} [cm^{2,5}/s]$
1ª engaste da régua	2,88	4,64	5,26
2ª engaste da régua	9,06	11,89	10,09
3ª engaste da régua	12,18	20,62	16,65

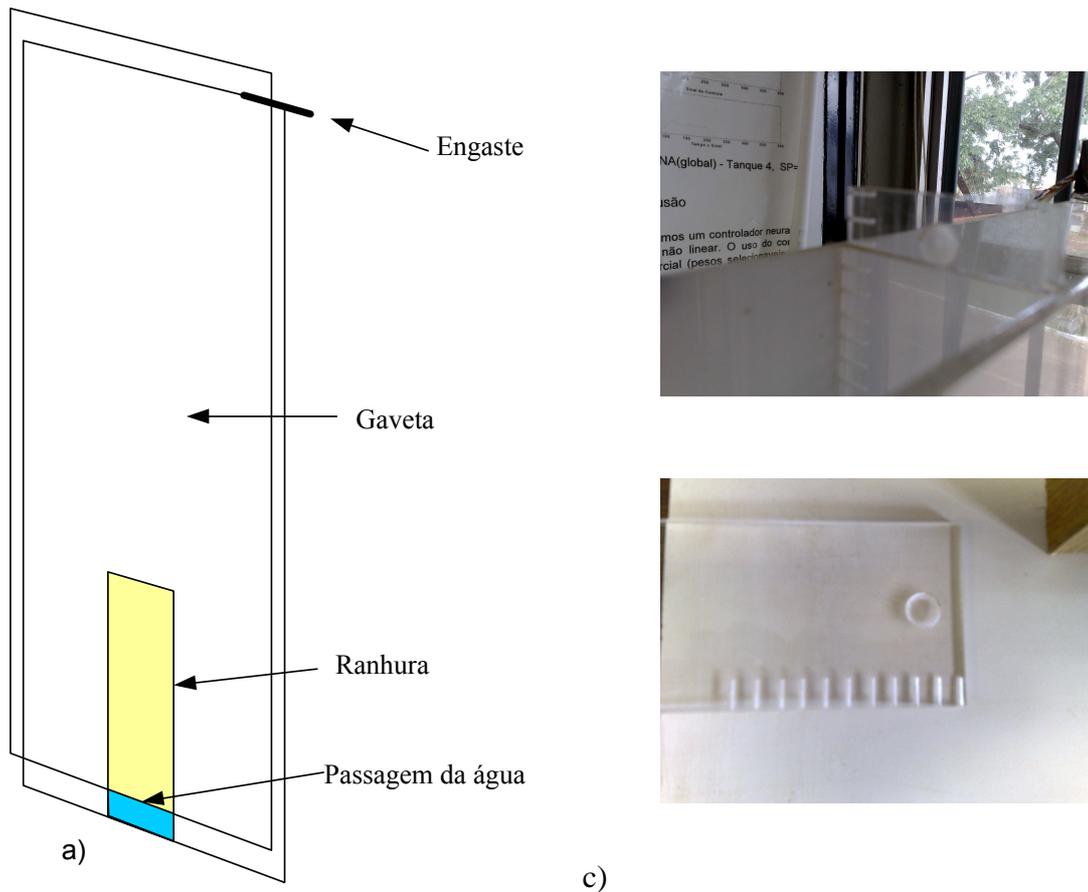


Figura 2.5 – Válvula implementada com ranhura e engaste: a) esquema, b) detalhe do engaste e c) opções de engaste na régua.

2.4 - INSTRUMENTAÇÃO DA PLANTA

2.4.1 - SENSORES DE NÍVEL

Os sensores de nível instalados nos quatro tanques são de pressão. Para a medição de nível de líquido através da pressão, utiliza-se o princípio de que a pressão só depende da densidade do material e da altura do líquido acima do ponto de referência para um líquido

uniforme. Nos quatro tanques acoplados, deseja-se utilizar um método capaz de realizar a medida dos níveis dos líquidos sem a necessidade de inspeção humana. Para estas aplicações podem-se utilizar, por exemplo, sensores piezo resistivos. Este tipo sensor possui vantagens como a alta resolução e linearidade. A Figura 2.6 ilustra a medição de nível e seus circuitos eletrônicos, utilizando o sensor de pressão do fabricante Freescale (modelo MPXM2010GS), que foi utilizado para obter os níveis dos tanques 1, 2, 3 e 4.



Figura 2.6 – a) Medição de nível: a) sensor de pressão, b) condicionamento dos circuitos do sensor de pressão

2.4.2 - MOTO BOMBA – ELEMENTOS FINAIS DE CONTROLE DA PLANTA

O atuador do sistema é uma moto-bomba que funciona com tensão entre 0-12 V. O acionamento das motos-bombas é realizado por meio de um circuito de potência, que é comandado através das saídas analógicas do controlador lógico programável. A Figura 2.7 mostra uma foto deste circuito, (Melo & Bernardes, 2006):



Figura 2.7 – Módulo de acionamento

2.5 - INTERFACES ANALÓGICAS

Neste experimento, a implementação dos controles foi realizada utilizando a interface analógica de entrada, referência 1771-IFE, e a interface de saída, referência 1771-OFE,

ambas da Allen Bradley Products, instaladas em chassi de rack do controlador lógico programável número 5/25, referência 1785-PLC5.

O controlador lógico programável, mostrado nas Figuras 2.8 e 2.9, foi utilizado, também, neste experimento, objetivando a transferência de dados, através dos módulos analógicos, entre o aplicativo Matlab e os módulos sensor de nível e acionamento do processo.



Figura 2.8 - Controlador programável e os módulos analógicos.



Figura 2.9 – Vista do controle e do processo.

3 - CONTROLE CONVENCIONAL DO PROCESSO DE TANQUES ACOPLADOS

3.1 - INTRODUÇÃO

O processo investigado nesta tese de doutorado é um processo não-linear e complexo. A característica deste processo torna-se o uso de controladores convencionais inadequado, justificando o uso dos controladores neurais. A excursão da variável de controle e a possibilidade de transbordamento de água são dificuldades adicionais. O problema de controle é, portanto, o seguimento da referência pelo nível h_4 sem que haja saturação ou transbordo (Bauchspiess, 2006).

Simulações para a planta nominal com os controladores convencionais PI, Espaço-de-Estados (EE) e Espaço-de-Estados com ação integral (EEI) serão comparados nesta seção.

Para o projeto dos controladores convencionais foi adotado como Ponto de Operação (PO) o nível $h_4 = 8$ cm, que corresponde a uma vazão de entrada $q = 35,3$ cm³/s.

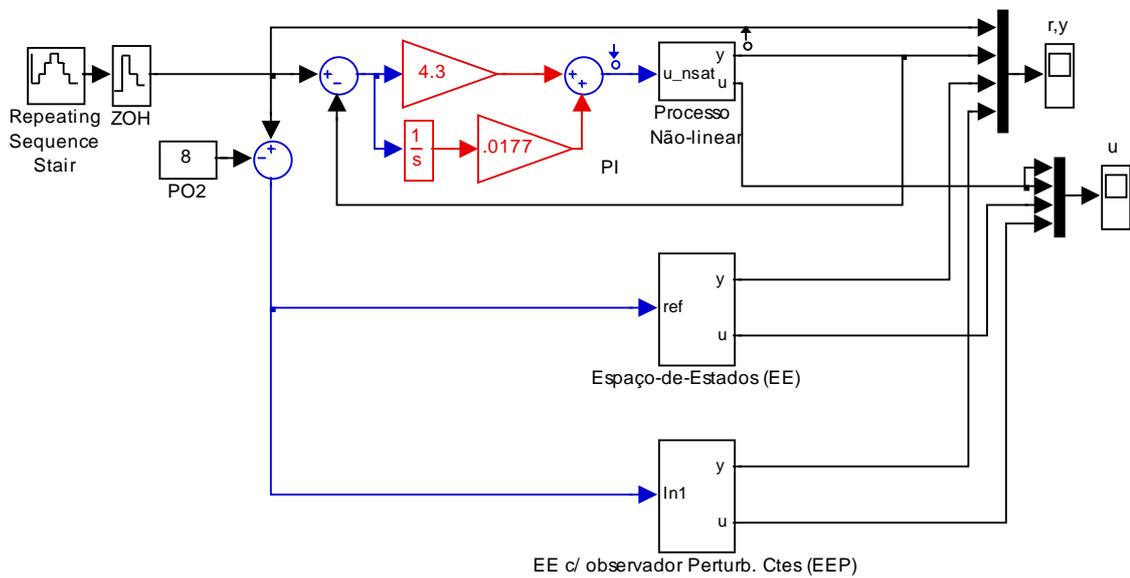


Figura 3.1 – Simulação dos controles PI, EE e EEP.

Na Figura 3.1, apresentam-se os controles PI, EE (Espaço de Estado) e EEP (Espaço de Estados com rejeição de Perturbações constantes), que foram utilizados para simulações. Os blocos EE e EEP são apresentados na Figura 3.2 e 3.3, respectivamente.

3.2 - PROCESSO LINEARIZADO

O processo linearizado em torno do PO de 8 cm (utilizando-se a ferramenta do MatLab® Simulink→Tools→Control Design→Linear Analysis) fornece o modelo no espaço de estados (Bauchspiess, 2006):

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} -0,01959 & 0,01959 & 0 & 0 \\ 0,01939 & -0,0395 & 0,01281 & 0 \\ 0 & 0,01281 & -0,02922 & 0,01641 \\ 0 & 0 & 0,01641 & -0,03105 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0,01684 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \\ y &= [0 \ 0 \ 0 \ 1] \mathbf{x}. \end{aligned} \quad (3.1)$$

Onde, y é a saída h_4 , u é a entrada q_{i1} e x são os níveis h_1 , h_2 , h_3 e h_4 .

Correspondente a função de transferência:

$$\frac{Y(s)}{U(s)} = \frac{6,8643 \times 10^{-8}}{(s + 0,004107)(s + 0,01509)(s + 0,0421)(s + 0,05807)}. \quad (3.2)$$

As constantes de tempo para diferentes pontos de operação são mostradas na Tabela 3.1, observa-se nessa tabela o grau de variação da constante de tempo em relação ao ponto de operação.

Tabela 3.1 – Modelo linearizado para diferentes pontos de operação.

PO(h_4)	Q/[cm ³ /s]	τ_1 /[s]	τ_2 /[s]	τ_3 /[s]	τ_4 /[s]	h_1 (cm)	h_2 (cm)	h_3 (cm)
4 cm	24,96	12,16	16,85	46,92	173,01	19,72	12,14	7,57
8 cm	35,30	17,20	23,83	66,35	244,68	39,44	24,28	15,13
10 cm	39,46	19,23	26,64	74,17	273,51	49,28	30,33	18,91
12 cm	43,23	21,07	29,19	81,26	299,64	59,15*	36,41	22,69

3.3 - CONTROLE PI

O controle convencional PI foi simulado no modelo da planta cancelando-se o pólo mais lento do processo. O ponto de operação 8 cm foi escolhido de tal forma que a dinâmica dominante tivesse 20% de sobrepasso:

$$\frac{U(s)}{E(s)} = 4,3 + \frac{0,0177}{s}. \quad (3.3)$$

Resultados de simulação do controlador PI podem ser vistos nas Figura 3.4 a 3.7.

3.4 - CONTROLE NO ESPAÇO-DE-ESTADOS

O projeto do controlador no espaço-de-estados é feito pela alocação de pólos. A fórmula de Ackermann (Mathworks, 2006), implementada no MatLab permite o cálculo tanto das realimentações de estado como dos parâmetros do observador de estados.

```
r=-.015;  
p=[r -.06 r+i*2*r r-i*2*r];  
po=3*p;  
K=acker(ml.a,ml.b,p)  
L=acker(ml.a',ml.c',po)
```

Onde p é o vetor de pólos desejados em malha fechada, po são os pólos do observador e ml é a matriz linearizada do processo no ponto de operação 8 cm. O parâmetro r foi introduzido para facilitar a escolha, por tentativa e erro, da posição dos pólos em um sistema não-linear com saturações.

Os valores obtidos foram:

$$K = [-0.8528 \quad 3.9167 \quad 5.3018 \quad -5.1637] \text{ e}$$
$$L = [10.4672 \quad 7.5158 \quad 1.0217 \quad 0.1956].$$

O fator de ajuste de ganho ($Nb = (\text{prod}(p)) / gma.\text{num}\{1\}(5)$) permite que o controlador siga uma referência sem erro. O valor encontrado foi $Nb = 14,75$.

O modelo de simulação do controlador no espaço-de-estados pode ser visto na Figura 3.2 (Bauchspiess, 2006).

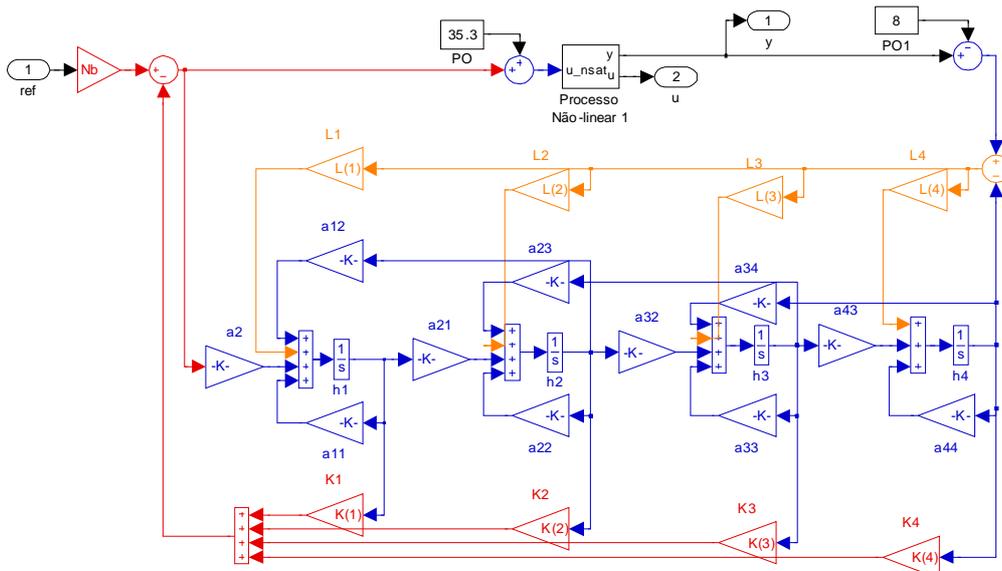


Figura 3.2 – Simulação do controlador EE, vide Figura 3.1 (Bloco EE)

3.5 - CONTROLE NO ESPAÇO-DE-ESTADOS COM AÇÃO INTEGRAL

O controlador no espaço-de-estados é linear, levando a erros sempre que a planta não for bem conhecida, haja perturbações ou tenha elementos não lineares. Para sanar esta deficiência pode ser utilizada uma ação integral. Duas formas alternativas podem ser empregadas: canal integral e observador de perturbações. Aqui a ação integral foi implementada com um observador de perturbações constantes, conforme pode ser vista na Figura 3.3.

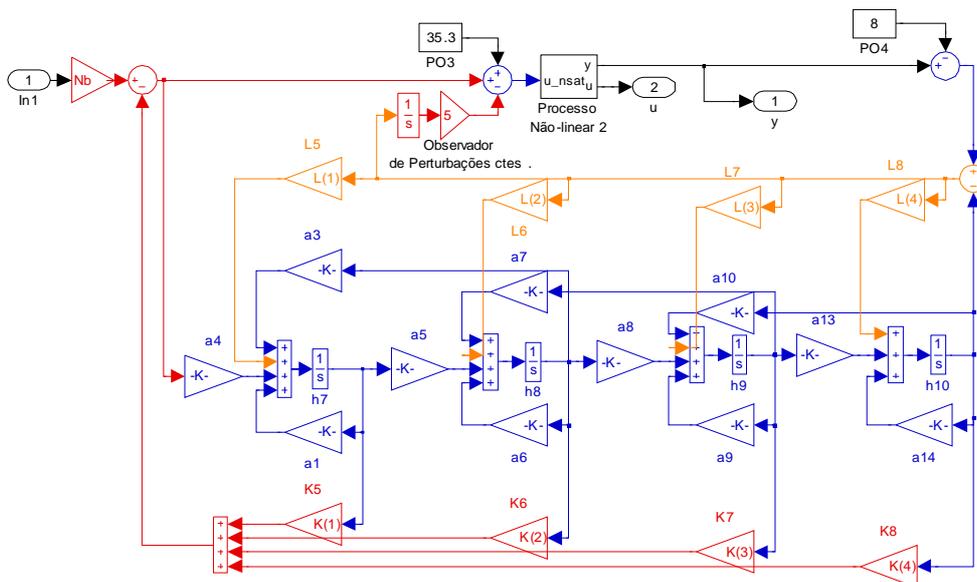


Figura 3.3 – Simulação do controlador EEP, vide Figura 3.1 (Bloco EEP).

3.6 - SIMULAÇÃO DE CONTROLADORES CONVENCIONAIS

Nas Figuras a seguir são apresentadas simulações dos controladores PI, EE e EEI. O propósito é realçar as limitações destes controladores para o controle de processos não lineares. Desta forma, destacar os desafios que serão a utilização de controladores neurais para este processo.

As Figuras 3.4 a 3.7 mostram que os três controladores convencionais aqui apresentados apresentam forte variação das características dinâmicas de acordo com o ponto de operação. O controlador PI é simples e consegue seguir as diferentes variáveis de referências, porém, como apenas o pólo mais lento é cancelado, é muito lento para pontos de operação baixos e apresenta elevado sobrepasso para pontos de operação altos.

O controlador EE apresenta uma boa dinâmica, porém apresenta erro em regime para todos os pontos que se afastem do ponto de operação utilizado em seu projeto.

O controlador EEI permite acompanhar referências e perturbações constantes, no entanto, a sua dinâmica é mais lenta e pontos de operação baixos e oscila mais que o EE em pontos de operação altos.

A expectativa desta tese é de o controlador neural tenha como lidar com as deficiências apontadas.

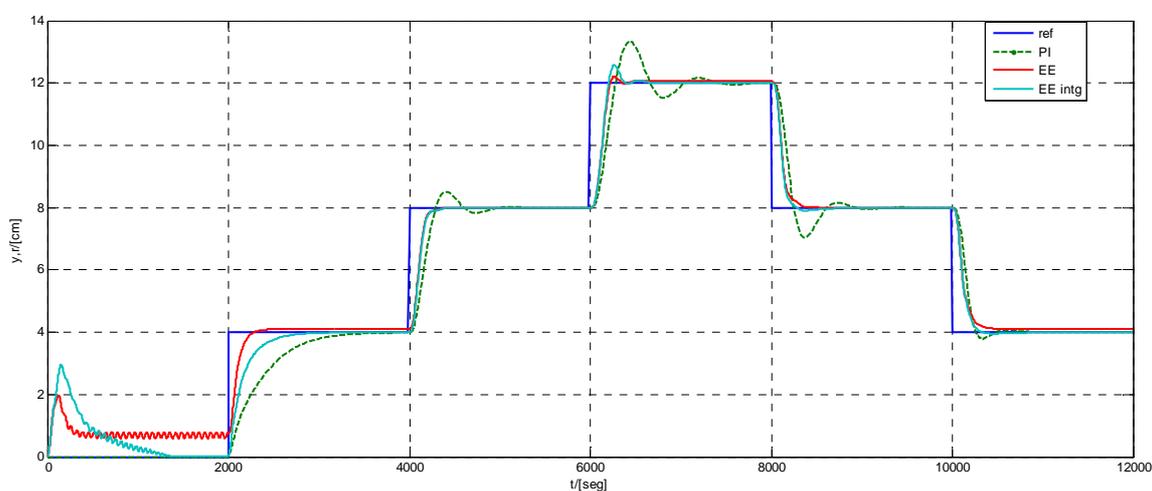


Figura 3.4 – Simulação dos controladores PI, EE e EEI - variáveis de referência e controle

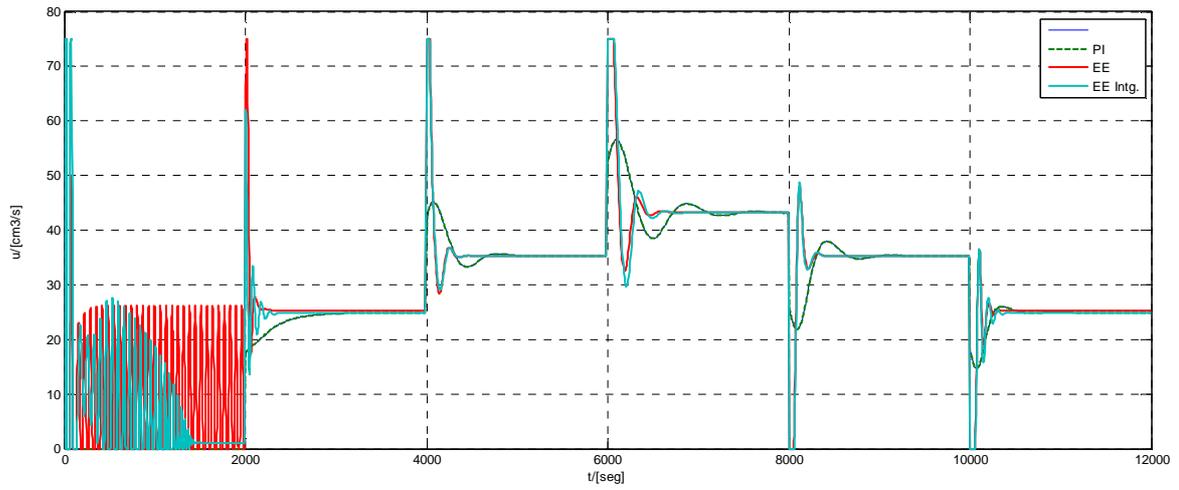


Figura 3.5 – Simulação dos controladores PI, EE e EEI - variáveis de controles

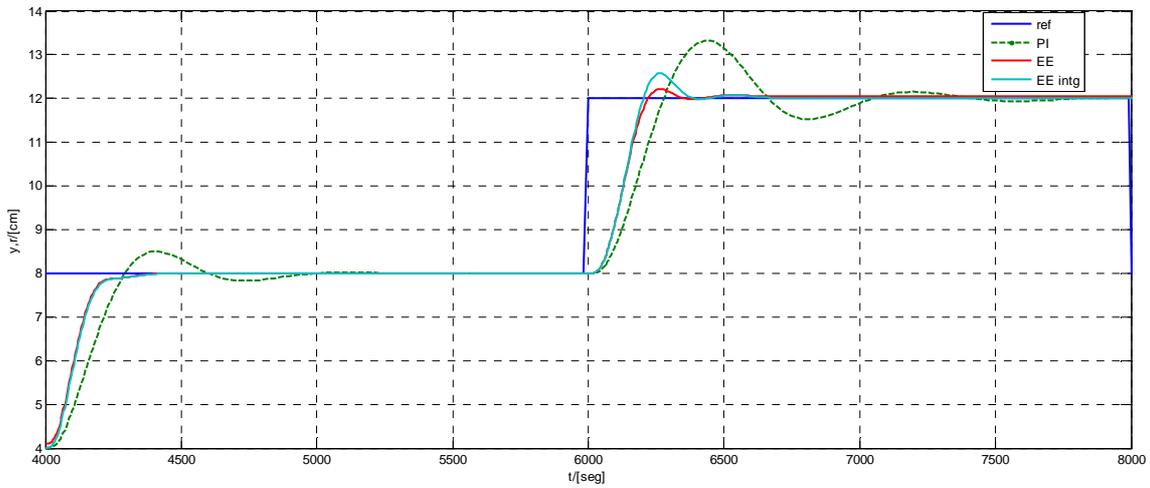


Figura 3.6 – Simulação dos controladores PI, EE e EEI:
referência e variável controlada - detalhe em torno do ponto de operação de 8 cm.

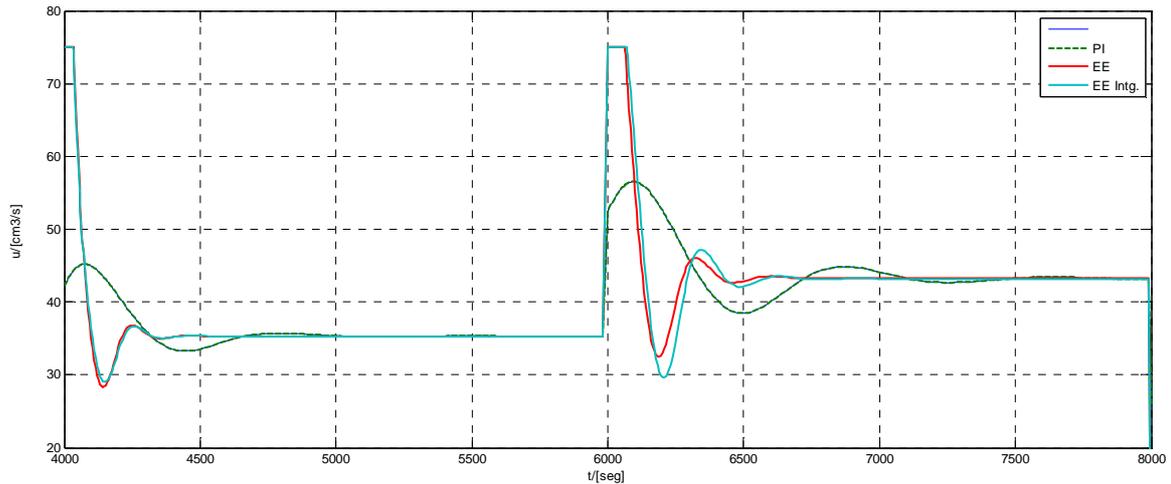


Figura 3.7 – Simulação dos controladores PI, EE e EEI: variável de controle, detalhe em torno do ponto de operação de 8 cm.

Obviamente diferentes sintonias podem ser adotadas nos projetos. Um caso interessante de sintonia PI, simulação apresentado na Figura 3.8 e diagrama de controle apresentado na Figura 3.9, em que uma boa resposta em torno do ponto de operação de 3 cm, leva a oscilações inaceitáveis a partir de 5 cm.

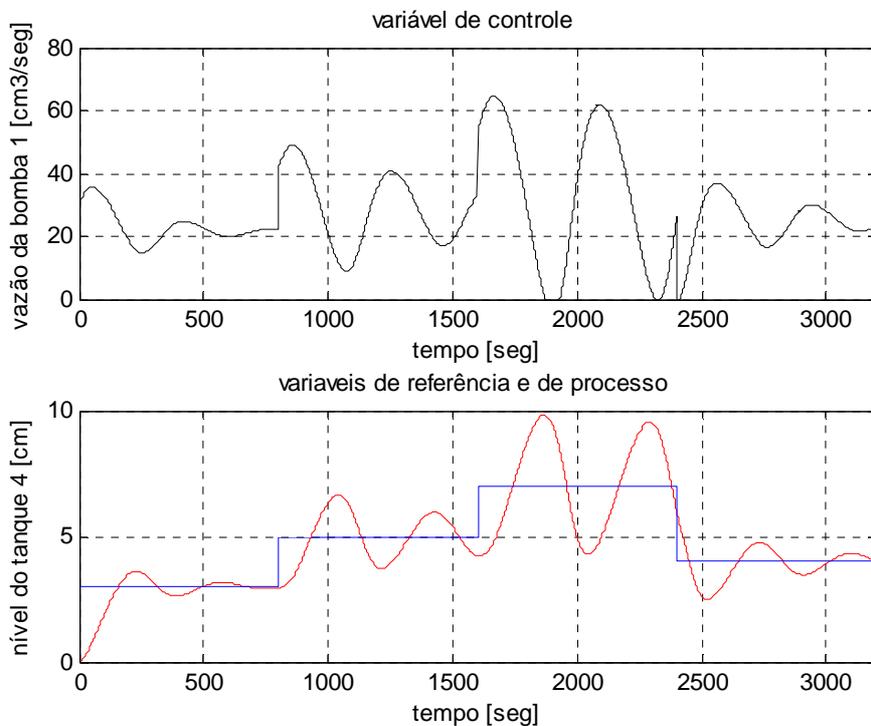


Figura 3.8 – Simulação do controlador PI, variável de referência em torno do ponto de operação de 3 cm

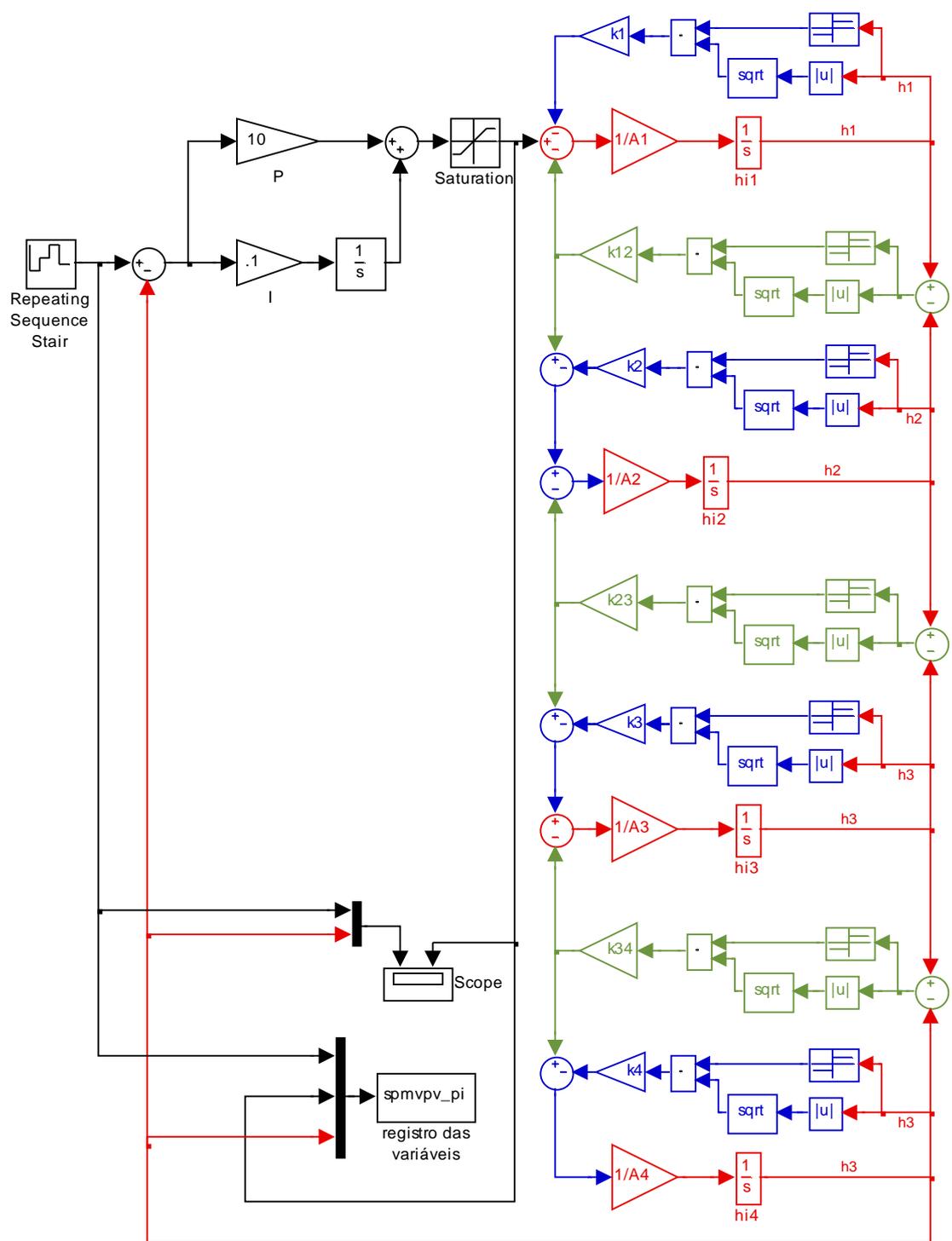


Figura 3.9 – Simulação do controle PI

3.7 - IMPLEMENTAÇÃO DO CONTROLADOR PI NO PROCESSO

Na Figura 3.10 é apresentada a configuração utilizada para a implementação do controlador PI na planta, e na Figura 3.11 são apresentados os sinais de controle na planta de quatro tanques acoplados, com índices de oscilações do tipo *stick-slip*.

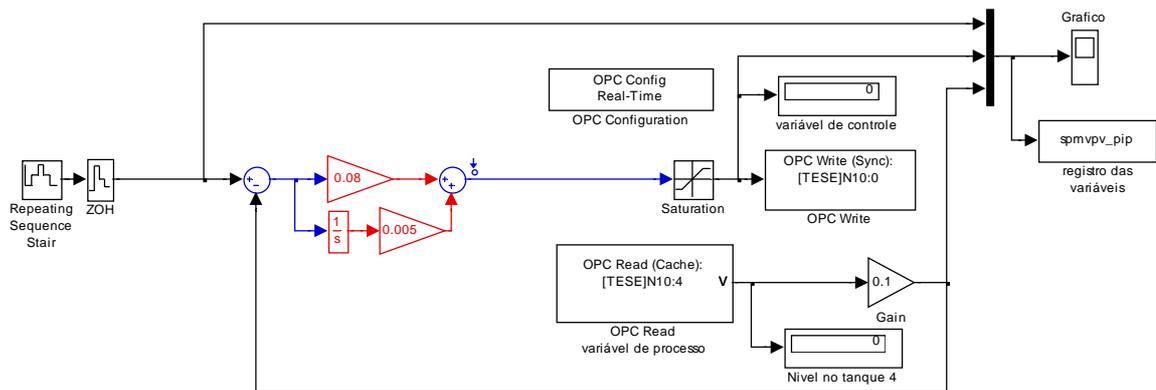


Figura 3.10 – Configuração do controlador PI implementado na planta.

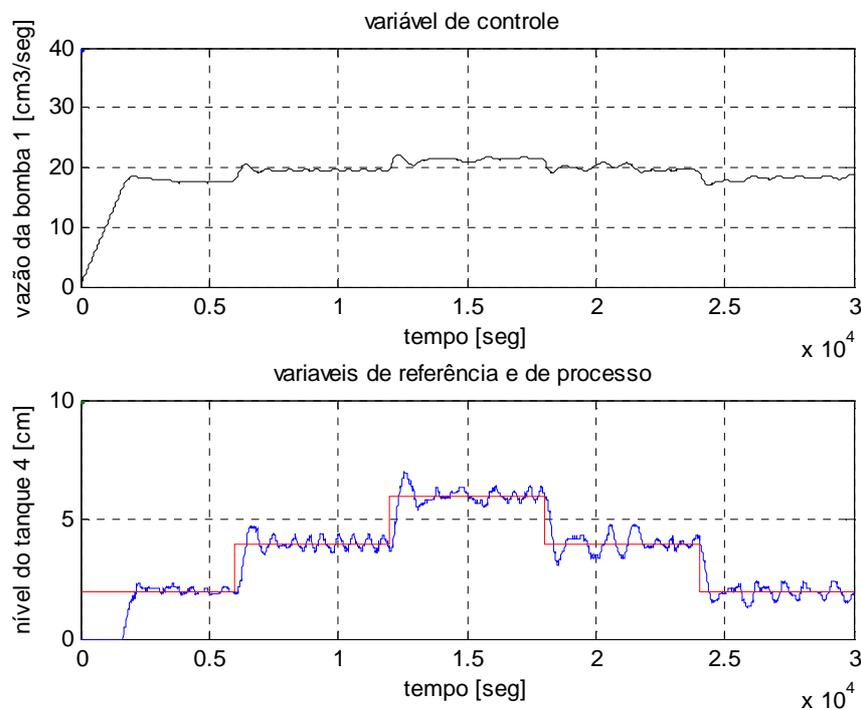


Figura 3.11 – Controle PI na planta

3.8 - CONTROLE NEURAL *GAIN SCHEDULING*

Apenas neste item, apresentam-se resultados obtidos com um processo de três tanques (Gosmann, H. L., 2002), cada um com capacidade de 4500 cm^3 , acoplados por meio de válvulas como mostra a Figura 3.12.

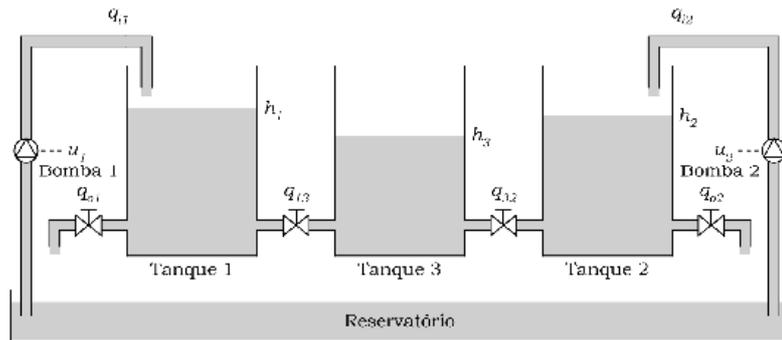


Figura 3.12 - Diagrama da planta de três tanques acoplados

Os Tanques 1 e 2, situados nas extremidades, possuem entradas de água, que é bombeada por meio de duas moto-bombas u_1 e u_2 . Através de manipulação da vazão de entrada no tanque 1 pela abertura da válvula q_{13} e q_{23} , podemos controlar o fluxo da vazão entre os tanques 3 e 2, respectivamente. O problema de controle abordado neste trabalho consiste na regulação da altura do nível do tanque 2 fornecida pelo medidor de nível h_2 , sendo a variável de processo (h_2) e a variável controlada a vazão de entrada no tanque 1 (u_1). A vazão de saída do tanque 2 é ajustada através do fechamento da válvula q_{o2} . Os tanques 1, 2 e 3 têm mesmas dimensões de área transversal de 150 cm^2 , e altura de 30 cm.

As vazões e as alturas foram assim definidas:

q_{i1} e q_{i2} = vazões de entrada nos tanques 1 e 2, respectivamente, [cm^3/s];

q_{13} e q_{32} = vazões entre os tanques 1-3 e 3-2, respectivamente, [cm^3/s];

q_{o1} e q_{o2} = vazões de saída nos tanques 1 e 2, respectivamente, [cm^3/s];

h_1 , h_2 e h_3 = altura do nível de água nos tanques 1, 2 e 3, respectivamente, [cm].

A é a área da seção transversal dos tanques

O modelo dinâmico completo do processo de três tanques acoplados é:

$$A \frac{dh_1}{dt} = q_{i1} + \text{senal} (h_3 - h_1) k \sqrt{|h_3 - h_1|} - k \sqrt{h_1} \quad (3.4)$$

$$A \frac{dh_2}{dt} = q_{i2} + \text{senal} (h_3 - h_2) k \sqrt{|h_3 - h_2|} - k \sqrt{h_2} \quad (3.5)$$

$$A \frac{dh_3}{dt} = - \text{senal} (h_3 - h_1) k \sqrt{|h_3 - h_1|} - \text{senal} (h_3 - h_2) k \sqrt{|h_3 - h_2|} \quad (3.6)$$

O controle neural com *gain scheduling*, Figura 3.13, foi projetado e experimentado na planta de três tanques acoplados, em controlador lógico programável. As Figuras. 3.14 a 3.16 apresentam os sinais na planta.

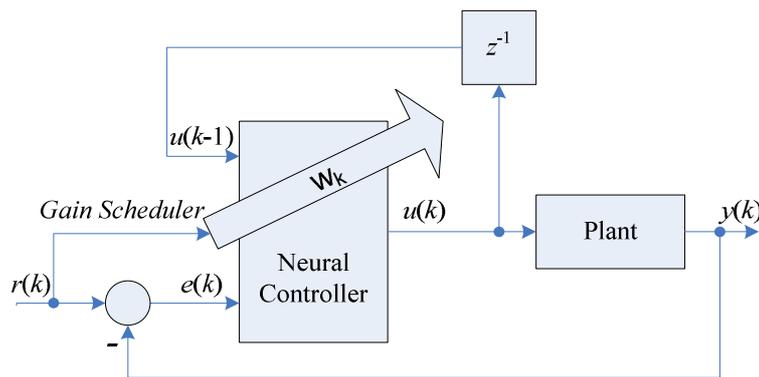


Figura 3.13 – Controle neural *gain scheduling*

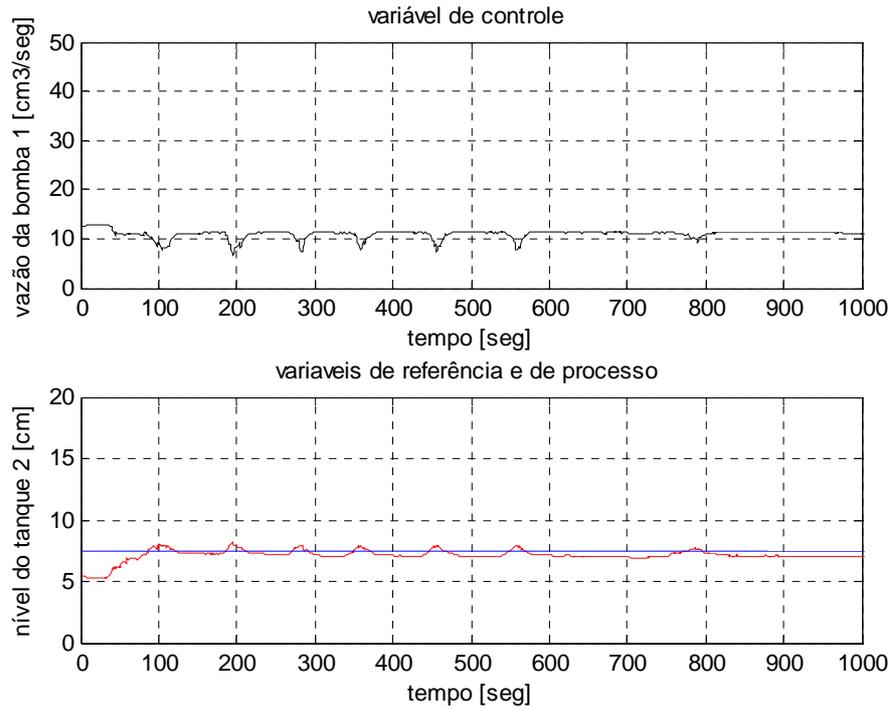


Figura 3.14 – Controle neural *gain scheduling* pesos da RNA para uma variável de referência de 7,5 cm

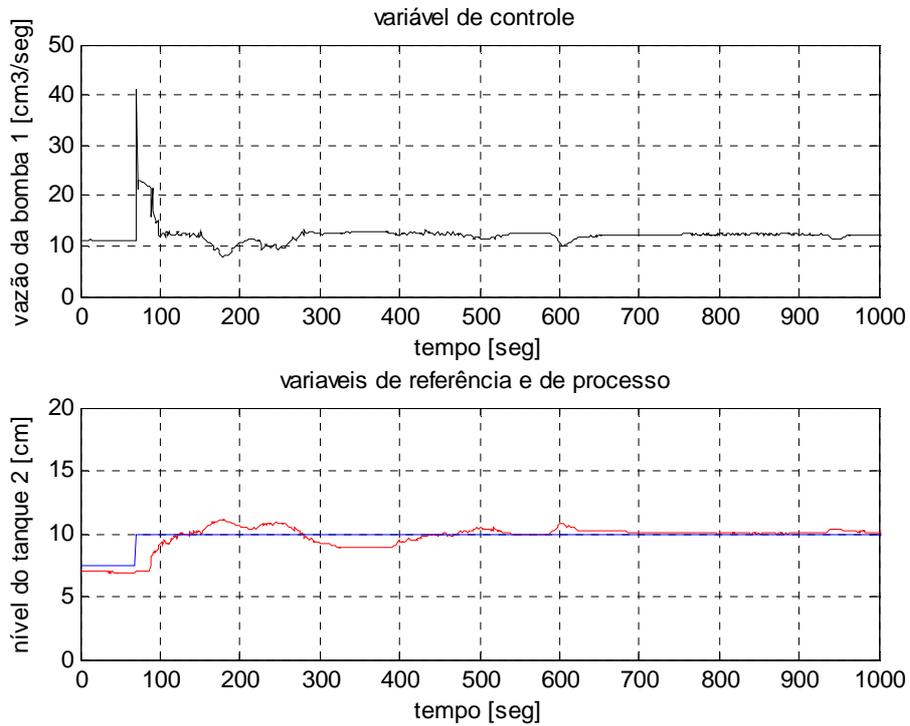


Figura 3.15 – Controle neural *gain scheduling* pesos da RNA para uma variável de referência de 10 cm

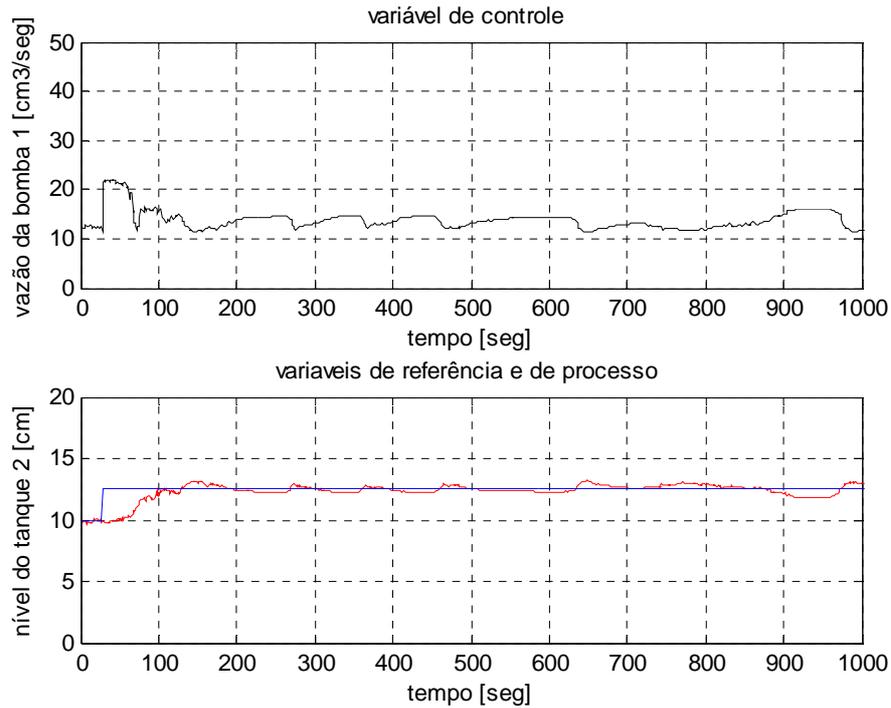


Figura 3.16 – Controle neural *gain scheduling* pesos da RNA para uma variável de referência de 12,5 cm

3.9 - COMENTÁRIOS

Neste item, foram discutidos os aspectos do processo (*hardware*) necessários ao controle. Estes aspectos são importantes para as fases subsequentes do desenvolvimento deste trabalho. Apresentou-se o controle do quarto tanque, por meio da variação da tensão na moto-bomba instalada no primeiro tanque, as estimativas dos parâmetros e a instrumentação da planta. A simulação de controladores convencionais foram apresentados.

Por meio do modelo não-linear da planta de quatro tanques acoplados (equações 2.3 a 2.6), podem-se realizar simulações, que serão mostradas nos capítulos vindouras, para subsidiar a metodologia proposta, objetivando o controle desse processo de quatro tanques acoplados que é não linear.

4 - REDES NEURAIS E CONTROLE NEURAL

4.1 - INTRODUÇÃO

Neste capítulo, são exploradas as potencialidades das Redes Neurais voltadas para as aplicações em controle de processo. Implementação interessante tem sido desenvolvida usando Redes Neurais, como em controle, identificação e processamento de sinais. Uma breve descrição é dada sobre os suportes teóricos dos sistemas neurais e sobre os algoritmos utilizados.

4.2 - REDES NEURAIS MULTICAMADAS

O algoritmo de retropropagação, ou *back-propagation*, para redes de neurônios de multicamadas foi essencialmente inventado e popularizado em 1986 por Rummelhart, Hinton e Williams (Rummelhart et al, 1986). A Figura 4.1 ilustra o diagrama funcional de um neurônio artificial.

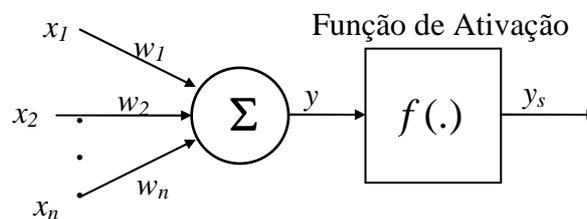


Figura 4.1 - Diagrama funcional de um neurônio artificial.

A operação realizada pelo Neurônio sobre um sinal de vetor $[x_0 \dots x_n]^T$ pode ser expressa por:

$$y_s = f\left(\sum_{k=0}^n x_k w_k\right) \quad (4.1)$$

As redes neurais são aproximadores universais de funções, são capazes de mapear, com precisão arbitrariamente satisfatória, um vetor x do espaço \mathcal{R}^N em um vetor y do espaço \mathcal{R}^M (CybenKo, 1989).

O vetor de pesos $[w_0 \dots w_n]^T$ que multiplica ponto a ponto cada uma das entradas $[x_0 \dots x_n]^T$ e $f(\cdot)$ representa uma função de ativação do neurônio. As diversas camadas ou níveis podem ser obtidos interligando os neurônios, como as saídas de uns neurônios compondo as entradas de outros neurônios.

As redes neurais multicamadas possibilitam a resolução de um espectro maior de problemas através da adição das camadas escondidas, aumentando o espaço de hipóteses que a rede pode representar e possibilitando grande poder computacional.

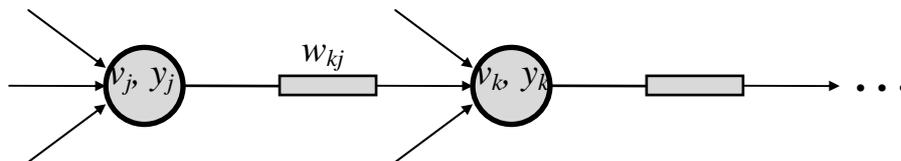
As camadas de uma rede de retroalimentação são:

- camada de entrada (sem processamento);
- camada intermediárias; e
- camada de saída.

4.3 - ALGORITMO DE TREINAMENTO

O Método retroalimentação que foi a técnica utilizada neste trabalho, pertence à família dos métodos *Least Mean Square* (LSM), nos quais se procuram o valor mínimo da função erro quadrático (Freeman, 1991).

Neste item (Braga, 1996), é apresentada a descrição matemática da forma de treinamento da rede *backpropagation*. Abaixo tem-se dois neurônios genéricos, um da penúltima e um da última camada de uma RNA:



O erro quadrático total das saídas da rede, para um dado padrão, pode ser calculado pela expressão abaixo,

$$\varepsilon(n) = \frac{1}{2} \sum_{j=1}^q e_j^2(n), \quad (4.2)$$

onde $e_j = d_j(n) - y_j(n)$ é o erro instantâneo para a saída j , com $d_j(n)$ sendo a saída desejada, $y_j(n)$ é a saída do nodo após a aplicação da função de ativação e q é o número de nodos do nível da saída da rede.

Assim, o erro médio levando em conta todos os N padrões do conjunto de treinamento é dado por:

$$\varepsilon_{ave}(n) = \frac{1}{N} \sum_{n=1}^N \varepsilon(n). \quad (4.3)$$

Aplicando as funções de ativações $f_j(\cdot)$ e $f_k(\cdot)$, para v_j e v_k equivalente às saídas não-lineares, temos:

$$y_j = f_j(v_j) \quad e \quad y_k = f_k(v_k).$$

Para um nodo j qualquer no nível de saída, temos:

$$v_j(n) = \sum_{i=0}^p w_{ji}(n) y_i(n), \quad (4.4)$$

p = número de entradas do nodo ligados ao nível anterior da rede

O gradiente instantâneo do erro quadrático das saídas em relação ao peso é:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (4.5)$$

Para obter a derivada parcial de ε em relação ao peso de cada conexão da penúltima camada com a última camada, temos:

$$\frac{\partial \varepsilon(n)}{\partial e_j(n)} = e_j(n), \quad (4.6)$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1, \quad (4.7)$$

$$\frac{\partial y_j(n)}{\partial v_j(n)} = f_j'(v_j(n)) , \quad (4.8)$$

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) . \quad (4.9)$$

Logo vem:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -e_j(n) f_j'(v_j(n)) y_i(n) . \quad (4.10)$$

Aplicando ao peso w_{ji} a correção no instante n , vem:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} , \text{ e} \quad (4.11)$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) . \quad (4.12)$$

Definindo o gradiente local como:

$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} . \quad (4.13)$$

$$\delta_j(n) = e_j(n) f_j'(v_j(n)) \quad (4.14)$$

No instante de treinamento $n+1$, tem-se o valor do peso w_{ji} do nodo de saída j atualizado como:

$$w_{ji}(n+1) = w_{ji}(n) + \eta \delta_j(n) y_i(n) . \quad (4.15)$$

Atualizando os pesos do nodo j da camada escondida recursivamente em relação ao erro da camada posterior a que está conectada vem:

$$\delta_j(n) = -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} , \text{ e} \quad (4.16)$$

$$\delta_j(n) = \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} f'_j(v_j(n)). \quad (4.17)$$

j = nodo de uma camada escondida

$\mathcal{E}(n)$ = erro da camada de saída

Pois não é possível estimar $e_j(n)$ sem o seu valor desejado da saída $d_j(n)$, não podendo ser obtido diretamente.

Re-escrevendo $\mathcal{E}(n)$ para um índice relativo à camada de saída (k), temos:

$$\mathcal{E}(n) = \frac{1}{2} \sum_{k=1}^K e_k^2(n). \quad (4.18)$$

Obtendo a derivada parcial do erro quadrático do nível de saída em relação à saída $y_j(n)$ no nodo j da camada escondida, vem:

$$\frac{\partial \mathcal{E}(n)}{\partial y_{ji}(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial y_j(n)}, \text{ e} \quad (4.19)$$

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)}. \quad (4.20)$$

Substituindo $e_k(n)$ por $d_k(n) - f_k(v_k(n))$ vem:

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -f'_k(v_k(n)), \quad (4.21)$$

$$v_k(n) = \sum_j w_{kj}(n) y_j(n), \text{ e} \quad (4.22)$$

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n). \quad (4.23)$$

Então, obtemos a derivada parcial do erro da camada de saída em relação ao valor da saída do nodo j na camada escondida, como:

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = - \sum_k e_k(n) f'_k(v_k(n)) w_{kj}(n), \text{ e} \quad (4.24)$$

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = - \sum_k \delta_k(n) w_{kj}(n). \quad (4.25)$$

Obtendo o gradiente $\delta_j(n)$ para um nodo arbitrário j da camada escondida, vem:

$$\delta_j(n) = f'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n). \quad (4.26)$$

Obtendo através dos gradientes locais dos nodos de saída o valor do peso $w_{ji}(n)$ entre o nodo j da camada escondida e o nodo i da camada de entrada, vem:

$$w_{ji}(n+1) = w_{ji}(n) + \eta \delta_j(n) y_i(n). \quad (4.27)$$

Algoritmo de treinamento da *backpropagation*

- 1) Apresentar o vetor de entrada à camada de entrada da rede neural e propagá-lo até a camada de saída;
- 2) Atualizar os pesos das camadas de saída e escondida, propagando o erro de volta às entradas;
- 3) Para os nodos j da camada de saída, atualizar os pesos da seguinte forma:

$$w_{ji}(n+1) = w_{ji}(n) + \eta \delta_j(n) y_i(n), \quad (4.28)$$

onde $\delta_j(n)$ equivale a:

$$\delta_j(n) = e_j(n) f'_j(v_j(n)). \quad (4.29)$$

4) Para os nodos j da camada escondida:

$$w_{ji}(n+1) = w_{ji}(n) + \eta \delta_j(n) y_i(n),$$

onde $\delta_j(n)$ equivale a:

$$\delta_j(n) = f'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n), \quad (4.30)$$

onde k se refere aos nodos da camada de saída.

4.4 - TREINAMENTO DAS REDES NEURAIIS COM RETROALIMENTAÇÃO

Treinar uma rede neural é encontrar um vetor de pesos das conexões que possibilite a rede aproximar uma função qualquer, dentro de uma faixa de precisão estipulada. A rede é treinada utilizando-se subconjuntos de dados, extraídos dos universos de entrada e saída. Após o treinamento, geralmente a rede é capaz de generalizar o que significa que ela é capaz de gerar valores aproximadamente corretos para um vetor de entradas arbitrário que não faz parte do subconjunto de treinamento.

As redes neurais podem ser treinadas segundo os paradigmas de aprendizado (Braga, 2000):

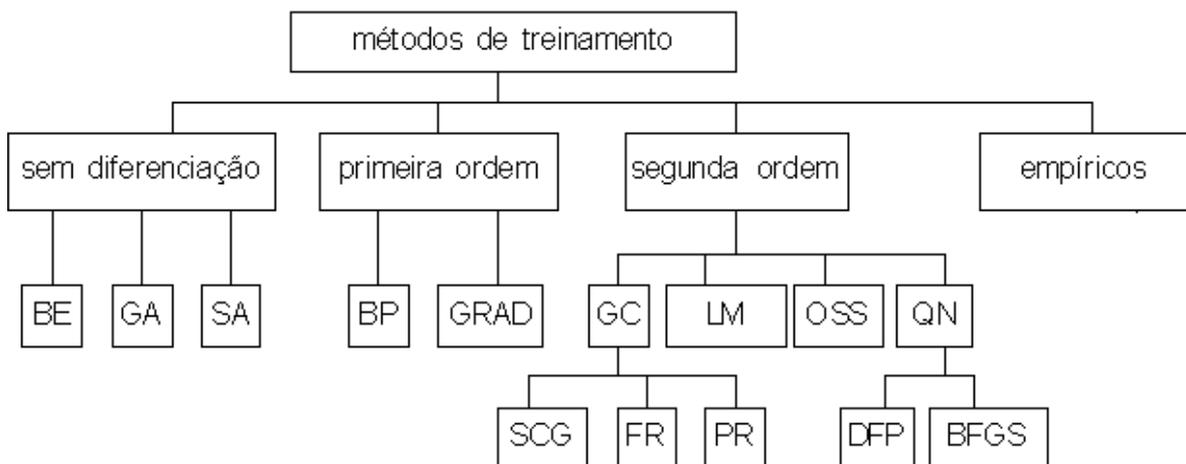
- Aprendizado supervisionado;
- Aprendizado não-supervisionado; e
- Aprendizado por reforço.

O método utilizado neste trabalho é o processo Supervisionado, onde os valores de saída da rede, resultado de cada apresentação das entradas, são comparados com a correspondente saída desejada, gerando um vetor de erro que é então supervisionado como orientador no processo de treinamento dos pesos. Assim, a rede deverá se ajustar internamente. Para o processo não-supervisionado, as entradas são agrupadas em classes, usualmente por similaridade ou proximidade geométrica dos sinais de entrada. Para o processo de aprendizado por reforço, um agente externo observa as ações da rede, se a ação tomada está de acordo com o comportamento desejado, o agente retorna um sinal de reforço para a rede, caso contrário um sinal de penalidade é retornado a esta rede.

O tempo necessário para o treinamento da rede depende fortemente dos pesos iniciais utilizados. Usualmente os pesos do vetor inicial W_0 são números aleatórios em torno de zero (-0,5 a +0,5). O processo de treinamento é interrompido quando todos os erros de apresentação dos subconjuntos de entrada e saída estiverem abaixo de uma referência estipulada (vide a equação. 4.3).

4.5 - CLASSES DOS ALGORITMOS DE TREINAMENTO

O *perceptron* multicamadas é uma extensão do *perceptron* de camada única. Esta arquitetura apresenta uma camada com unidades de entrada, conectada a uma ou mais unidades intermediárias, chamadas camadas ocultas, e uma camada de unidades de saída. Os algoritmos de treinamento de *perceptrons* multicamadas podem ser classificados nos seguintes itens apresentados na Figura 4.2 (Haykin, 1999):



BE-	Busca Exaustiva	FR-	Fletcher-Reeves
GA-	Genetic Algorithms	PR-	Polak-Ribière
SA-	Simulated Annealing	LM-	Levenberg-Marquardt
BP -	BackPropagation	OSS-	One Step Secant
GRAD-	Gradiente	QN-	Quase Newton
GC-	Gradiente Conjugado	DFP-	Davidon-Fletcher-Powell
SCG-	Gradiente Congudado Scaled	BFGS-	Broyden-Fletcher-Goldfarb-Shanno

Figura 4.2 - Classes dos algoritmos de treinamento.

Primeira classe - algoritmos que não requerem derivação, apenas avaliação da função em diferentes pontos do espaço, são chamados métodos sem diferenciação; Segunda classe -

faz uso da derivada primeira a ser minimizada, são chamados métodos de primeira ordem; Terceira classe de algoritmos são os métodos de segunda ordem, e utilizam informações sobre a derivada segunda; Outra classe consiste no ajuste de pesos usando método de tentativas e erros, e é denominado método empírico.

4.5.1 – MÍNIMOS LOCAIS E GLOBAIS

- O ponto x^* é um mínimo local de $F(x)$ se existe um escalar $\varepsilon > 0$, tal que

$$F(x^*) < F(x + \Delta x) \text{ para todo } \Delta x \text{ tal que } 0 < \|\Delta x\| < \varepsilon .$$

- O ponto x^* é um mínimo global único, Figura 4.3, de $F(x)$ se $F(x^*) < F(x + \Delta x)$ para todo $\Delta x \neq 0$.

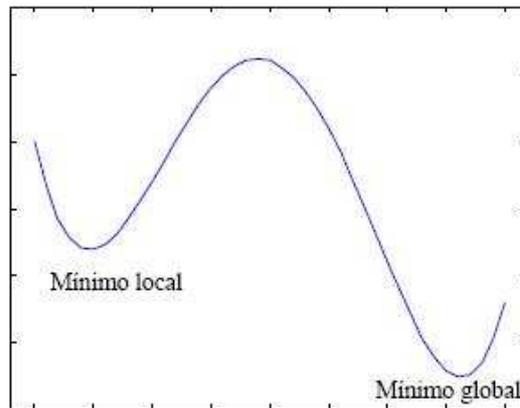


Figura 4.3 – Representação: mínimo local e global

Os mínimos locais são pontos na vizinhança dos quais o erro possui o menor valor. Os métodos de segunda ordem, teoricamente, não são mais capazes de encontrar um mínimo global do que os métodos de primeira ordem. A determinação de mínimos globais, é difícil devido a impossibilidade fundamental de se reconhecer um mínimo global utilizando-se apenas informações locais, mesmo dentro de um conjunto de mínimos locais bem definidos. Mínimos locais não são únicos pela razão da função que é multimodal e se a Hessiana é singular em um mínimo local, este mínimo constitui um conjunto compacto ao invés de um ponto isolado, o valor da função deve ser constante ao longo de uma direção, de um plano ou de um subespaço maior.

O método sem diferenciação podemos citar *simulated annealing* (SA), algoritmos genéticos (GA) e busca exaustiva (BE).

O método de primeira ordem, o erro quadrático médio a ser minimizado pode ser apresentado considerando-se seus termos até segunda ordem pela seguinte equação:

$$J_{quad}(\theta) = J(\theta_i) + \nabla J(\theta_i)^T (\theta - \theta_i) + (\theta - \theta_i)^T \nabla^2 J(\theta_i) (\theta - \theta_i) \quad (4.31)$$

onde $\nabla J(\theta_i)$ é o vetor gradiente e $\nabla^2 J(\theta_i)$ é a matriz hessiana de $J(\theta)$, ambos calculados no ponto $\theta = \theta_i$ e $J_{quad}(\theta)$ representa a aproximação até segunda ordem para $J(\theta)$. Para o método de primeira ordem apenas os termos constante e linear em θ da expansão em série de Taylor são considerados.

Os métodos, onde o gradiente local determina a direção de minimização d ; onde $\theta_{i+1} = \theta_i + \alpha d_i$ para $i \geq 0$, onde $\theta_i \in \mathfrak{R}^P$ é o vetor de parâmetros, $\alpha_i \in \mathfrak{R}^+$ é um escalar que define o passo de ajuste e $d_i \in \mathfrak{R}^P$ é a direção de ajuste, todos definidos na iteração i ; são conhecidos como métodos da direção de maior decrescimento (*steepest descent* ou *gradient descent*). O vetor de parâmetro θ deste item, corresponde ao w_{ij} descrito no item 4.3.

O algoritmo padrão de primeira ordem funciona quando a rede está em um estado θ_i , o gradiente $J(\theta_i)$ é calculado e um passo de minimização na direção oposta ao gradiente $d = -J(\theta_i)$ é efetuado. No algoritmo padrão, a minimização é geralmente feita com um passo α fixo. A determinação do passo α é fundamental, pois para valores muito baixos, o tempo de treinamento pode tornar-se exageradamente alto, e para valores muito altos, os parâmetros podem divergir. A velocidade de convergência é geralmente melhorada adicionando-se um termo de momento $\theta_{i+1} = \theta_i + \alpha_1 d_i + \beta_i \Delta \theta_{i-1}$, para $i \geq 0$. Este termo adicional evita oscilações no comportamento do erro, pois pode ser interpretado como a inclusão de uma aproximação da informação de segunda ordem.

Dentre os métodos que utilizam diferenciação e busca, o método do gradiente é o mais simples para obtenção da direção d_i , pois utiliza apenas informações de primeira ordem. Na i -ésima iteração, a direção d_i é definida como a direção de módulo unitário de maior decrescimento da função J :

$$d = -\frac{\nabla J(\theta)}{\|\nabla J(\theta)\|}. \quad (4.32)$$

A lei de ajuste do método do gradiente é, então, dada por:

$$\theta_{i+1} = \theta_i - \alpha_i \frac{\nabla J(\theta_i)}{\|\nabla J(\theta_i)\|}. \quad (4.33)$$

Nos métodos de segunda ordem é considerada a maneira mais eficiente de se fazer o treinamento de redes neurais do tipo multicamadas. Pois estes algoritmos recorrem a um rigor matemático baseado em modelos de otimização não-linear irrestrita bem definidos, não apresentando assim um vínculo natural com a inspiração biológica inicialmente proposta para as redes neurais artificiais.

Para todos os métodos a serem descritos a seguir, a exceção dos métodos de gradiente conjugado escalonado, utilizam um procedimento de busca unidimensional para encontrar um α_i , que seja solução ótima do problema:

$$\min J(\theta_i + \alpha_i d_i) \quad \alpha_i \in [0,1]. \quad (4.34)$$

O método de Newton pode ser considerado como o método local básico que utiliza informações de segunda ordem. A sua aplicação prática aos perceptrons multicamadas é pouco recomendada uma vez que o cálculo da matriz Hessiana representa um elevado custo computacional, pois exige a inversão, análise espectral e armazenagem de uma matriz quadrada que é da ordem do número de parâmetros P a serem ajustados.

O vetor θ_{i+1} , é a solução que minimiza exatamente $J(\theta)$ dado pela equação (4.31), atendendo portanto à condição de otimalidade:

$$\frac{\partial J_{quad}(\theta_{i+1})}{\partial(\theta_{i+1})} = 0. \quad (4.35)$$

Aplicando a equação (4.35) na equação (4.31) resulta:

$$\theta_{i+1} = \theta_i - [\nabla^2 J(\theta_i)]^{-1} \nabla J(\theta_i), \quad (4.36)$$

onde $\nabla^2 J(\theta_i)$ é a matriz hessiana e $\nabla J(\theta_i)$ o vetor gradiente.

Para o mesmo raciocínio empregado no caso do método do gradiente, como a função $J(\theta)$ não é necessariamente quadrática, a minimização de sua aproximação quadrática $J_{quad}(\theta)$ pode não fornecer uma solução θ_{i+1} tal que $J(\theta_{i+1}) < J(\theta_i)$. A lei de ajuste (4.36) torna-se então:

$$\theta_{i+1} = \theta_i - \alpha_i [\nabla^2 J(\theta_i)]^{-1} \nabla J(\theta_i) \quad (4.37)$$

Conforme descrito, o método de Newton ainda não apresenta garantia de convergência, pois nada pode ser afirmado sobre o sinal da matriz hessiana, que deve ser definida positiva para garantir que a aproximação quadrática tenha um mínimo; e para garantir que a inversa da matriz hessiana exista, que é a condição necessária, para resolver a equação (4.37) a cada iteração.

Portanto, é necessário testar a positividade de $\nabla^2 J(\theta)$ a cada iteração, e na eventualidade de se obter $\nabla^2 J(\theta) \leq 0$, deve-se aplicar um procedimento de positivação desta matriz. Dada uma matriz A simétrica, conhecendo-se o menor autovalor λ_{\min} desta matriz, é possível obter uma matriz M definida positiva a partir de A na forma: se $\lambda_{\min} > 0$, então $M = A$ e se $\lambda_{\min} \leq 0$, então $M = A + (\varepsilon - \lambda_{\min})I$, com $\varepsilon > \lambda_{\min}$.

A lei de ajuste do método de Newton torna-se:

$$\theta_{i+1} = \theta_i - \alpha_i M_i^{-1} [\nabla^2 J(\theta_i)]^{-1} \nabla J(\theta_i) \quad (4.38)$$

Onde,

$$M_i = \nabla^2 J(\theta_i) \text{ se } \lambda_{\min}^{[i]} > 0 \text{ e } M_i = \nabla^2 J(\theta_i) + (\varepsilon - \lambda_{\min}^{[i]})I \text{ se } \lambda_{\min}^{[i]} \leq 0$$

com $\lambda_{\min}^{[i]}$ o autovalor mínimo de M_i . Ainda não existem resultados que conduzam à determinação automática de um valor ótimo para ε . Em lugar de positivar M_i poder-se-ia, em princípio, utilizar qualquer outra matriz definida positiva de dimensões apropriadas. A razão para optar pelo processo de positivação da hessiana é a analogia com o tradicional método de Levenberg-Marquardt (que será visto a seguir) que pode ser interpretado como uma combinação entre a lei de ajuste do método do gradiente e

a lei de ajuste do método de Newton. A determinação exata da matriz hessiana não é uma tarefa fácil. C. Bishop, (1992) apresenta expressões que permitem o cálculo exato da matriz hessiana para arquiteturas multicamadas com uma ou mais camadas intermediárias.

O método de Levenberg-Marquardt, assim como o método de Newton, é bastante eficiente quando estamos tratando de redes que não possuem mais do que algumas centenas de conexões a serem ajustadas. Isto se deve ao fato de que estes algoritmos necessitam armazenar uma matriz quadrada cuja dimensão é da ordem do número de conexões da rede. O funcional de erro apresentado na equação (4.39) representa o erro quadrático médio (MSE), envolvendo a distância entre funções:

$$J(\theta) = \frac{1}{N} \sum_{l=1}^N (g(x) - \hat{g}(x, \theta))^2, \quad (4.39)$$

Onde $\hat{g}(\cdot, \theta)$ é uma função apenas do vetor de parâmetros $\theta \in \mathcal{R}^P$.

Considera-se como funcional a soma dos erros quadráticos (SSE), e leva-se em conta que o problema pode ter múltiplas saídas, obtém-se a seguinte expressão para o funcional de erro:

$$J(\theta) = \sum_{i=1}^N \sum_{j=1}^m (g_{ij}(x) - \hat{g}_{ij}(x, \theta))^2 = \sum_{k=1}^q r_k^2, \quad (4.40)$$

onde N é o número de amostras, l o número de unidades intermediárias, r o erro residual, m o número de saídas, e q o produto $N \times m$.

Seja J o Jacobiano (matriz das derivadas primeiras) do funcional J dado pela equação (4.17), pode ser escrita da seguinte forma:

$$J \equiv \begin{bmatrix} \nabla r_1^T \\ \vdots \\ \nabla r_q^T \end{bmatrix}, \text{ onde } r \text{ é denominado de erro residual, e diferenciando obtém:}$$

$$\nabla J = 2J^T r = 2 \sum_{k=1}^q r_k \nabla r_k, \quad (4.41)$$

$$\nabla^2 J = 2(J^T J + \sum_{k=1}^q r_k \nabla^2 r_k) \quad (4.42)$$

Quando os erros residuais são suficientemente pequenos, a matriz hessiana pode ser aproximada pelo primeiro termo da equação (4.42), ficando $\nabla^2 J \approx 2J^T J$. Esta aproximação geralmente é válida em um mínimo de J para a maioria dos propósitos, e é a base para o método de Gauss-Newton. A lei de atualização torna-se $\Delta\theta = [J^T J]^{-1} J^T r$. A modificação de *Levenberg-Marquardt* para o método de Gauss-Newton fica como $\Delta\theta = [J^T J + \mu I]^{-1} J^T r$.

O efeito da matriz adicional μI é adicionar μ a cada autovalor de $J^T J$. Uma vez que a matriz $J^T J$ é semi-definida positiva e portanto o autovalor mínimo possível é zero, qualquer valor positivo, pequeno, mas numericamente significativo, de μ será suficiente para restaurar a matriz aumentada e produzir uma direção descendente de busca.

O método de *Davidon-Fletcher-Powell* (DFP), como o método BFGS, é classificado como método quase-Newton, fazendo uma aproximação iterativa da inversa da matriz hessiana, de forma que $\lim_{i \rightarrow \infty} H_i = \nabla^2 J(\theta)^{-1}$.

Os métodos teoricamente são considerados mais sofisticados na solução de problemas de otimização não-linear irrestrita e representam o ápice do desenvolvimento de algoritmos através de uma análise detalhada de problemas quadráticos, gerando as direções do método do gradiente conjugado ao mesmo tempo em que constrói a inversa da Hessiana. A cada passo a inversa da Hessiana é aproximada pela soma de duas matrizes simétricas de posto 1, procedimento que é geralmente chamado de correção de posto 2 (*rank 2 correction procedure*).

Sendo a construção da inversa $H_{i+1} = H_i + \frac{p_i p_i^T}{p_i^T q_i} - \frac{H_i q_i q_i^T H_i}{q_i^T H_i q_i}$, $i = 0, 1, \dots, P - 1$.

Onde $p_i = \alpha_i d_i$ e $q_i \equiv g_{i+1} - g_i = \nabla^2 J(\theta) p_i$. A avaliação do gradiente em dois pontos fornece informações sobre a matriz hessiana $\nabla^2 J(\theta)$.

Para o método de Broyden-Fletcher-Goldfarb-Shanno (BFGS), onde p e q são obtidos como anteriormente, tem-se que $H_{i+1} = H_i + \frac{p_i p_i^T}{p_i^T q_i} [1 + \frac{q_i^T H_i q_i}{p_i^T q_i}] - \frac{H_i q_i p_i^T + p_i q_i^T H_i}{p_i^T q_i}$.

O método das Secantes de um Passo (OSS – *One Step Secant*), provém do fato de que as derivadas são aproximadas por secantes avaliadas em dois pontos da função gradiente, onde a sua complexidade é de ordem $O(P)$, ou seja, é linear em relação ao número P de parâmetros, enquanto a complexidade dos métodos DFP e BFGS é de ordem $O(P^2)$, (Battiti, 1992).

A direção de busca d_{i+1} é obtida da forma $d_{i+1} = -g_i + A_i s_i + B_i q_i$, onde $s_i = \theta_{i+1} - \theta_i = p_i$ e

$$A_i = - \left[1 + \frac{q_i^T q_i}{s_i^T q_i} \right] \frac{s_i^T g_i}{s_i^T q_i} + \frac{q_i^T g_i}{s_i^T q_i}, \quad B_i = \frac{s_i^T g_i}{s_i^T q_i}. \quad (4.43)$$

O gradiente conjugado (GC): *Polak-Ribière* (PR) e *Fletcher-Reeves* (FR), trata-se de problemas de grande escala de maneira efetiva. Os métodos do gradiente conjugado possuem sua estratégia baseada no modelo geral de otimização apresentado no algoritmo padrão e do gradiente, mas escolhem a direção de busca d_i , o passo α_i e o coeficiente de momento β_i mais eficientemente utilizando informações de segunda ordem. O gradiente conjugado é projetado para exigir menos cálculos que o método de Newton e apresentar taxas de convergência maiores que as do método do gradiente. Basea-se no método das direções conjugadas proposto para tratar problemas quadráticos como $J(\theta) = \frac{1}{2} \theta^T Q \theta - b^T \theta$.

A lei de ajuste do método das direções conjugadas é dada por $\theta_{i+1} = \theta_i - \frac{d_i^T \nabla J(\theta_i)}{d_i^T Q d_i} d_i$. As direções Q-conjugadas $d_i \in \mathbb{R}^P$, $i=0, \dots, P-1$, (Bazaraa et. al., 1993) para $d_0 = -\nabla J(\theta_0)$, $d_{i+1} = -\nabla J(\theta_{i+1}) \beta_i d_i$, $i \geq 0$, com $\beta_i = \frac{\nabla J(\theta_{i+1})^T Q d_i}{d_i^T Q d_i} d_i$, e para problemas não-quadráticos a matriz Q deve ser aproximada pela matriz hessiana calculada no ponto θ_i , envolvendo um procedimento de busca unidimensional do passo de ajuste e a aproximação do parâmetro β utilizando informações de primeira ordem (gradiente).

Uma destas aproximações é dada pelo método de Polak-Ribière (PR), se $(i \bmod P \neq 0)$, faça $d_{i+1} = g_i + \beta_i d_i$, onde $\beta_i = \frac{g_{i+1}^T (g_{i+1} - g_i)}{g_i^T g_i}$, senão faça $d_i = g_i$. Outra aproximação é dada pelo método de Fletcher-Reeves (FR), se $(i \bmod P \neq 0)$, faça $d_{i+1} = g_i + \beta_i d_i$, onde

$$\beta_i = \frac{\|g_{i+1}\|^2}{\|g_i\|^2}, \text{ senão faça } d_i = g_i.$$

Os métodos de segunda ordem utilizam um procedimento de busca unidimensional para a determinação da taxa de aprendizagem, envolvendo um grande número de avaliações da função ou de sua derivada, tornando o processo bastante custoso do ponto de vista computacional. Uma nova variação no algoritmo de gradiente conjugado foi introduzida (Moller, 1993), Gradiente Conjugado Escalonado – SCG, que evita a busca unidimensional a cada iteração utilizando uma abordagem de Levenberg-Marquardt cujo objetivo é fazer um escalonamento do passo de ajuste α .

O método do Gradiente Conjugado Escalonado Modificado (SCGM) resulta de uma modificação importante do algoritmo SCG, que permite o cálculo exato da informação de segunda ordem. A utilização de um operador diferencial capaz de calcular exatamente o produto entre a matriz hessiana e qualquer sinal desejado, sem a necessidade de se calcular e armazenar a matriz hessiana foi proposta (Pearlmutter, 1994). Para o caso dos perceptrons de múltiplas camadas, o vetor a ser multiplicado pela hessiana corresponde a direção de busca na iteração anterior.

4.6 - CONTROLE NEURAL

O controle neural para o controle dos níveis dos tanques do LAVSI foi inicialmente simulado no MATLAB[™] e posteriormente implementado no controlador programável. Devido à grande utilidade do controlador programável em chão de fábrica, a implementação de técnicas de controle, proposta dessa tese, por meio do algoritmo de controle neural em um equipamento voltado a indústria, totalmente independente do computador comercial e análise dessas respostas seriam de grande valia para estudos atuais e futuros. Para a aplicação da técnica de controle neural, foi utilizado o modelo da planta. Inicialmente foram sintonizados controladores PID para testes em degrau em diversos pontos de operação da planta. Através de simulação foram gerados dados de entrada / saída dos diversos controladores PID relacionados de acordo com o ponto de operação da planta. Estes dados foram então utilizados para o treinamento do Controlador Neural.

4.7 - IMPLEMENTAÇÃO DE CONTROLE NEURAL PARA O PROCESSO

A seguir, apresenta-se as técnicas de projeto de controlador neural que foram aplicadas no modelo do processo de tanques acoplados. O problema de controle deste trabalho é regular a variável medida $y(t)$ no valor de referência r através do sinal de controle $u(t)$.

O sinal de controle para a moto-bomba é $u(t)$, e o sinal de saída dos níveis é $y(t)$. O sinal de controle para a moto-bomba é constante para um consumo ou saída em queda livre constante de $x\%$ do valor máximo de vazão de saída.

Para a obtenção dos sinais, para treinamento, em torno de cada ponto de operação serão implementados vários controladores PI's ao longo da faixa de operação da planta (nível dos tanques acoplados). A técnica de controle PI foi utilizada, pois possui características bem conhecidas e de baixa complexidade, possibilitando que o esforço principal fosse centrado em relação à implementação do controlador neural. Assim, foi possível obter um controlador não-linear global por este *gain scheduling* obtidos para cada ponto de operação. O controle PI de malha fechada mantém a variável de processo em valor de referência desejado.

A equação de um controlador PI discreto é dado por:

$$u(k) = K_p e(t) + K_i \sum_{i=0}^t e(t) \quad (4.44)$$

$$e(t) = h_r(t) - h(t) \quad (4.45)$$

onde K_p e K_i são respectivamente o ganho proporcional e o ganho integral.

A implementação de um controlador neural *gain scheduling* pode ser implementada nesta planta. Segundo a abordagem do controlador neural *gain scheduling*, é requerida a introdução de um integrador na entrada da planta, definido como $v(t) = u(t) - u(t-1)$, no estudo do controlador *gain scheduling*. Assim, em dois instantes de amostragem consecutivos obtém-se:

$$v(t) = K_p [e(t) - e(t-1)] + K_i e(t), \quad (4.46)$$

que é uma forma do controlador discreto PI incremental. Esta forma de controlador poderá ser usada na etapa de projeto de controladores para diversos pontos de operação da planta de Tanques Acoplados.

O modelo incremental PI foi implementado na planta dos tanques acoplados conforme Figura 4.4. Em regime estacionário para cada ponto de operação (r, u^* e q_0) obtém-se:

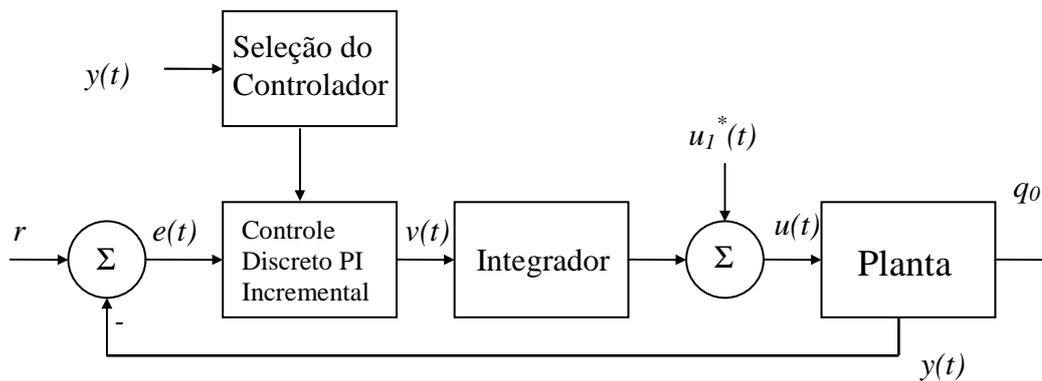


Figura 4.4 - Modelo incremental PI no sistema de tanques acoplados.

Para a sintonia dos vários controladores PIs em cada ponto de operação, serão empregados os ajustes recomendados por Ziegler-Nichols, 1942 (Ogata, 2003). Em essência para sistemas estáveis em malha aberta, esse método consiste em se aumentar gradativamente o ganho proporcional do controlador até que o sistema comece a oscilar. O período dessa oscilação T_u , e o ganho do controlador K_u correspondentes são utilizados na determinação dos parâmetros K_p , T_i e T_d .

Para a sintonia dos controladores PI, discretiza-se o modelo da planta com período de amostragem de 1 segundo e simula-se no MATLAB. Estes controladores PIs serão sintonizados para testes de degrau entre 15% e 25% da faixa de controle.

De posse dos vetores de entradas e saídas dos diversos controladores PI em malha fechada gerados pela Figura 4.4, partiremos para o treinamento do controlador neural *gain scheduling*. Foi escolhido para o controle neural *gain scheduling* uma rede neural com duas camadas e com função de ativação tangente hiperbólica.

A estrutura da rede neural de melhor desempenho nos tanques acoplados é mostrada na Figura 4.5, com 2 sinais na entrada e um sinal na saída.

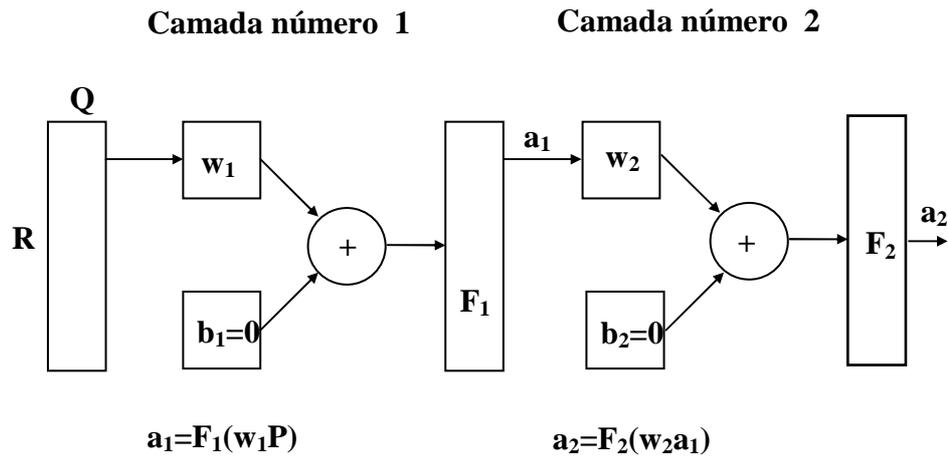


Figura 4.5 - Estrutura da rede neural artificial usada para o treinamento.

Onde:

$\mathbf{P} = \mathbf{R} \times \mathbf{Q}$	vetor de entrada,	
$\mathbf{T} = \mathbf{S} \times \mathbf{Q}$	vetor de saída	\mathbf{a}_2 ,
$\mathbf{w}_1 = \mathbf{S} \times \mathbf{Q}$	pesos ajustáveis,	
$\mathbf{b}_1 = \mathbf{0}$	limiar de excitação	
\mathbf{F}_1	função tangente hiperbólica	\mathbf{a}_1 ,
$\mathbf{w}_2 = \mathbf{S} \times \mathbf{R}$	pesos ajustáveis,	
$\mathbf{b}_2 = \mathbf{0}$	limiar de excitação	
\mathbf{F}_2	função tangente hiperbólica.	

A simulação foi realizada no MATLAB, onde a estrutura mostrada acima pode ser inicializada através do comando *initff*, após inicializada a rede neural foi treinada com o comando *trainbpx*. A simulação foi realizada diversas vezes até ser encontrado o número de neurônios da camada escondida que melhor satisfaz o controle proposto. É interessante que seja um número bem reduzido de neurônios na camada escondida para diminuir os esforços da implantação da rede no controlador lógico programável.

Para o ajuste dos pesos durante o treinamento, foi feito com variantes do algoritmo *backpropagation* de forma a minimizar a soma dos erros quadráticos obtidos entre os valores desejados e atuais de saída para valores dos dados de entrada. Esta variação foi feita continuamente no sentido da descida de gradiente em relação ao erro.

O diagrama da Figura 4.6 mostra como foi projetado o controlador neural *gain scheduling*, onde y^* é o valor da variável em um determinado ponto de operação do sistema.

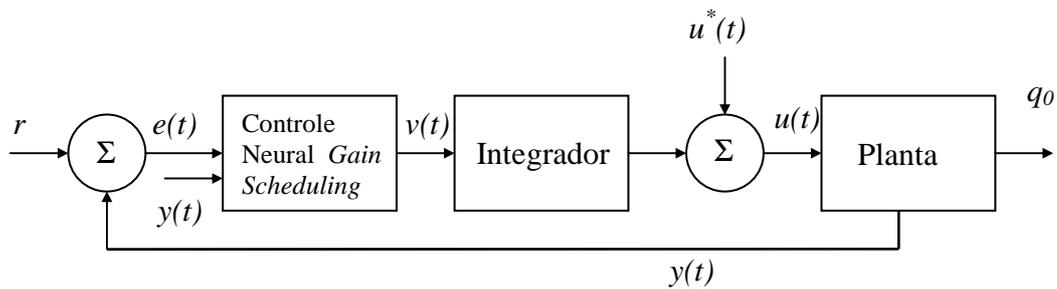


Figura 4.6 - Modelo de simulação do controle *gain scheduling*.

4.8 - IDENTIFICAÇÃO E CARACTERIZAÇÃO DA REDE NEURAL

A habilidade da rede neural de aproximar grandes classes de funções não-lineares permite o uso em modelo dinâmico para a representação de plantas não-lineares. Os métodos da rede neural podem ser usados para o ajuste de seus parâmetros que foram empregados nos controladores. Os modelos para a representação do processo de sistemas SISO podem igualmente ser generalizados ao caso multivariável. Os modelos de identificação possuem redes neurais multicamadas como subsistemas. Estes modelos foram usados na literatura dos sistemas adaptáveis para a identificação e controle de sistemas lineares e podem ser considerados como generalização aos sistemas não-lineares.

4.8.1 - CARACTERIZAÇÃO

Os quatro modelos das plantas em tempo discreto podem ser descritos pelas seguintes equações de diferença não-lineares (Narendra, 1990):

1º Modelo:

$$y_p(k+1) = \sum_{i=0}^{n-1} \alpha_i y_p(k-i) + g[u(k), u(k-1), \dots, u(k-m+1)] \quad (4.47)$$

2º Modelo:

$$y_p(k+1) = f[y_p(k), y_p(k-1), \dots, y_p(k-n+1)] + \sum_{i=0}^{m-1} \beta_i u(k-i) \quad (4.48)$$

3º Modelo:

$$y_p(k+1) = f[y_p(k), y_p(k-1), \dots, y_p(k-n+1)] + g[u(k), u(k-1), \dots, u(k-m+1)] \quad (4.49)$$

4º Modelo:

$$y_p(k+1) = f[y_p(k), y_p(k-1), \dots, y_p(k-n+1), u(k), u(k-1), \dots, u(k-m+1)] \quad (4.50)$$

Onde $[u(k), y_p(k)]$ representam os pares de entrada e saída da planta SISO no tempo k e $m \leq n$. A representação dos modelos está mostrada nas Figuras 4.7 a 4.10. As funções $f: R^n \longrightarrow R$ no modelo 2 e 3 e $f: R^{n+m} \longrightarrow R$ no modelo 4, e $g: R^m \longrightarrow R$ são assumidos para as funções diferenciais e seus argumentos. Nos modelos, a saída da planta $k+1$ depende do valor de n passado, $y_p(k-i)$ ($i=0,1,\dots,n-1$), assim como os valores passados de m na entrada de $u(k-j)$ ($j=0,1,\dots,m-1$). A dependência dos valores passados $y_p(k-i)$ é linear no primeiro modelo, quanto no segundo modelo a dependência nos valores passados da entrada $u(k-j)$ é assumido linear. No terceiro modelo, a dependência não linear de $y_p(k+1)$ em $y_p(k-i)$ e $u(k-j)$ é separado. No quarto modelo em que $y_p(k+1)$ representa uma função não linear de $y_p(k-i)$ e $u(k-j)$ no primeiro e terceiro modelos.

Uma planta dinâmica não-linear pode ser representada por um dos quatro modelos apresentados através das equações (4.47) a (4.50). Cada planta pode ser identificada usando os dados de entrada/saída, implicando que o modelo escolhido represente a planta que foi controlada. A pesquisa das propriedades da estabilidade utilizando multicamadas representa uma área importante que se deve ser explorada.

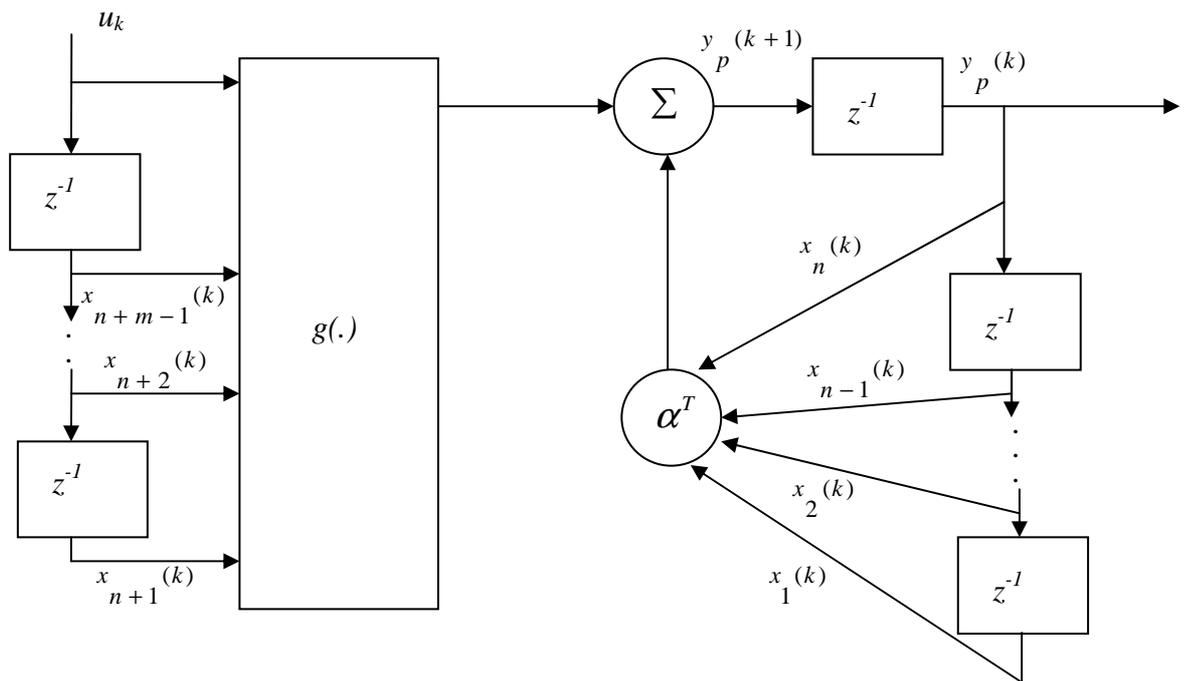


Figura 4.7. Representação esquemática da planta SISO - (a) Model I.

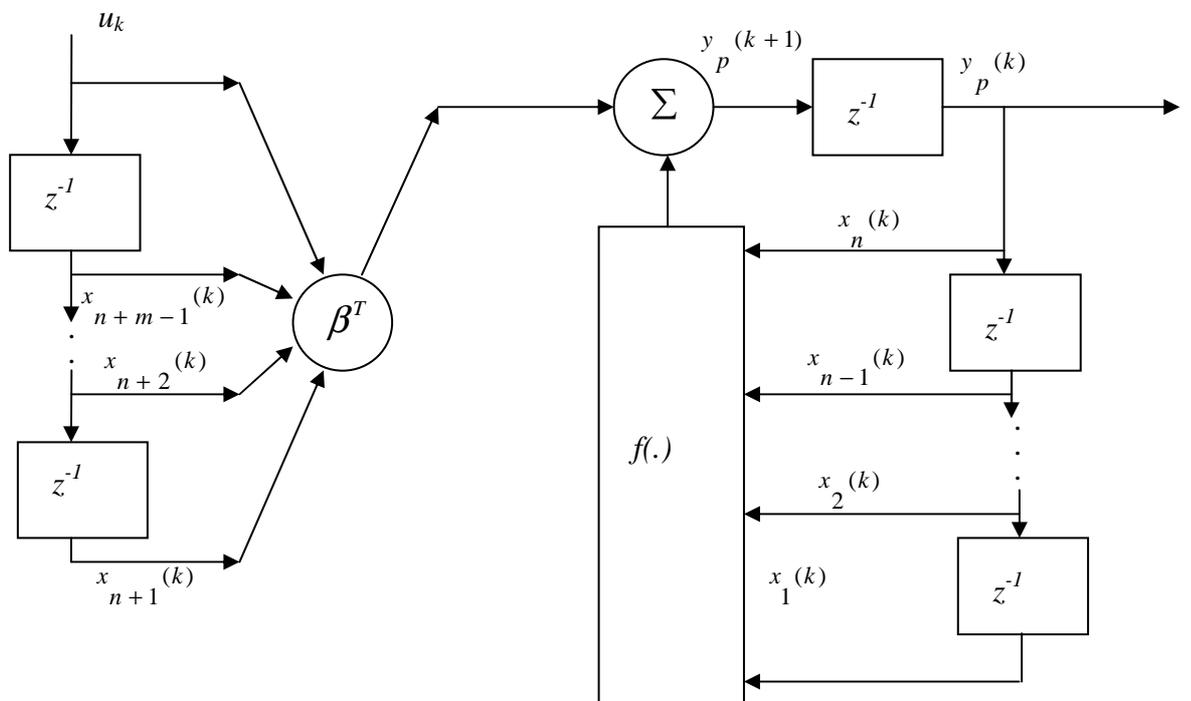


Figura 4.8. Representação esquemática da planta SISO. (b) Model II

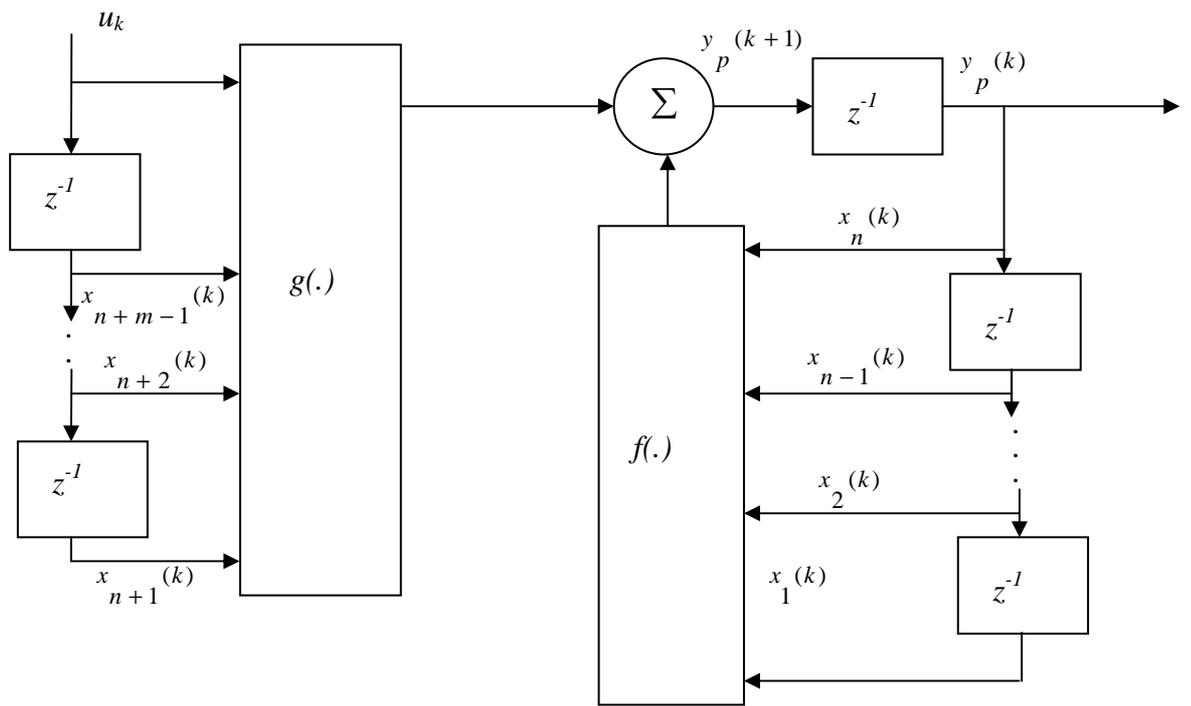


Figura 4.9. Representação esquemática da planta SISO. (c) Model III

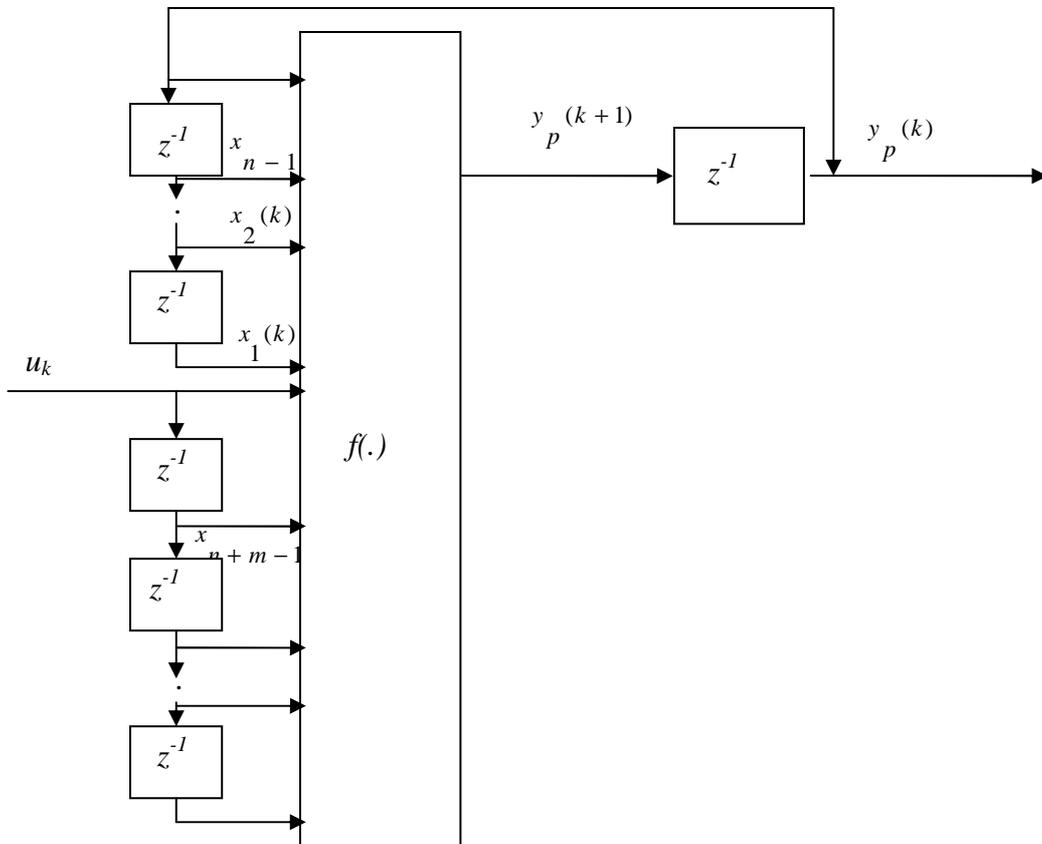


Figura 4.10. Representação esquemática da planta SISO. (d) Model IV

4.8.2 - IDENTIFICAÇÃO

O problema de identificação consiste basicamente na seleção de uma estrutura de modelo e no ajuste adequado de seus parâmetros de forma a aperfeiçoar uma função de desempenho baseada no erro entre a planta e o da saída do modelo identificado. Desde que as funções não lineares, que representam a planta são assumidas e conhecidas. A estrutura do modelo deve ser escolhida idêntica ao da planta.

A matriz de pesos da rede neural no modelo de identificação, devem ter as mesmas condições iniciais. A planta e o modelo devem ter a mesma saída e a mesma entrada. Logo o procedimento de identificação consiste em ajustar os parâmetros da rede neural, baseado entre o erro da planta e do modelo. Entretanto, precauções devem ser tomadas para se assegurar de que os resultados do procedimento na convergência da identificação modelem parâmetros a seus valores desejados.

A seguir são apresentados dois exemplos de identificação de modelo em paralelo e em serie-paralelo com $n = 2$ e $m = 1$.

1 – Identificação do modelo em paralelo

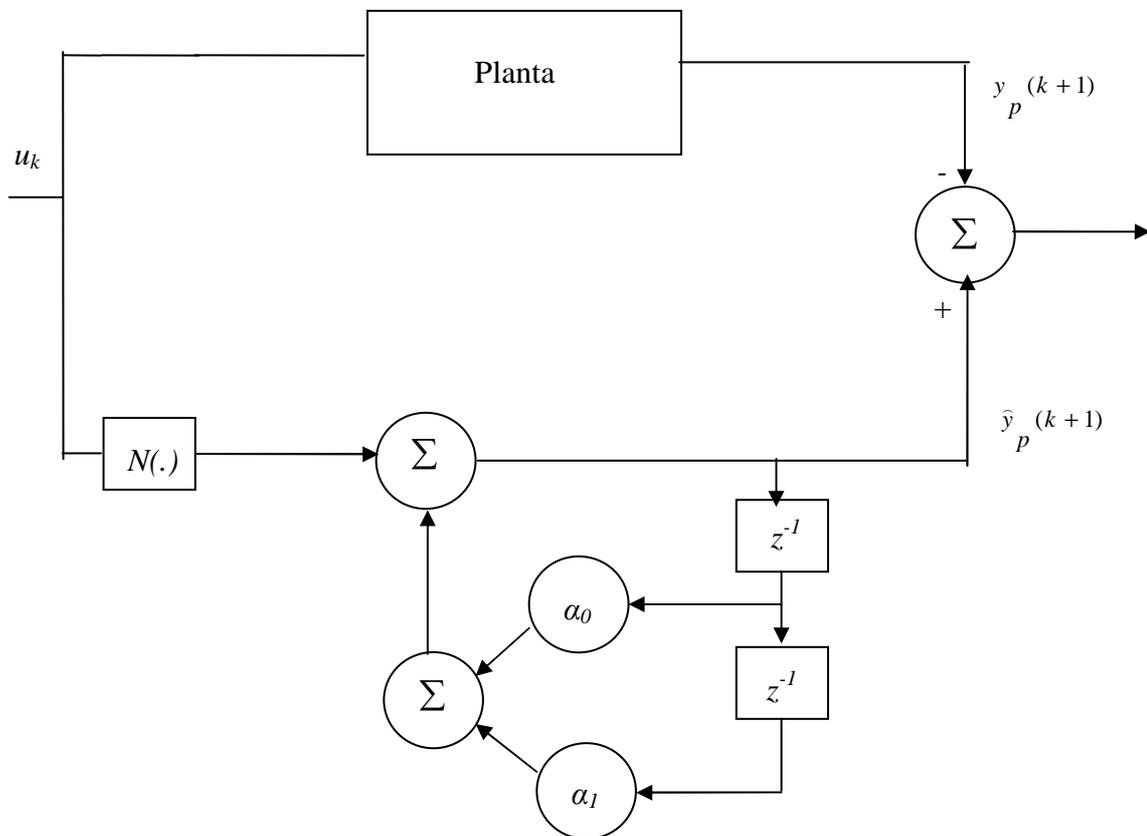


Figura 4.11. Representação esquemática para a identificação paralela

A Figura 4.11 mostra uma planta que pode ser representado pelo modelo 1. Para identificar essa planta pode-se assumir uma estrutura de identificação igual ao da Figura 4.11 e descrita pela seguinte equação:

$$y_p(k+1) = \alpha_0 y_p(k) + \alpha_1 y_p(k-1) + N[u(k)] \quad (4.51)$$

A identificação envolve avaliação dos parâmetros α_i , assim como os pesos da rede neural baseada em erro $e(k)$ entre a saída modelo $\hat{y}_p(k)$ e o valor atual da saída da planta real $y(k)$. A planta, nesse caso, tem a estabilidade BIBO (*Bounded Input-Bounded Output*), pois para qualquer sinal limitado aplicado entrada do sistema implica que o sinal de saída também fica limitado (independente do que ocorre com as variáveis internas do sistema). Em contraste, a estabilidade do modelo da identificação com uma rede neural não pode ser assegurada e tem que ser provada. Não há nenhuma garantia que os parâmetros convergirão ou que o erro de saída tenderá a zero, se o modelo paralelo for aplicado no modelo da planta.

2 – Identificação do modelo serie-paralelo

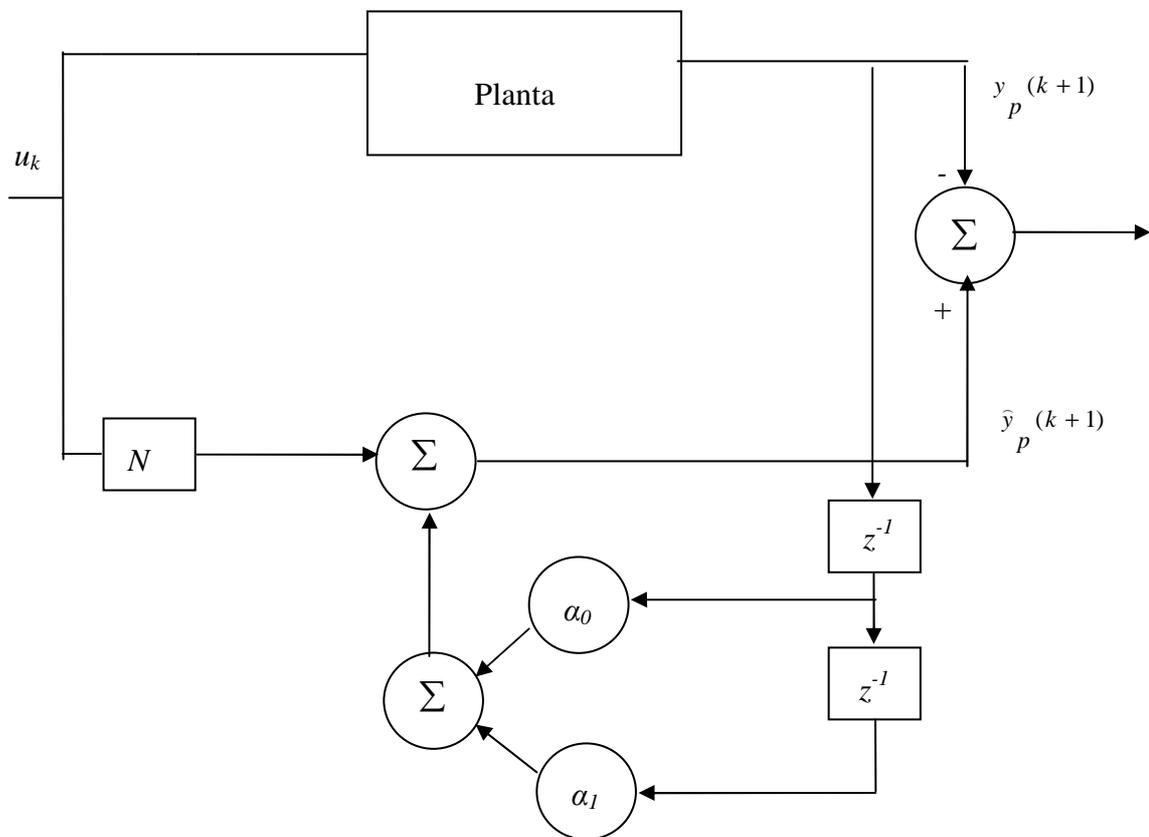


Figura 4.12. Representação esquemática para a identificação série

No modelo série-paralelo, a saída da planta é alimentada novamente no modelo da identificação, Figura 4.12. conforme descrito na equação:

$$\hat{y}_p(k+1) = \alpha_0 y_p(k) + \alpha_1 y_p(k-1) + N[u(k)] \quad (4.52)$$

Na Figura 4.13, mostra o bloco “*delay time*”, onde o sinal de saída tem os seus elementos em atraso em relação ao sinal de entrada. Os valores passados da entrada e da saída da planta formam o sinal de entrada da rede neural cuja a saída $y(k)$ corresponde à estimativa da saída de planta em um instante de tempo k . O modelo séries-paralelo possui vantagens em relação ao modelo paralelo. Possuindo a planta uma estabilidade BIBO, todos os sinais usados no procedimento da identificação (isto é, entrada na redes neural) são limitados. A retroalimentação é usada para ajustar os parâmetros e reduzir o esforço computacional. O erro tende a ser pequeno diminuindo de forma $\hat{y}_p(k)$ aproxima a $y(k)$.

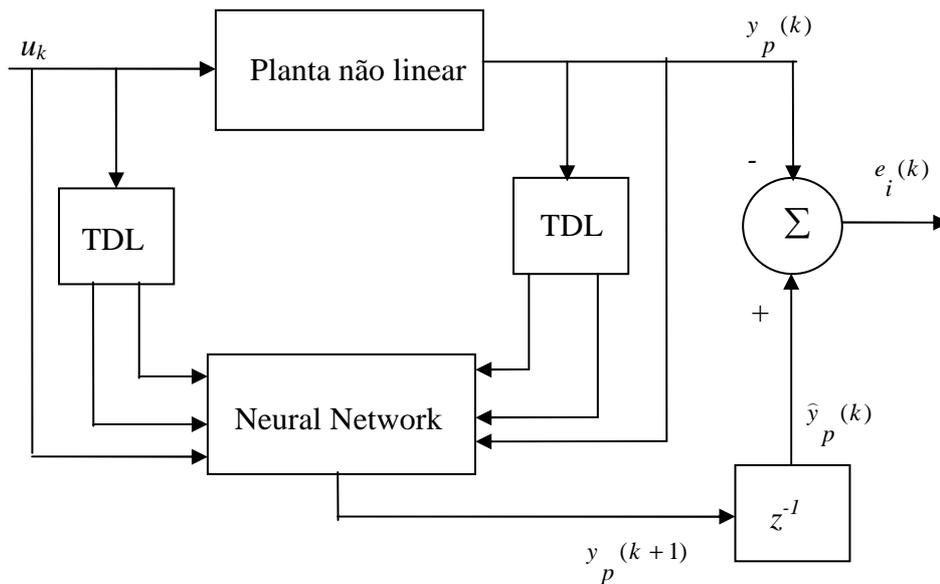


Figura 4.13. Representação esquemática para a identificação não linear

4.9 - REDES NEURAS COMO UMA OPÇÃO ALTERNATIVA

As redes neurais são consideradas como uma opção alternativa para resolução de problemas. Existem críticas em relação à sua metodologia e os resultados generalizados só podem ser comprovados através de métodos empíricos. Mesmo assim, existem vários

fatores que tornam as redes neurais uma escolha ideal em determinadas situações (Haykin, 1999), como as seguintes características:

- Não-linearidade: Uma rede neural formada por neurônios com propriedades não-lineares, acaba se tornando também não-linear;
- Mapeamento de entrada-saída: A rede neural é treinada para apresentar uma saída através do processamento sobre uma entrada fornecida;
- Adaptabilidade: A rede neural não possui conhecimento estático. É possível treiná-la para que se adapte a alterações em seu meio;
- Resposta a evidências: A rede é capaz de diferenciar fatos nos quais possui maior crença e outros em que não se tem tanto conhecimento;
- Informação contextual: Cada neurônio possui seu estado afetado pelas entradas. Como a saída final é encontrada com base neste conjunto de estados, a informação contextual é tratada naturalmente pela rede neural;
- Tolerância a falhas: Devido ao fato de possuir uma arquitetura distribuída, a rede neural pode ter alguns de seus neurônios destruídos e mesmo assim continuar funcionando com uma leve perda de qualidade;
- Implementação em VLSI: A rede neural tem uma paridade com as implementações em hardware VLSI (*very-large-scale-integration*) devido ao grande número de pequenos componentes que a constitui;
- Uniformidade de análise de projeto: As redes neurais são formadas por diversos neurônios, mas estes neurônios são todos idênticos, ou pelo menos semelhantes, entre si. Este fato favorece a manutenção e organização do sistema;

4.10 - COMENTÁRIOS

A investigação das técnicas neurais abordadas neste capítulo torna se uma base bem estruturada para a implementação destas técnicas neurais nos quatro tanques de água acoplados. A literatura sobre identificação de dinâmicas não lineares com redes neurais mostra várias técnicas de controle neural, como o artigo de Narendra e Parthasarathy (Narendra, 1990), que colocou de forma bastante clara o problema da aplicabilidade de redes multicamadas à identificação e controle das dinâmicas não lineares.

5 - CONTROLE NEURAL DO PROCESSO DE NÍVEL DE LÍQUIDOS

5.1 - INTRODUÇÃO

Para o projeto, treinamento, implementação, simulação e controle neural no processo, foram implementados em um computador comercial de uso geral. Para esse projeto de controle, utilizou-se o software Matlab/Simulink. A compatibilidade com os padrões elétricos devem ser atendidas.

Este trabalho investiga a aplicação do controlador neural em um computador comercial associado a uma planta com interface por meio de módulos analógicos instalados em controlador programável, concluindo que esta abordagem é factível para a utilização na indústria. Os experimentos apresentados neste capítulo possuem as seguintes aplicações para um processo de nível de líquidos:

- a) Simulação utilizando o software Matlab/Simulink, e
- b) Controle Neural do processo.

5.2 - IMPLEMENTAÇÃO DO CONTROLADOR NEURAL NO MATLAB

Depois de treinada, a rede é então utilizada para controlar a planta. Em alguns pontos de operação ao longo da altura de nível do quarto tanque, foram sintonizados controladores PID por meio de regras clássicas de Ziegler-Nichols, conforme item 2.3 e ($h_4 \leq 14,7\text{cm}$).

Para os testes realizados neste trabalho, utilizamos algumas variáveis desejadas na faixa de operação do tanque 4, com o objetivo de verificar o desempenho do Controlador Neural Genérico (um único conjunto de pesos para toda a faixa de operação). A taxa de amostragem usada para o controle foi de um segundo.

A Rede Neural foi treinada para aprender o mapeamento não linear (equações 2.3 a 2.6), usando o sinal de entrada para o processamento $e(t)$, $[e(t)-e(t-1)]$ e $h_4(t)$ garantindo o erro nulo entre a saída da planta e o valor de referência.

A arquitetura da Rede Neural Artificial que apresentou melhor desempenho para o controle foi à arquitetura composta dos seguintes parâmetros (Tabela 5.1): duas entradas, cinco neurônios na camada escondida (função de ativação tangente hiperbólica), um neurônio na camada de saída (função de ativação logística), com termos de polarização e utiliza-se o algoritmo de aprendizado retropropagação, conforme apresentado na Figura 5.1.

Para todos os experimentos da rede neural, utilizam-se os seguintes pesos e polarizações:

Tabela 5.1 – Parâmetros utilizados na RNA

w1 =[-11.9709 -19.7538	b1 =[1.2393	w2 = [-1.3384
20.8541 9.9310	-0.5798	1.2631 -0.8778
18.1510 14.2849	-1.8770	1.0339 1.6167]
18.6657 -13.6054	-0.0104	
14.9753 17.5857]	0.3015]	b2 =[-0.5758]

5.3 - RNA CONTROLANDO O MODELO DO PROCESSO

Apresenta-se neste item um controlador neural, para o controle de um processo de nível de líquidos de quatro tanques, realizado no SIMULINK[®] do software MATLAB[®].

Nessa aplicação, o controlador neural controla o modelo do processo tendo sido treinado com diversos algoritmos. Utiliza-se o modelo do sistema de controle conforme mostrado na Figura 5.1.

Foram realizados testes experimentais com a variável de referência no tanque 4 igual a 4,6 cm, para os parâmetros das válvulas entre os tanques k_{12} , k_{23} , e k_{34} , com valores iguais a 23,41 [cm^{2,5}/s] e os parâmetros dos furos de 2mm, k_2 e k_4 , localizados nos tanques 2 e 3 respectivamente com os valores iguais a 2,98 [cm^{2,5}/s] e 2,50 [cm^{2,5}/s]. A variável de processo é (h_4) e a variável de controle é a vazão de entrada no tanque 1.

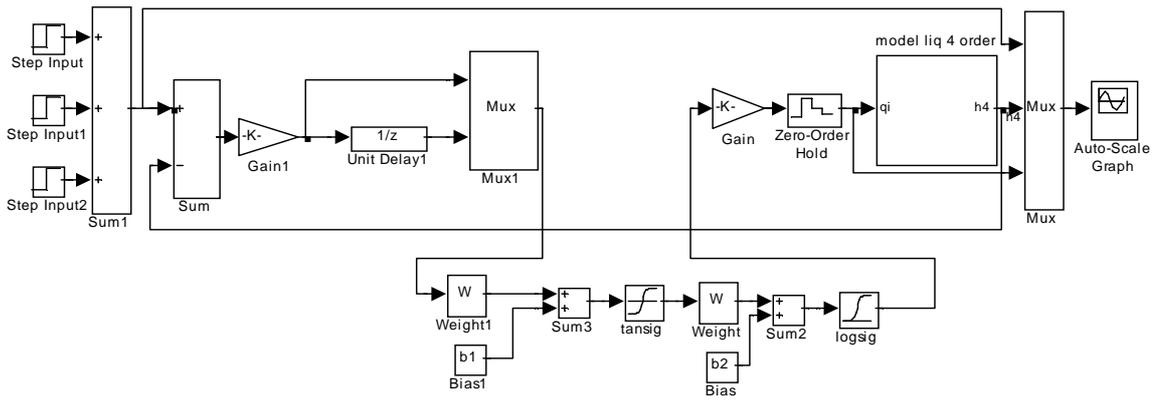


Figura 5.1 - Representação do Controlador Neural

A Figura 5.2 foi obtida através da implementação do controlador PID na planta, a Figura 5.3 foi obtida através de simulação do controlador neural no modelo da planta e a Figura 5.4 foi obtida através do controlador neural na planta, utilizando o aplicativo Matlab/Simulink e interface OPC.

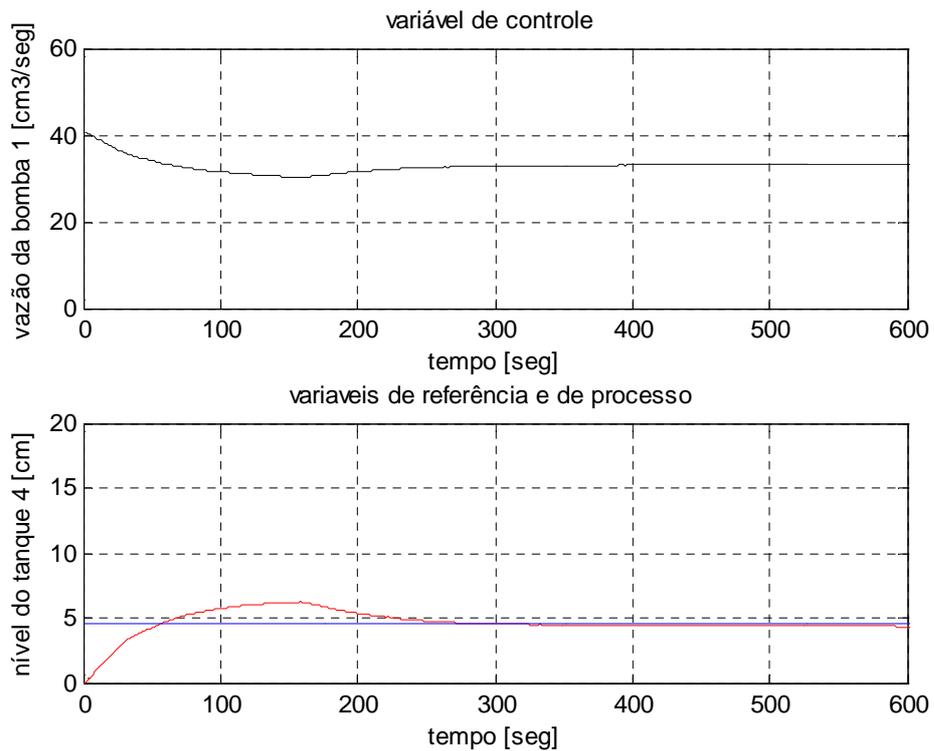


Figura 5.2 – Controle PID (referência: 4,6 cm).

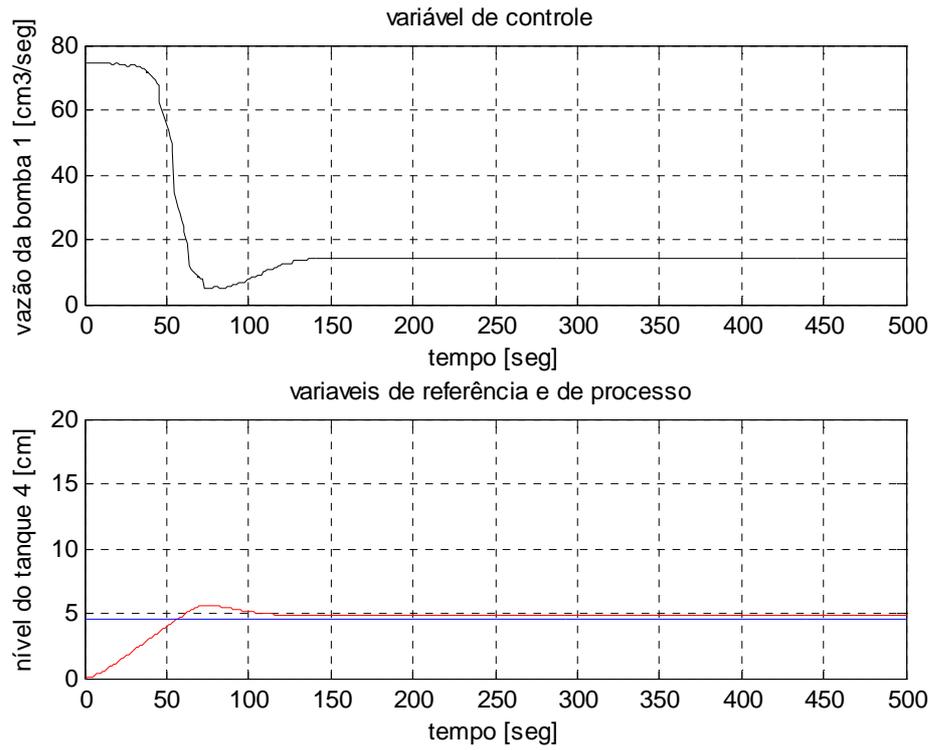


Figura 5.3 – Simulação do controlador neural (referência: 4,6 cm)

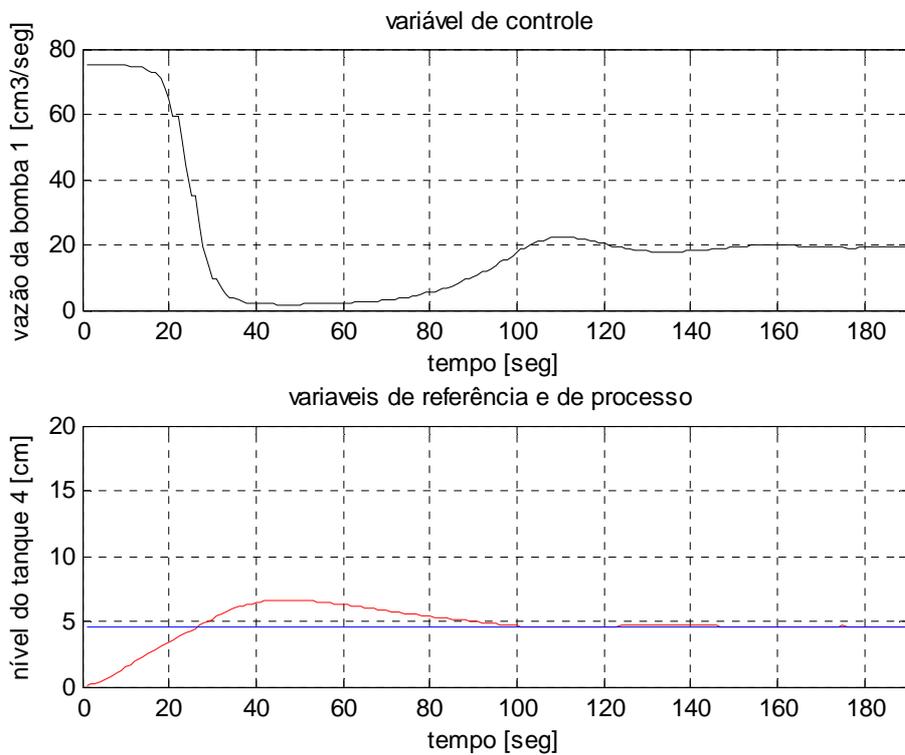


Figura 5.4 – Controlador Neural (referência: 4,6 cm)

Outros testes experimentais foram realizados para os parâmetros das válvulas entre os tanques k_{12} , k_{23} , e k_{34} , com valores iguais a $12 \text{ [cm}^{2.5}/\text{s]}$, $2,5 \text{ [cm}^{2.5}/\text{s]}$ e $6 \text{ [cm}^{2.5}/\text{s]}$ e os parâmetros dos furos de 2mm, k_2 e k_4 , localizados nos tanques 2 e 3 respectivamente com os valores iguais a $5 \text{ [cm}^{2.5}/\text{s]}$.

O gráfico da figura 5.5 foi obtido no ambiente Matlab, através do modelo da planta, a variável de referência foi de 4,6 cm. Já o gráfico da Figura 5.6 foi obtido, também em ambiente Matlab, com uma variável de referência igual a 7,4 cm. Nessas Figuras (5.5 e 5.6), a variável de processo é (h_4) e a variável de controle é a vazão de entrada no tanque 1. O gráfico da Figura 5.7 foi obtido com a variável de referência, também em 7,4 cm, porém simulando ruídos na medição dos 4 tanques.

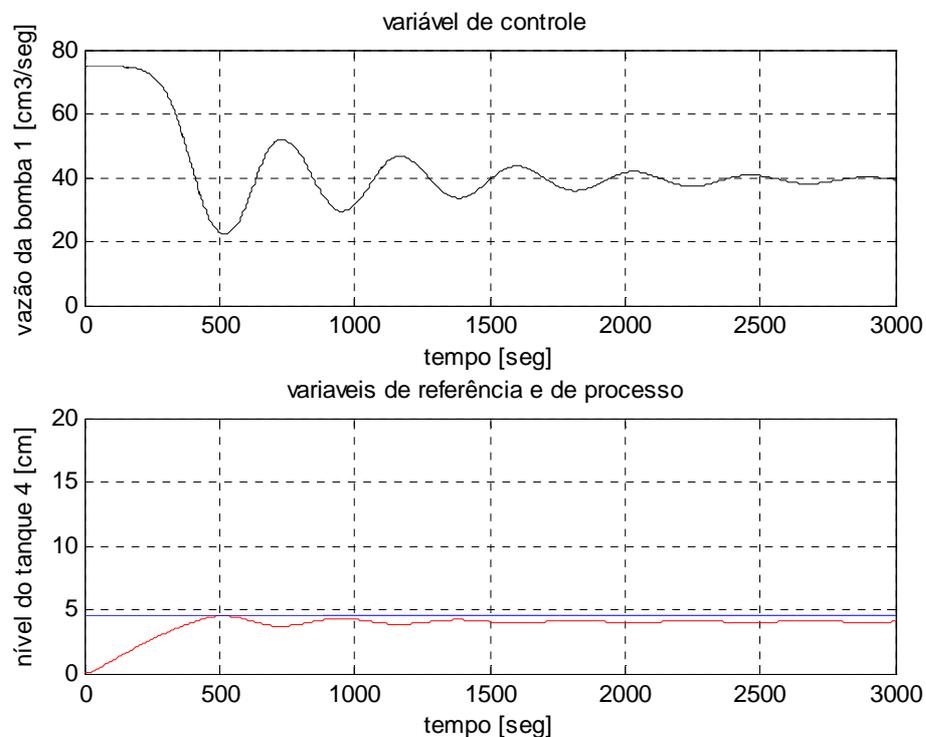


Figura 5.5 – Simulação do controlador neural, novos parâmetros da planta (referência: 4,6 cm)

No gráfico da Figura 5.7, mostra-se o comportamento das variáveis de níveis em todos os tanques desse processo, quando a variável de referência é igual a 7,4 cm. Notamos que existe uma saturação dos tanques 1 e 2. Como nestes tanques o valor da altura máxima é de 50 cm, então ocorre transbordamento do tanque 1, limitando-se o controle do nível do tanque 4.

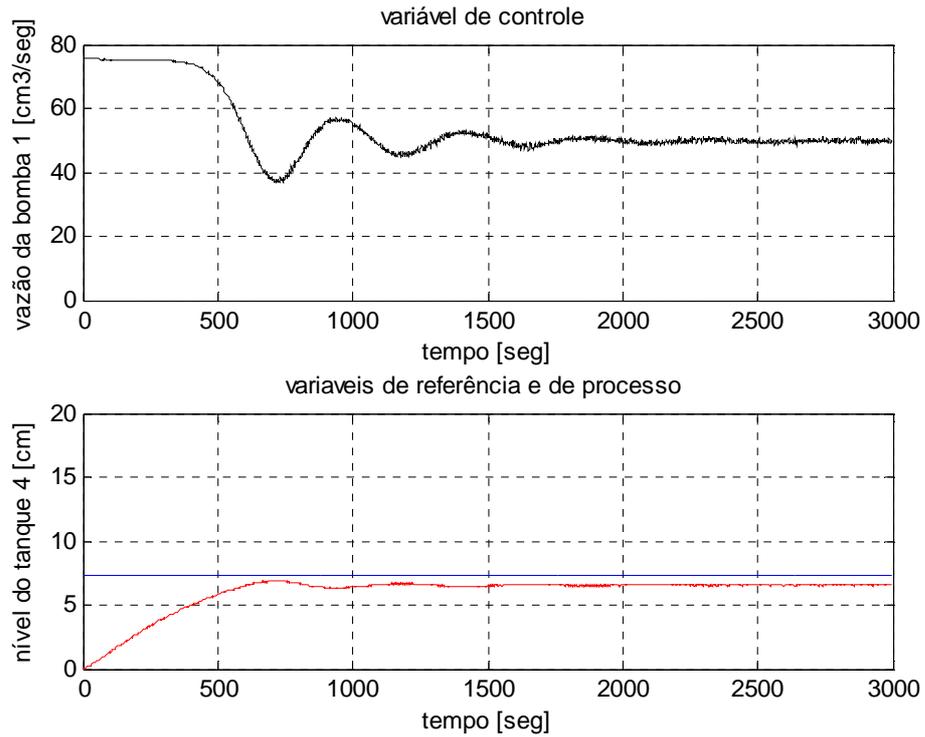


Figura 5.6 – Simulação do controlador neural, novos parâmetros da planta (referência: 7,4 cm)

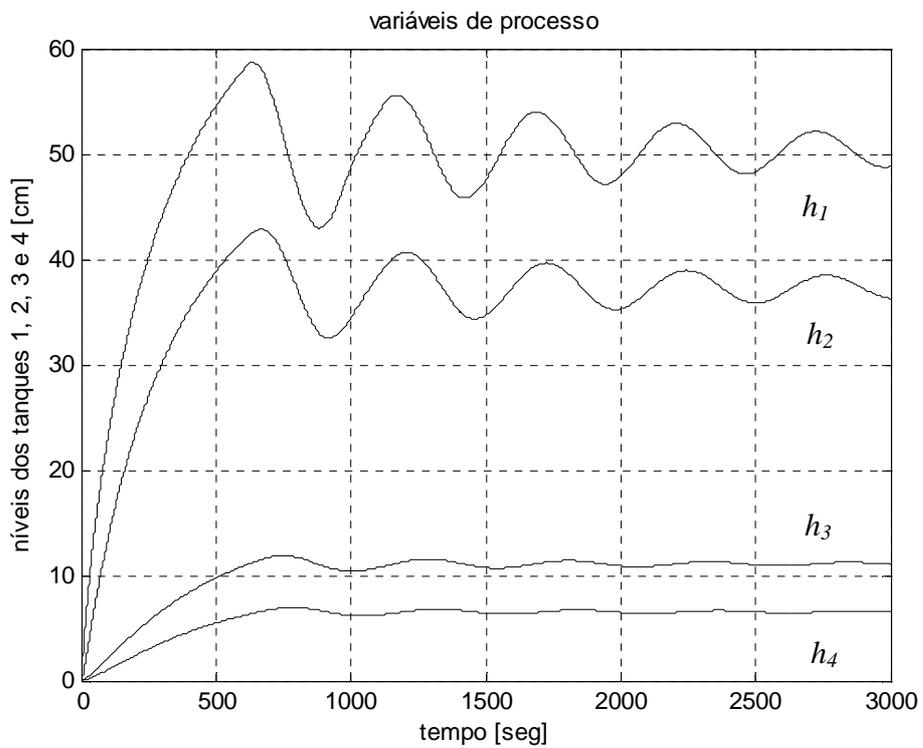


Figura 5.7 – Simulação do controlador neural (referência: 7,4 cm) e os respectivos níveis dos tanques (h_1 , h_2 , h_3 e h_4)

O período de amostragem usado nos experimentos foi de um segundo. Nos gráficos temos demonstrado a variável de processo (nível do tanque 4 - $h_4(t)$), a variável de referência e a variável de controle $u_1(t)$, para a moto-bomba que controla o fluxo da vazão de entrada no tanque 1.

5.4 - IMPLEMENTANDO UMA RNA COMPOSTA DE TRÊS CAMADAS

Apresentamos neste item, uma rede neural, mostrada na Figura 5.8, composta de 3 camadas, projetada e implementada, para o controle do processo de nível de líquidos de quatro tanques, realizado, discutidos no capítulo 2. Essa implementação, também, foi realizada no SIMULINK[®] do software MATLAB[®].

A arquitetura da rede neural que apresentou melhor desempenho na implementação do controle foi a arquitetura composta dos seguintes parâmetros: cinco entradas, quatorze neurônios na primeira camada escondida (função de ativação tangente hiperbólica), sete neurônios na segunda camada escondida (função de ativação tangente hiperbólica) e um neurônio na saída da rede neural (função de ativação logística), com termos de polarização e utilizamos o algoritmo de aprendizado de retropropagação, os valores treinados estão no apêndice.

As Figuras 5.9 e 5.10 apresentam a simulação do controle neural obtido no modelo da planta, utilizando o Matlab. A Figura 5.11 apresenta o diagrama do controlador neural de três camadas e a planta, utilizando a interface OPC. Na Figura 5.12, tem-se os sinais de controle para uma referência igual a 2 cm.

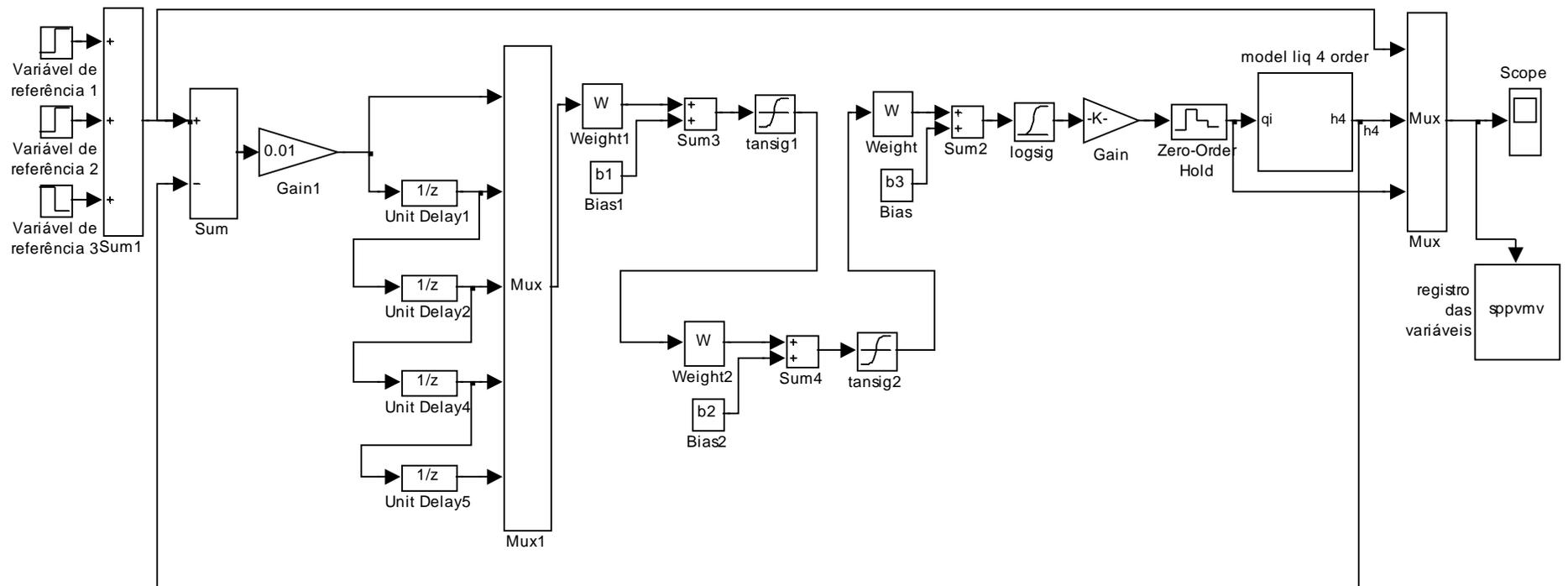


Figura 5.8 - Controlador Neural de três camadas, controlando o modelo da planta.

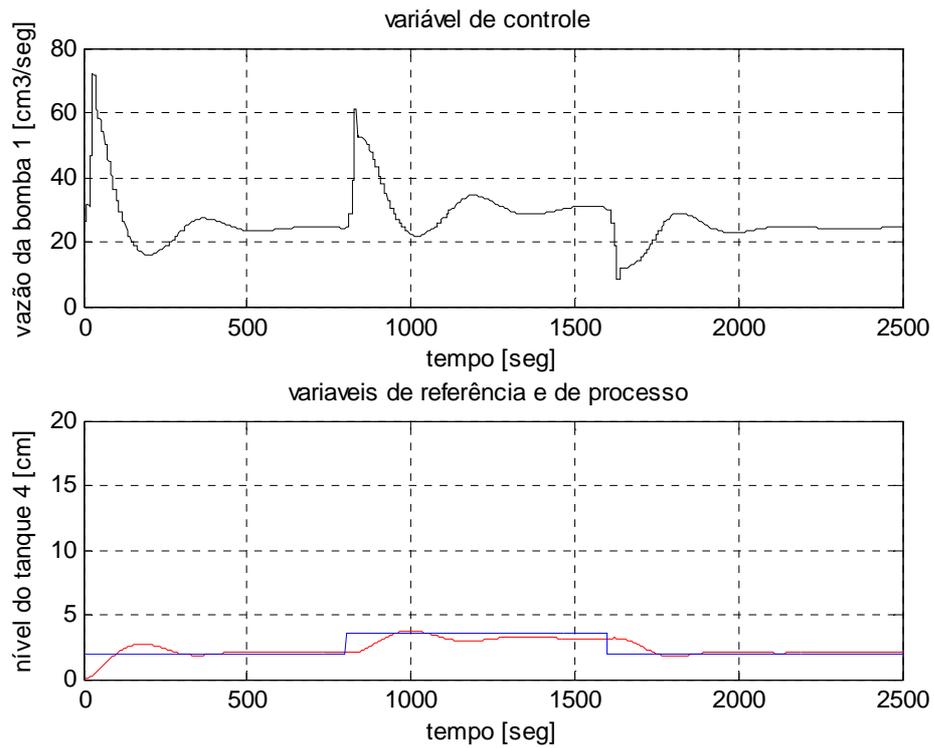


Figura 5.9 – Simulação do controlador neural (referências: 2 cm, 4 cm e 2 cm)

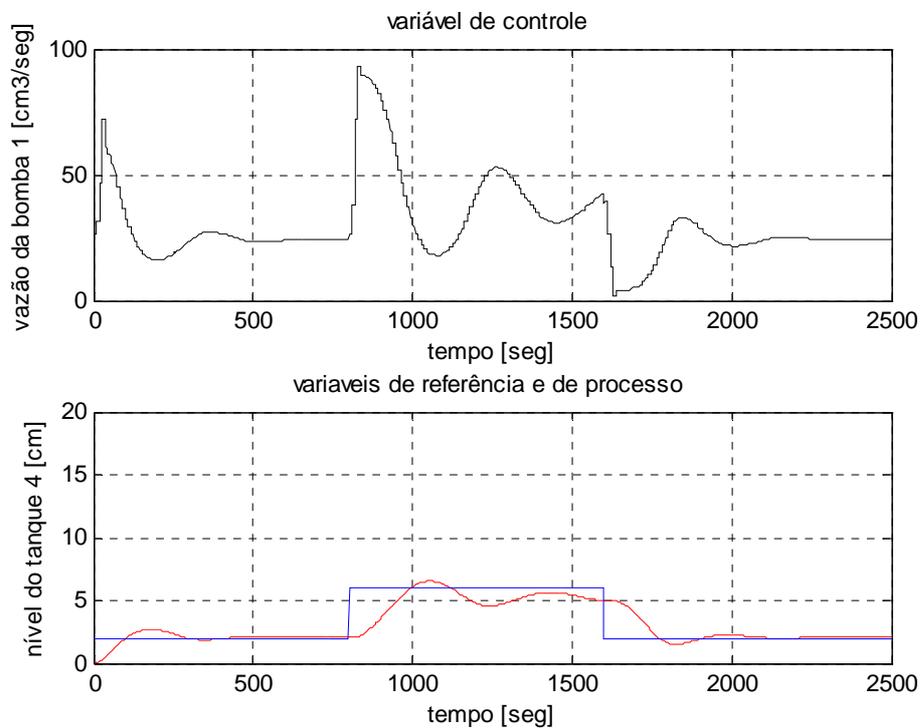


Figura 5.10 – Simulação do controlador neural (referências: 2 cm, 6 cm e 2 cm)

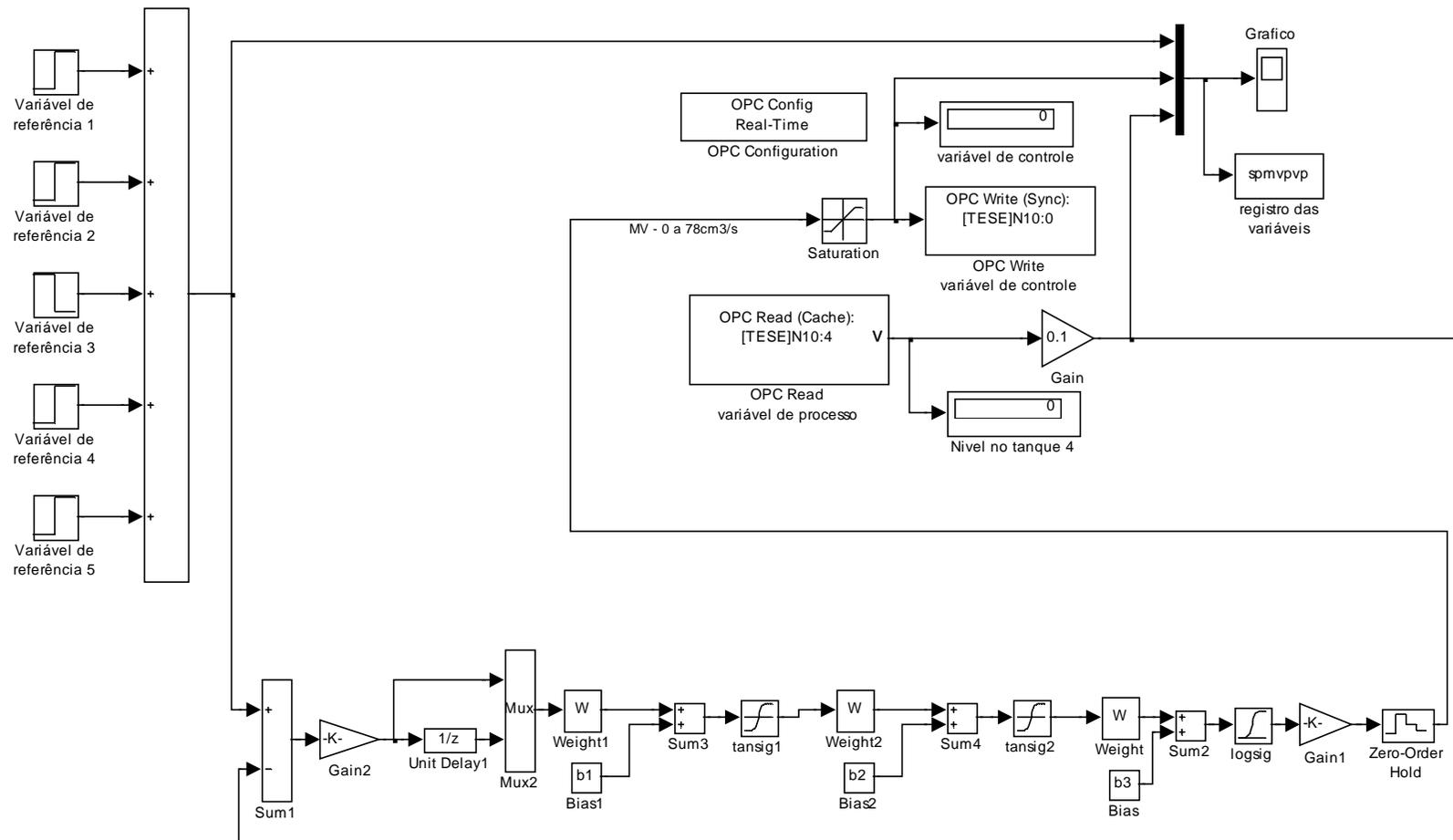


Figura 5.11 - Controlador Neural de três camadas, controlando a planta via interfaces OPC.

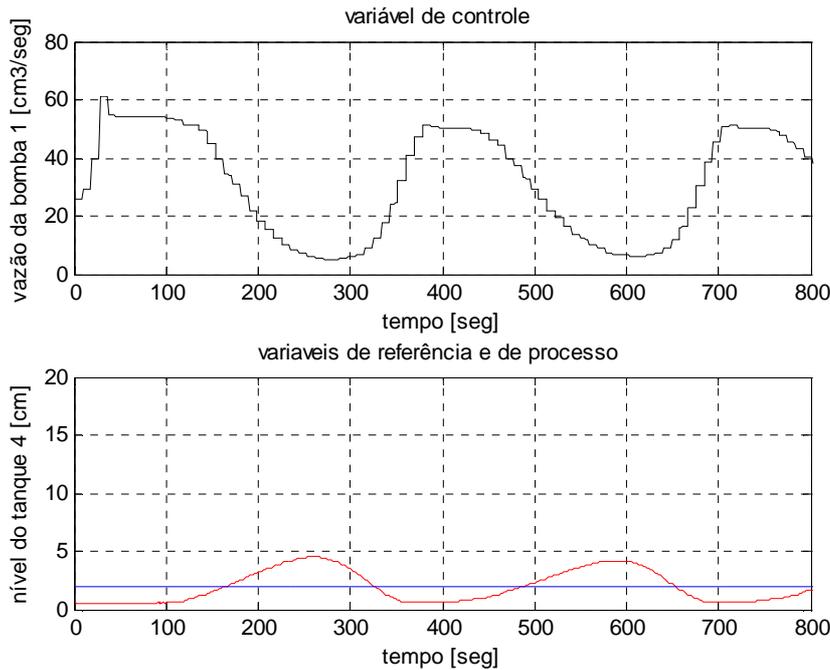


Figura 5.12 – Controlador neural (referência: 2 cm)

5.5 - COMENTÁRIOS

Estudos e experimentos foram projetados e implementados para que uma única rede neural, com mais entradas em atraso (u e y) possa ser treinada para “aprender” todos os pontos de operação do processo de nível de quatro tanques, mais rapidamente, produzindo um controlador não-linear fiel ao processo físico em questão.

Mas os resultados obtidos no processo mostram que o controlador neural implementado na planta não obteve um bom desempenho para alguns casos, conforme apresentado nas Figuras dos itens 5.3 e 5.4. Observa-se que a variável de processo apresentada na Figura 5.12 não acompanha a variável de referência, ultrapassando o tempo necessário para que a curva de resposta alcance valores em uma faixa de 5% (Ogata, 2003). Mesmo utilizando a viabilidade da abordagem neural usando pesos fixos para o controle.

Entretanto, nos experimentos, observa-se que as redes neurais deixam a desejar quando existem diferenças grandes na variável desejada e, também, muitos pesos na rede. Observou que resultam em muito tempo de treinamento, que a convergência é difícil para uma única solução nessa planta de quatro tanques acoplados.

6 - CONTROLADOR NARMA-L2

6.1 - FERRAMENTA NARMA-L2

O modelo aproximador NARMA-L2 (*Nonlinear Autoregressive-Moving Average-Norma-L2*) foi proposto por Narendra e Mukhopadhyay em 1997. O modelo NARMA-L2 pode ser usado com o modelo das plantas não-lineares. Tipicamente, o número de atrasos desta rede neural incrementa com a ordem da planta. O controlador NARMA-L2 procura transformar o sistema não linear em um sistema linear através do cancelamento das não linearidades.

Diversos modelos de equações diferenciais não lineares apropriados para a identificação foram apresentados em livros e artigos. Nos trabalhos de Billings, (1985), encontram-se diversas modelagens por equações diferenciais não lineares.

O controlador NARMA-L2 é um rearranjo da rede neural que modela o processo que é treinado *offline*. Diversas etapas devem ser seguidas para aplicar o controlador NARMA-L2 ao sistema.

A primeira etapa é treinar uma rede neural para representar a dinâmica do sistema. O modelo NARMA é representado pelo seguinte modelo, onde $u(k)$ é a entrada ao sistema, e $y(k)$ a saída do sistema, representado pela equação 6.3 e pela Figura 6.1:

O modelo usado para a implementação da planta é:

$$y(k + d) = N [y(k), y(k - 1), \dots], \dots, y(k - n + 1), u(k), u(k - 1), \dots, u(k - n + 1)]; \quad (6.1)$$

onde: $u(k)$ e $y(k)$ são as entradas e as saídas respectivamente.

O treinamento da rede Neural, durante a fase da identificação, é realizado a fim aproximar a função não-linear N.

Se o sistema segue uma trajetória desejada y_r da referência então o controlador não-linear deve ser da forma:

$$u(k) = G [y(k), y(k - 1), \dots, y(k - n + 1), y_r(k + d), u(k - 1), \dots, u(k - m + 1)]. \quad (6.2)$$

O treinamento da rede Neural (minimização do meio - erro quadrado) é criar a função de g do controlador.

$$\hat{y}(k+d) = f[y(k), y(k-1), \dots, y(k-n+1), \dots, u(k-m+1)] + g[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)].u(k) \quad (6.3)$$

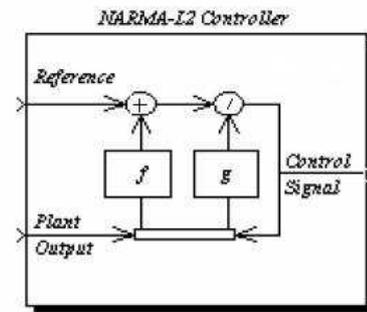


Figura 6.1 - Controlador NARMA-L2

Onde: y é a saída, d é o atraso, k é o tempo discreto, n são os atrasos de saída, m são os atrasos de entrada, u é a entrada e em que o atraso $d \geq 2$.

O controlador NARMA-L2 é dado por:

$$u(k) = \frac{y_r(k+d) - f[Y,U]}{g[Y,U]}, \quad (6.4)$$

$$u(k) = \frac{y_r(k+d) - f[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)]}{g[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)]} \quad (6.5)$$

O desempenho do controlador NARMA-L2 depende essencialmente da identificação do sistema com uma rede neural. Um bom modelo de uma rede neural para o sistema fornece bons resultados nos pontos de operação desejados.

O modelo NARMA-L2 é mostrado na Figura 6.2, onde os blocos TDL são linhas de atrasos que armazenam valores precedentes do sinal. As funções de ativação são a tangente hiperbólica e a linear para as duas camadas da rede.

NARMA-L2 é também uma ferramenta incluída na toolbox Neural Network do software MATLAB[®] que utiliza a teoria de redes neurais e identificação de parâmetros para realizar a identificação e o controlador.

A interface gráfica GUI NARMA-L2 é composta por três campos: arquitetura da rede, dados de treinamento e parâmetros de treinamento. A identificação inicia-se pela geração dos dados de treinamento que podem ser gerados pela própria ferramenta (*Generate Training Data*) ou importados de algum arquivo (*Import Data*). Define-se a topologia da rede (*Network Architecture*): a escolha do número de camadas, atrasos e intervalo de amostragem é feita baseada no sistema não-linear a ser controlado. Na etapa de identificação, é necessário escolher os parâmetros de treinamento (*Training Parameters*), que também serão definidos baseados na complexidade do sistema estudado e na capacidade computacional. Definidos os três campos, inicia-se o treinamento. O erro médio quadrático destes dados serve como critério de finalização para o treinamento. No término do treinamento, são fornecidas de forma gráfica as seguintes respostas: erro médio quadrático, erro quadrático, sinal de entrada, sinal de saída da planta e sinal de saída da rede.

Por fim, os parâmetros da rede neural obtida são aplicados ao bloco de controle NARMA-L2[®] (Figura 6.3). Nesta ferramenta o controlador utilizará a rede neural, ou seja, a própria rede será inserida na malha do sistema (realização de $f(\cdot)$ e $g(\cdot)$). O treinamento procura obter o modelo do sistema dinâmico.

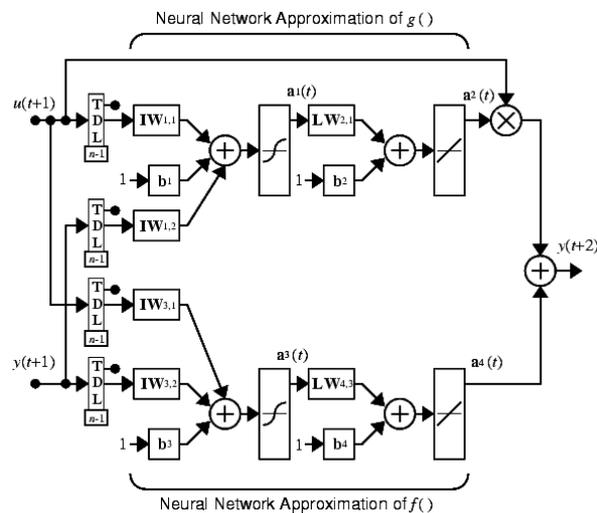


Figura 6.2 - Diagrama em blocos do modelo NARMA-L2 (Neural Network Toolbox, ML 2007)

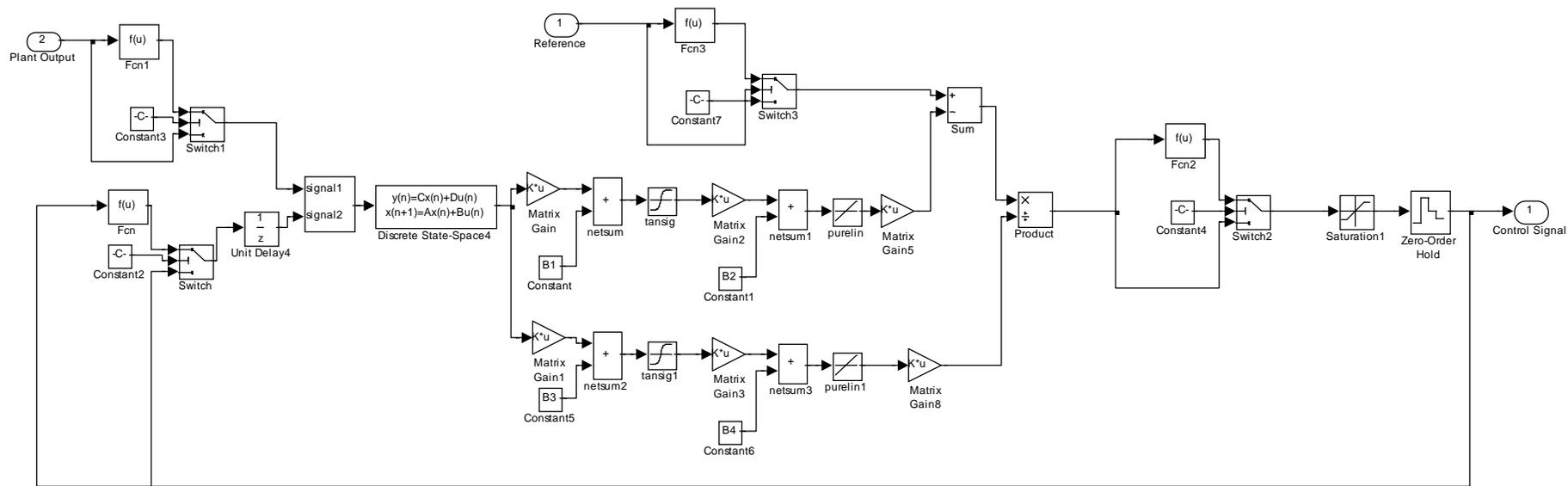


Figura 6.3 - Diagrama em blocos do controlador NARMA-L2 (Neural Network MLToolbox 2007).

6.2 - TAXA DE AMOSTRAGEM NO DESEMPENHO DO CONTROLADOR

A taxa de variação do sinal deve respeitar a taxa de amostragem da aquisição e o sinal deve ser adequado a faixa de entrada. O procedimento foi iniciado, encontrando-se o intervalo de amostragem para a determinação de amostras que representam o modelo desejado. Aplicou-se um degrau na entrada do sistema, caracterizando o tempo de resposta da planta. Por tentativa, a taxa de amostragem foi variada (2 a 10 seg), sendo o intervalo de amostragem que representa o melhor desempenho de controle para essa planta foi de 8 a 10 segundos. Aplicou-se um degrau na entrada do sistema.

O tempo de amostragem ou capturas por segundo, influencia significativamente no desempenho do controlador. Um valor pequeno, da ordem de um segundo para o processo de quatro tanques acoplados, apresenta um atraso no tempo de estabilização do sinal de saída, permitindo *overshoots* e *undershoots*. Já um aumento desse valor aumentaria a quantidade de informação obtida. O resultado da simulação foi obtido ao se estabelecer um limite máximo de $40[\text{cm}^3/\text{s}]$ e um mínimo de $0[\text{cm}^3/\text{s}]$ para o sinal de controle, com uma taxa de amostragem entre 8 a 10 segundos. Com essa limitação estabelecida, a simulação apresentou um *undershoot* com uma variação permitida nas especificações de desempenho. Isto implica que uma taxa de amostragem entre 8 a 10 segundos é admitida pelo controlador neural, fato desejável, para permitir sua implementação mesmo quando há limitação da frequência de amostragem do aparelho medidor dos sinais em questão.

6.3 - METODOLOGIA PARA A IMPLEMENTAÇÃO DO CONTROLADOR NARMA-L2

Definido a taxa de amostragem, a planta foi identificada pelos dados de treinamento que podem ser gerados pela ferramenta *Generate Training Data* do NARMA-L2. Feito isso, define-se a topologia da rede *Network Architecture*, escolhendo o número de camadas, atrasos e intervalo de amostragem. A escolha do número de camadas foi por tentativa, tendo como objetivo o projeto com a menor rede neural e observando-se o controle, principalmente a ação da variável de controle de forma que não permaneça por muito tempo no valor limite e que não venha a estressar o dispositivo responsável pela atuação na planta, que no nosso caso, é a bomba. A variável de processo deverá ser também,

acompanhada para que os tempos de subida, pico e sobre-sinal não venham a ter valores piores que um controle convencional. O número de neurônios mais adequado nos experimentos foi de 5. Tipicamente, o número de atrasos incrementa com a ordem da planta, para a nossa planta esse número foi de 4.

Após vários experimentos, a rede treinada foi aplicada ao controlador. Para se verificar a resposta do controlador utilizou-se a configuração descrita na Figura 6.4. Esse procedimento foi repetido para várias configurações da rede neural, como de algoritmo de treinamento, número de camadas e atrasos.

Durantes os experimentos foi verificado se o sinal de saída do processo acompanha adequadamente o sinal de referência e reflete satisfatoriamente o comportamento esperado na saída da planta. Observando, ainda, que o sinal de saída da rede neural atinge o regime sempre em valores de amplitudes de $q_i = 0$ a $76 \text{ cm}^3/\text{s}$. Todas as simulações dos diagramas de blocos descritos acima foram realizadas no ambiente SIMULINK® do software MATLAB®. Apresenta-se na Figura 6.4, o controlador no modelo do sistema de tanques,

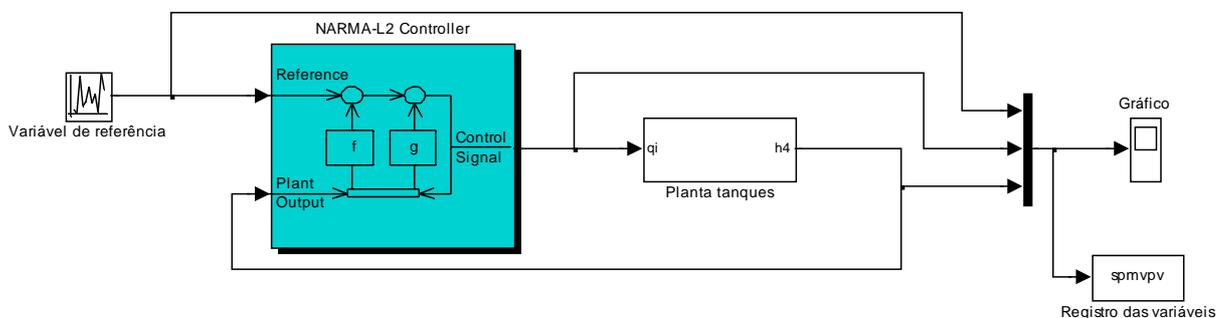


Figura 6.4 - Modelo do sistema de controle.

6.4 - O PROCEDIMENTO DE IDENTIFICAÇÃO UTILIZADO

Inicialmente, alguns dos trabalhos experimentais foram desenvolvidos nesta tese de doutorado utilizando a planta de três tanques acoplados do Laboratório de Automação, Visão e Sistemas Inteligentes – LAVSI (Gosmann, 2002). Estudos e pesquisas nesta planta de três tanques acoplados geraram-se experimentos que resultaram no artigo “Neural Control of a Non Linear Process With Gain Scheduling”, (Oliveira & Bauchspiess, 2004).

Este artigo apresenta a aplicação de controle neural com pesos selecionáveis pelo ponto de operação. Como processo crítico, que eventualmente deve operar por meses sem falhas, é necessário utilizar equipamento robusto. Assim o controlador neural foi implementado em controlador lógico programável (CLP), na linguagem *ladder logic* como ambiente de programação. Utilizando um controlador PID sintonizado em cada um dos quatro pontos da faixa de operação (30%, 40%, 50% e 60%), foram coletados dados para treinar uma rede neural, com uma estrutura definida, porém com diferentes conjuntos de pesos. No artigo referenciado apresentam-se resultados do controle neural e são discutidas as vantagens e restrições da implementação de redes neurais em controladores lógicos programáveis.

Para a continuação dos experimentos na planta de quatro tanques acoplados, foi necessário identificar a moto bomba para conhecer a sua dinâmica. A moto bomba foi acionada em 100% ou seja +5Vcc de controle via saída analógica, que corresponde +10Vcc no drive de potência, e sem nenhum engaste da régua, para representar k_{12} igual zero, para que nenhuma vazão flua para o tanque 2. Assim, obtive-se um tempo de 38 segundos para encher até 46 cm de altura no tanque 1, representando um volume de 2.760,0 cm³. Portanto a vazão neste experimento é de aproximadamente 72,6 cm³/seg. Ao escoar a água do tanque 1 para o reservatório de água obteve-se um tempo de esvaziamento igual a 41 segundos.

A moto-bomba u bombeia água do reservatório para o tanque 1, através de q_i , o tanque 2 e 4 possuem um dreno através de um furo de diâmetro de 4 mm, permitindo o dreno de água, pela gravidade para o reservatório através de q_{02} e q_{04} .

7 - CONTROLE NARMA-L2 NO PROCESSO DE LÍQUIDOS

7.1 - INTRODUÇÃO

Neste capítulo para a pesquisa, projeto de treinamento e implementação do controle neural simulado e do controle neural na planta, foi implementado o controlador NARMA-L2 em um computador comercial de uso geral. Para esse projeto de controle, utiliza-se o software Matlab/Simulink. A compatibilidade com os padrões elétricos e eletromecânicos industriais mínimos devem ser atendidas, aproximando ao máximo da situação real em chão de fábrica. Assim, esse computador processa a comunicação entre o ambiente Matlab/simulink e esses módulos analógicos por meio do usuário OPC.

Este trabalho investiga a aplicação do controlador NARMA-L2 em um computador comercial associado a uma planta com interface por meio de módulos analógicos instalados em controlador programável. Os experimentos apresentados neste capítulo possuem as seguintes aplicações para um processo de quatro tanques:

- a) Controle neural simulado usando o modelo de aproximação NARMA-L2, utilizando diversos algoritmos de treinamento implementados no modelo da planta utilizando o software Matlab/Simulink.
- b) Controle Neural na planta usando o modelo de aproximação NARMA-L2.

No controle lógico simulado, na implementação do controlador NARMA-L2, utilizou-se o modelo da planta (eq. 2.14 a 2.17). que gerou o artigo técnico com o título “Controle Neural de um Processo de Nível de 4ª Ordem”, (Oliveira & Bauchspiess, 2007, SBAI).

Já, para o controle neural, em outra implementação do controlador NARMA-L2, utilizou-se a planta, que gerou o artigo técnico com o título “Neural Control of a 4rd Order Level Process”, (Oliveira & Bauchspiess, 2007, COBEM). Artigo técnico publicado na ABCM Symposium Series in Mechatronics - Vol. 3 - pp.197-206, Copyright 2008 by ABCM ISBN 978-85-85769-38 (arquivo SSM3_I_22.pdf).

7.2 - TÉCNICAS DE CONTROLE

Neste processo, há dois problemas: o da regulação, ou seja, referência fixa e rejeição de perturbação, onde muitas vezes o processo é levado manualmente até o ponto de operação; e o seguimento de trajetória, onde ocorrem mudanças freqüentes no ponto de referência. Assim, serão abordados os dois problemas. Para garantir que o nível no tanque h_4 seja mantido nos valores desejados, diferentes estratégias de controle podem ser utilizadas. Nesta planta de quatro tanques acoplados, as seguintes técnicas de controle foram utilizadas, a saber:

- Controlador clássico PID (proporcional-integral-derivativo): corresponde à maior parte dos controladores utilizados nas indústrias.
- Controlador neural: algoritmo de aprendizado *back-propagation*.
- Controlador *Gain Scheduling* : este controlador é treinado em diversos pontos de operação de forma a representar diversos controladores lineares, explorando a utilização das redes neurais em um sistema não linear de forma que aprenda com o PID o controle de toda a larga faixa de operação desse tipo de processo.
- Controlador NARMA-L2: ferramenta para identificação de uma planta que transforma o sistema não linear em um sistema linear através do cancelamento das não linearidades.

A idéia básica inicial deste trabalho é explorar a utilização das técnicas de controle neurais, principalmente a utilização do controlador NARMA-L2, um sistema não linear de forma que produza um controle em toda a larga faixa de operação do tanque 4 desse tipo de processo. Observando que, como em todo sistema físico real, há presença de restrições no processo: vazão limitada nas bombas e nível limitado nos tanques. Entretanto, não são consideradas técnicas de controle com tratamento ativo das restrições, que evitem, por exemplo, que o líquido nos tanques transborde.

7.3 - O CONTROLADOR NARMA-L2 EM OPERAÇÃO

Apresenta-se neste item um controlador NARMA-L2, para o controle de um processo de quatro tanques acoplados, realizado no SIMULINK[®] do software MATLAB[®].

O diagrama em blocos do controlador NARMA-L2, apresentado na Figura 6.3, possui as entradas para as variáveis de referência e de processo e a saída para a variável de controle. Nessa aplicação, o controlador NARMA-L2, entra em operação, controlando o modelo do processo tendo sido treinado com diversos algoritmos. Utilizamos o modelo do sistema de controle conforme mostrado na Figura 7.1.

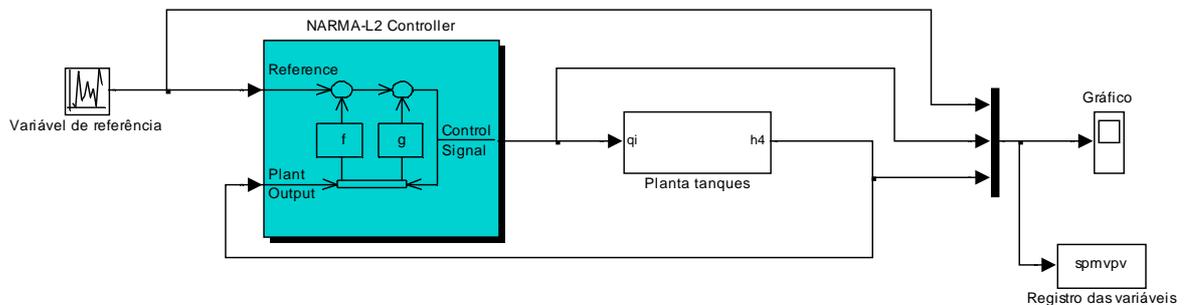


Figura 7.1 - Modelo do sistema de controle.

A abordagem de controle utilizada é tal que o controlador NARMA-L2 regule o nível de água do tanque 4 da planta de tanques acoplados, para uma variável de referência de nível desejada, por meio de uma variável de controle e com realimentação da variável de saída da planta.

Como descrito, foram obtidas respostas para várias configurações dos tanques acoplados de água. Para a apresentação dos resultados seguintes, utilizaram-se os seguintes parâmetros:

$$q_{i1} = 0 \text{ a } 66 \text{ cm}^3/\text{s} \text{ e } h_4 = 0 \text{ a } 14 \text{ cm};$$

$$k_{02}=5, \text{ e } k_{04}=5 \text{ cm}^{2,5}/\text{s};$$

$$k_{12} = 12, \text{ } k_{23} = 2,5, \text{ e } k_{34} = 6 \text{ cm}^{2,5}/\text{s}.$$

A planta foi identificada pelos dados de treinamento que podem ser gerados pela ferramenta *Generate Training Data* do NARMA-L2. Feito isso, definimos a topologia da rede *Network Architecture*, escolhendo o número de camadas, atrasos e intervalo de amostragem. Na teoria das classes dos algoritmos de treinamento (Figura 4.2), denomina-se como método de treinamento empírico (Haykin, 1999).

A Tabela 7.1 apresenta as respostas para vários algoritmos utilizados nos tanques acoplados de água, para várias configurações considerando o número de camadas, atrasos de entrada e de saída (*delay in* e *delay out*) e os erros quadráticos somados (SSE) de treinamento e validação para a identificação. O objetivo desta comparação é obter dados para avaliar o desempenho de cada algoritmo. Para isto utilizaremos uma rede neural composta: números de camadas e atrasos na entrada e na saída. Esta rede está organizada em diversas camadas. Representa-se a variação dos erros durante os diversos algoritmos usados na rede. Quando o erro mínimo de validação não apresentou mais redução significativa, o treinamento foi abortado, significando que a rede encontrou o máximo de fidelidade possível no mapeamento da função que se deseja que aprenda ou então que encontrou um mínimo local na paisagem de erro.

Vários testes foram realizados, com a preocupação de verificar o desempenho da excursão da variável de controle, verificando a baixa saturação dessa variável em relação a variável de processo para uma variável desejada. Assim, as Figuras 7.2(a), 7.2(b) e 7.2(c) mostra-se a simulação de controle no modelo da planta, que se obteve um melhor resultado.

As Figuras 7.3(a) e 7.3(b) mostram os sinais no modelo da planta, quando o controle obteve uma melhor excursão da variável de controle, com baixa saturação desta variável em relação a variável de processo para uma determinada variável desejada.

Tabela 7.1 - Respostas para vários algoritmos utilizados nos tanques

Algoritmo de treinamento	Número de neurônios	Atraso de entrada	Atraso de saída	Erro
trainlm	24	16	16	0.00160
trainlm	24	14	14	0.41661
trainlm	24	12	12	0.02894
trainlm	24	10	10	0.00148
trainlm	24	8	8	0.00047
trainlm	24	6	6	0.00094
trainlm	24	4	4	0.01716
trainlm	18	6	6	3.87861e-005
trainlm	18	4	4	0.00061
trainlm	16	4	4	0.0118349
trainlm	12	4	4	0.00544766
trainlm	9	8	8	0.00018
trainlm	9	6	6	0.00018
traincgb	24	16	16	0.455994
traincgb	24	14	14	0.488887
traincgb	24	12	12	0.249189
traincgb	24	10	10	0.0418296
traincgb	24	8	8	1.13997
traincgb	24	6	6	0.706627
traincgb	24	4	4	0.172294
traincgb	18	6	6	0.169614
traincgb	18	4	4	0.108574
traincgb	16	4	4	0.476335
traincgb	12	4	4	0.303887
traincgb	9	8	8	3.06086
traincgb	9	6	6	1.25626
traincgb	9	4	4	0.37180
traincgb	5	4	4	0.10040
traincgb	5	3	3	8.82390
traincgf	24	16	16	0.0939663
traincgf	24	10	10	0.552893
traincgf	18	6	6	0.985485
traincgf	12	6	6	0.635163
traincgf	12	4	4	0.426794
traincgf	9	4	4	0.515499
traincgf	5	6	6	2.69296
traincgf	5	4	4	1.23600
traincgp	24	16	16	0.532624
traincgp	18	10	10	0.814717
traincgp	18	6	6	0.238396
traincgp	12	6	6	0.710284
traincgp	12	4	4	0.774039
traincgp	8	4	4	0.547462
traincgp	5	6	6	0.954711
traincgp	5	4	4	0.575398

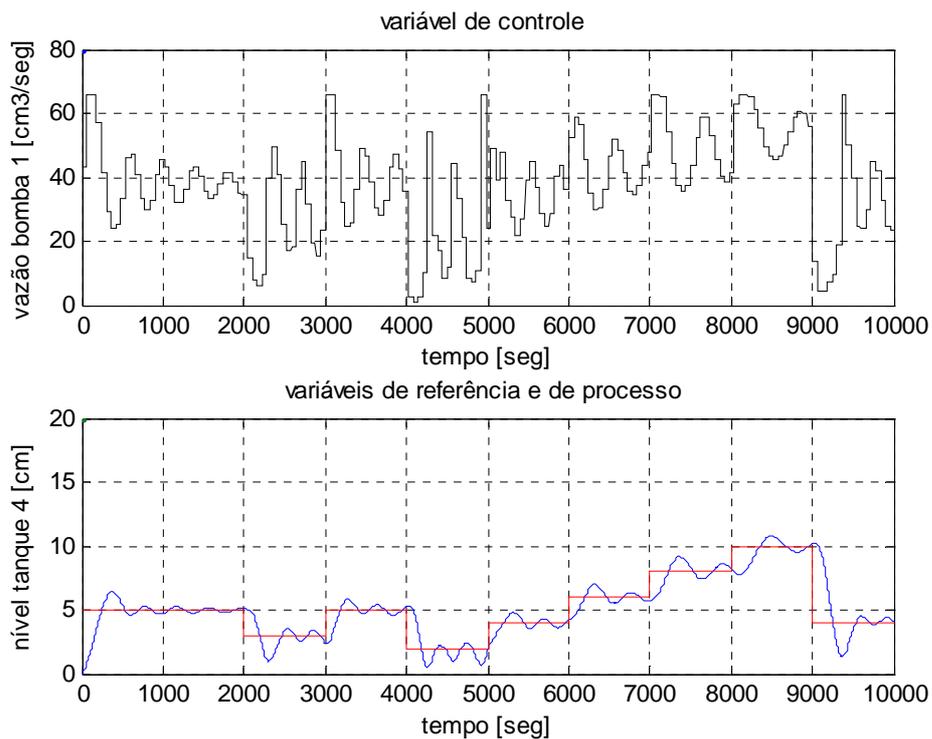


Figura 7.2(a) – Simulação no modelo da planta (referências: 5 cm, 3 cm, 5 cm, 2 cm, 4 cm, 6 cm, 8 cm, 10 cm e 4 cm), teste 1

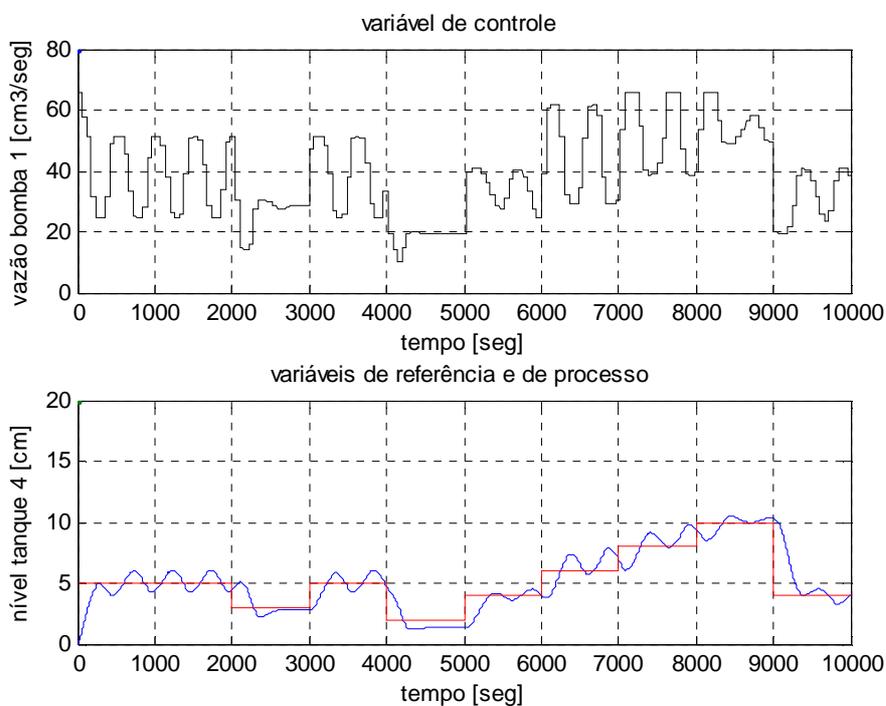


Figura 7.2(b) – Simulação no modelo da planta (referências: 5 cm, 3 cm, 5 cm, 2 cm, 4 cm, 6 cm, 8 cm, 10 cm e 4 cm), teste 2

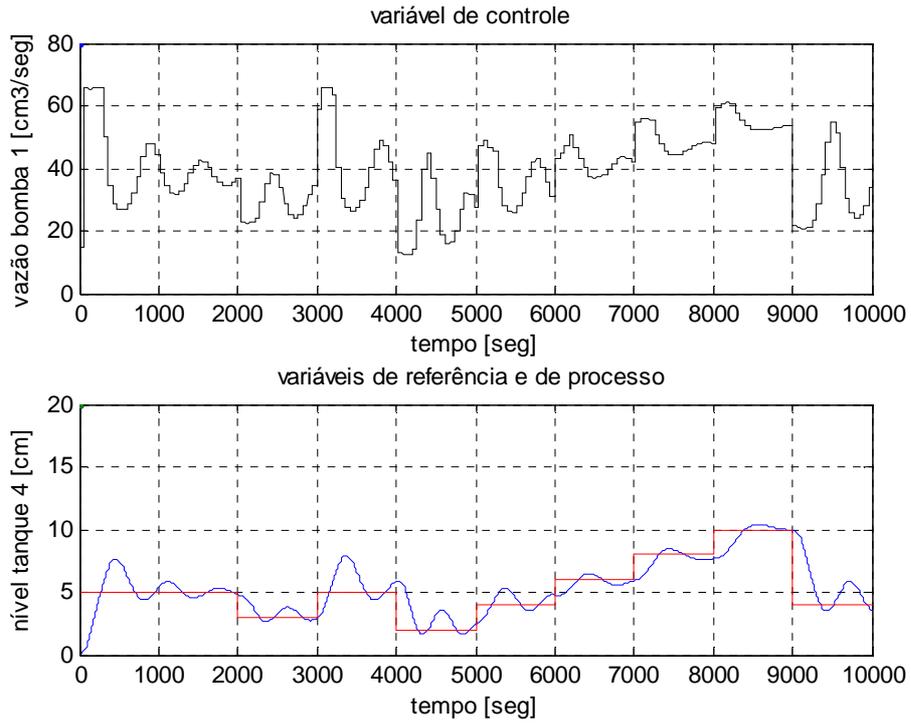


Figura 7.2(c) – Simulação no modelo da planta (referências: 5 cm, 3 cm, 5 cm, 2 cm, 4 cm, 6 cm, 8 cm, 10 cm e 4 cm), teste 3

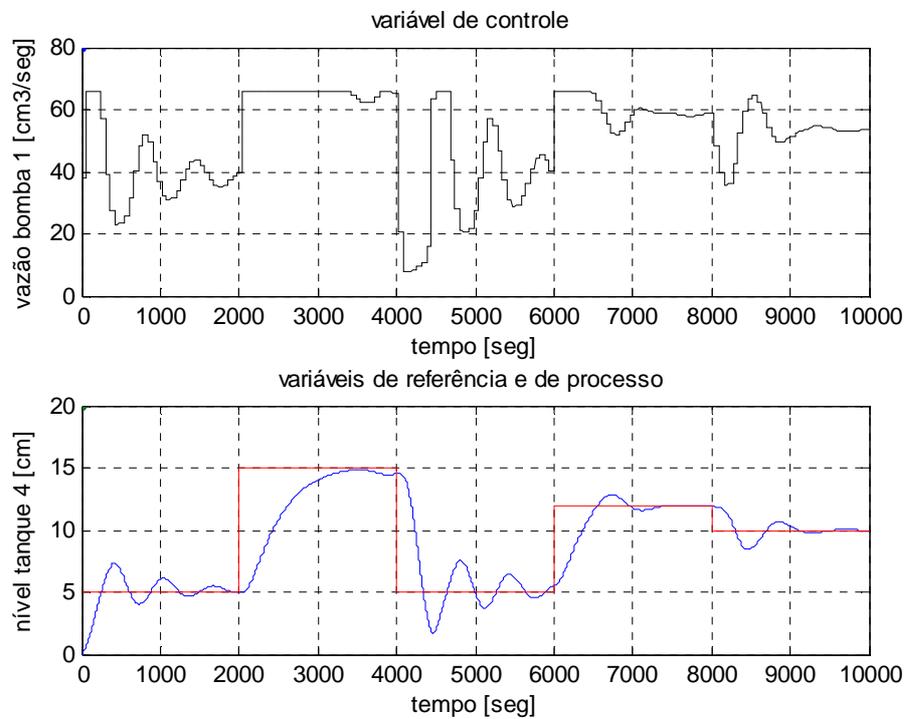


Figura 7.3(a) – Simulação no modelo da planta (referências: 5 cm, 15 cm, 5 cm, 12 cm e 10 cm)

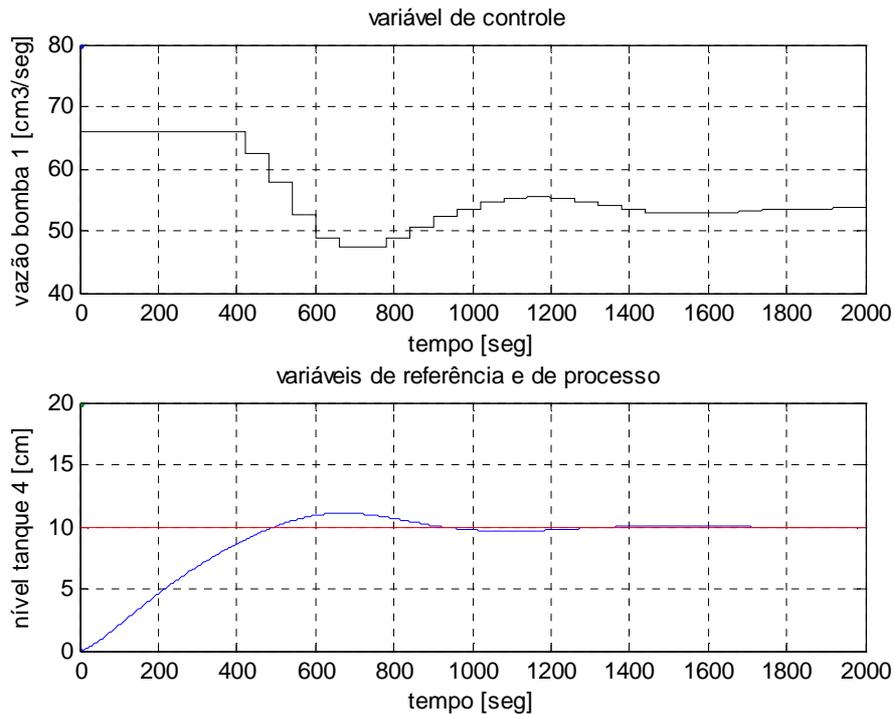


Figura 7.3(b) – Simulação no modelo da planta (referência: 10 cm).

Na Tabela 7.2 apresenta o tempo de resposta nos experimentos para o controlador NARMA-L2. Nessa tabela apresenta: o valor desejado, o tempo de subida da variável de nível em segundos, o tempo de pico da variável de nível em segundos, o sobre sinal da variável de nível em percentual e o tempo de acomodação em segundos que obteve um menor erro e uma melhor excursão da variável de controle para o processo de nível de líquidos.

Tabela 7.2 - Respostas para várias configurações dos tanques

Valor desejado h_4 (cm)	Tempo (seg): Subida (0 a 100%)	Tempo (seg): Pico	Sobresinal (%)	Tempo (seg): acomodação (+/-2%)
5 cm	251	416	46,5	1877
10 cm	498	672	10,96	1242
14 cm	1064	1225	2,08	1775

7.4 - NARMA-L2 CONTROLANDO O MODELO DO PROCESSO

Apresenta-se neste item, também, um controlador NARMA-L2, para um controle de modelo do processo de nível de líquidos, realizado no SIMULINK® do software MATLAB®. No item anterior, na Tabela 7.1 obtiveram-se diversas configurações para de controladores. Na aplicação deste item, o controlador NARMA-L2 controla o modelo do processo utilizando o controlador NARMA-L2 que apresentou um desempenho satisfatório da excursão da variável de controle com baixa saturação.

Na Figura 7.4, apresentam-se os sinais de controle, utiliza-se o modelo aproximado NARMA-L2.

O conjunto acoplado de quatro reservatórios de água (modelo do processo) foi identificado através do *Plant Identification - NARMA-L2*: Nas Tabelas 7.3, 7.4 e 7.5 mostram-se a arquitetura de rede, dados do treinamento e o erro, respectivamente. A taxa de amostragem foi discutida no item 6.2. Já na Tabela 7.6, mostra-se os tempos de respostas para a configuração usada para esse processo de quatro tanques.

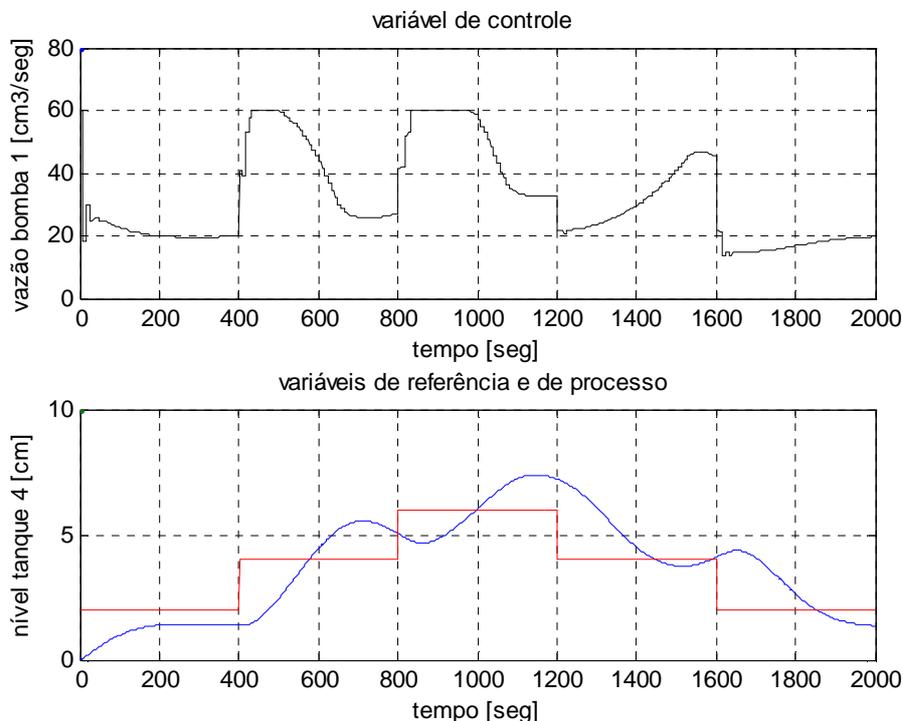


Figura 7.4 – Simulação de Controle (referências: 2 cm, 4 cm, 6 cm, 4 cm e 2 cm)

Tabela 7.3 - Arquitetura de rede - simulação

Arquitetura da Rede	
Tamanho da camada escondida	5
Entradas atrasadas da planta	4
Saídas atrasadas da planta	4
Intervalo de amostragem (seg)	8
Critério de parada do treinamento	número de épocas alcançado

Tabela 7.4 - Dados do treinamento - simulação

Dados do treinamento	
Amostragem	5000
Máxima entrada na planta (cm ³ /s)	60
Mínima entrada na planta (cm ³ /s)	0
Intervalo máximo	500
Mínimo intervalo	50
Épocas de treinamento	5000
Função de treinamento	<i>traincgf</i>

Tabela 7.5 - Resposta obtida pela configuração do processo - simulação

Número de neurônios	Atrasos na entrada	Atrasos na saída	Erro
5	4	4	1.236

Na Tabela 7.6, apresenta-se os tempos de resposta para o controlador NARMA-L2, realizados no ambiente Matlab e Simulink, para o modelo da planta. Nesta tabela apresentam-se: as mudanças do *setpoint* e os tempos de subida/descida, tempo de pico, sobresinal/subsinal e o tempo de acomodação. Não foi atingido o valor desejado de 0 a 2 cm, a variável de processo chegou a 1,4 cm.

Tabela 7.6 - Tempo de resposta da planta para o controlador NARMA-L2

Valor desejado h_4 (cm)	Tempo (seg): Subida/descida (0 a 100%)	Tempo (seg): Pico	Sobresinal/ Subsinal (%)	Tempo (seg): acomodação (+/-2%)
2 cm	203	256	2,1%	270
4 cm	176	312	38,6	398
6 cm	199	347	23,1	398
4 cm	249	316	6,51	374
2 cm	254	398	31,45	398

7.5 - NARMA-L2 CONTROLANDO O PROCESSO.

Apresenta-se neste item, também, o controlador NARMA-L2, para um controle do processo de nível de líquidos, por meio das informações obtidas do modelo realizadas no SIMULINK® do software MATLAB®. No item anterior, o controlador NARMA-L2 controla o modelo do processo de quatro tanques acoplados.

Depois de treinada, a rede neural é usada no controlador da planta. Para essa implementação foi usado o diagrama mostrado na Figura 7.6. No controlador neural NARMA-L2, foram usados os seguintes parâmetros:

$$q_{i1} = 0 \text{ a } 66 \text{ cm}^3/\text{s} \text{ e } h_4 = 0 \text{ a } 14 \text{ cm};$$

$$k_{02}=5, \text{ e } k_{04}=5 \text{ cm}^{2.5}/\text{s};$$

$$k_{12} = 12, \text{ } k_{23} = 2.5, \text{ e } k_{34} = 6 \text{ cm}^{2.5}/\text{s}$$

Os dados experimentais foram transferidos usando a conexão OPC (Figura 7.5). Os dados foram adquiridos dos sensores localizados nos tanques. Os sinais da variável desejada, de controle e de processo são transferidos e gravados na área de trabalho do MATLAB, em tempo real, a cada ½ segundo. A transferência de dados para o controlador programável é de sentido único de modo que o processo de nível de líquido não seja interrompido. O processo de transferência de dados é iniciado ao executar o controle NARMA-L2.

Esquema de registro de dados: Como representado na Figura 7.5, a transferência de dados entre computador (Matlab/Simulink) para os módulos analógicos de entrada e de saída foi realizada pelo toolbox OPC, através de um interfaceador de comunicação instalado no computador, via protocolo de comunicação RS232, entre o microcomputador e o controlador programável, responsável pelo funcionamento dos módulos analógicos. Na configuração deste interfaceador de comunicação podemos configurar o protocolo de comunicação.

No toolbox de OPC (Open Process Control) é uma coleção de funções que estende a potencialidade do ambiente computacional do MATLAB®, e os blocos que estendem o ambiente dinâmico da simulação do Simulink®. Usando o toolbox OPC, pode-se adquirir

dados em tempo real diretamente na área de trabalho do MATLAB, escrevendo e lendo dados diretamente do usuário OPC.

Na Figura 7.6, o controlador NARMA-L2, através do computador (a) processa a rede neural e transfere ou escreve o sinal de controle $u(k)$ para o módulo analógico de saída (b) que realiza o controle da vazão q_0 do tanque 1, por meio da bomba; (c) transferindo ou lendo o nível 4 do tanque h_4 , através do OPC/RS232 (Figura 7.5).

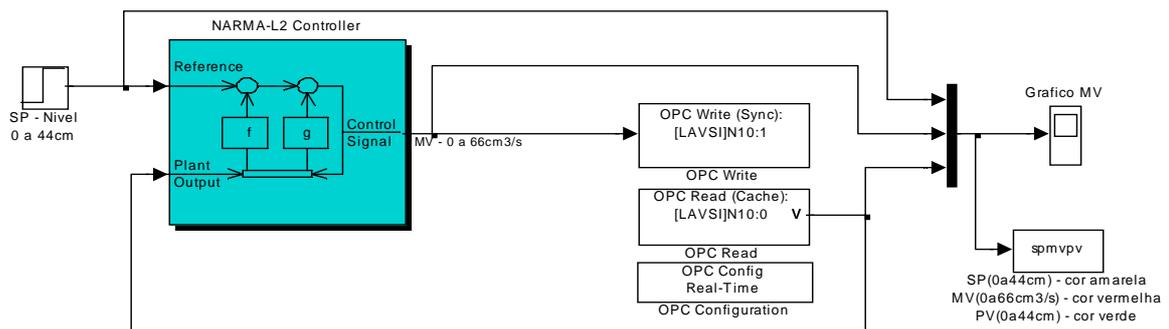


Figura 7.5 - Sistema de controle implementado para o controle da planta.

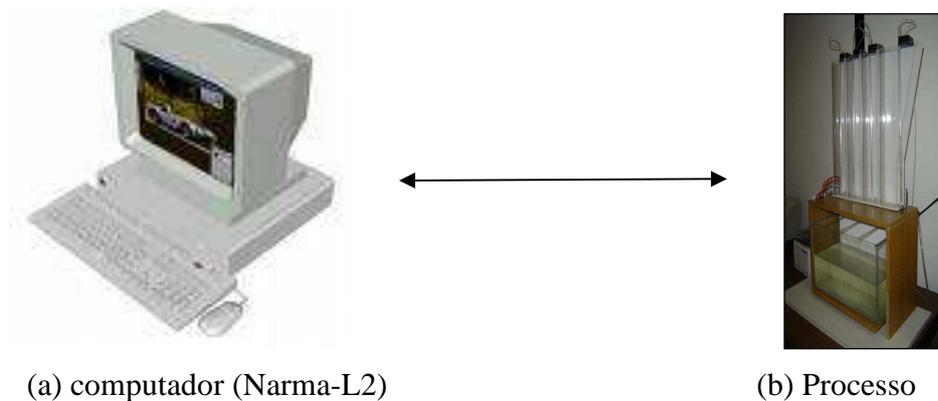


Figura 7.6 - Diagrama do sistema de comunicação entre o computador e o processo.

Na Figura 7.7, apresentam-se os sinais de processo, durante o controle NARMA-L2 e na Figura 7.8, apresentam-se os sinais de processo, durante o controle PID. Ambos controles aplicados no processo de quatro tanques acoplados, em diversos pontos de operações. A sintonia do controlador PID foi manual (tentativa e erro) apresentando dificuldades na sintonia na faixa de operação desejada.

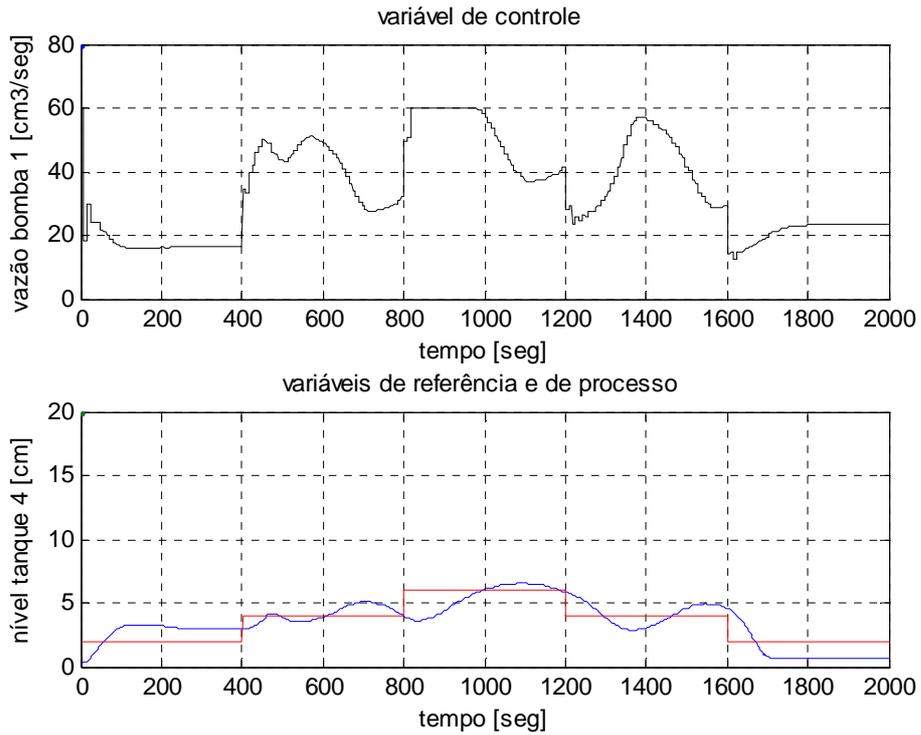


Figura 7.7 – Controlador NARMA-L2, sinais de processo (referências: 2 cm, 4 cm, 6 cm, 4 cm e 2 cm)

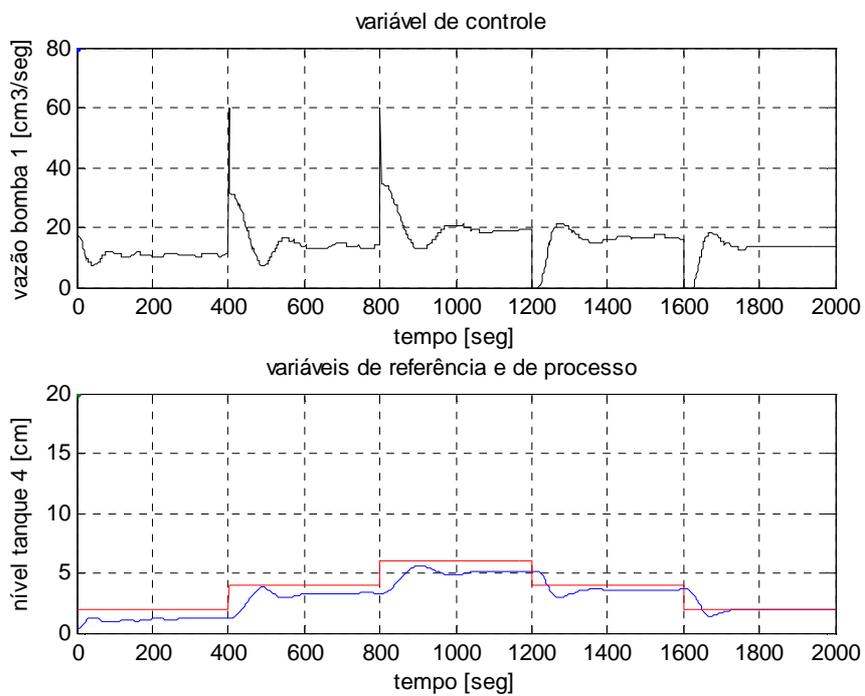


Figura 7.8 – Controlador PID, sinais de processo (referências: 2 cm, 4 cm, 6 cm, 4 cm e 2 cm)

7.6 - IDENTIFICAÇÃO DO PROCESSO PELO NARMA-L2.

Apresenta-se neste item o controlador NARMA-L2, para a identificação de um processo de nível de líquidos, realizado no SIMULINK[®] do software MATLAB[®].

Utiliza-se o modelo do sistema controlador, conforme mostrado na Figura 7.9, para o controle do processo, ilustrado na Figura 7.10.

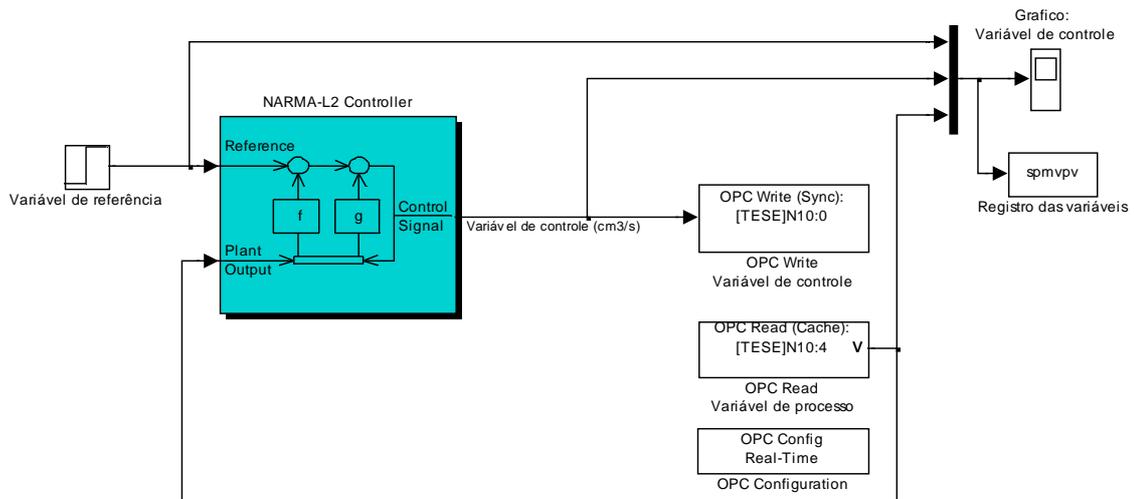


Figura 7.9 - Sistema de controle implementado no processo

A abordagem de controle utilizada é tal que o controlador NARMA-L2 identifique a planta de quatro tanques acoplados de água apresentada na Figura 2.1. Foram obtidas respostas para várias configurações dessa planta. Para a apresentação dos resultados gráficos, utilizaram-se os seguintes parâmetros: 'ks' para que uma relação entre os pólos mais rápido e mais lento fosse mínima, encontramos a relação de 14,22 para uma vazão constante de entrada de 15,96 cm³/s e pontos de operação: $h_1 = 8,06$ cm; $h_3 = 4,96$ cm; $h_2 = 3,09$ cm; $h_4 = 1,63$ cm, representando a função de transferência para o ponto de operação h_4 :

$$G(s) = \frac{7,43 \times 10^{-7}}{s^4 + 0,2637 s^3 + 0,02161 s^2 + 0,0005722 s + 3,595 \times 10^{-6}} \quad (7.1)$$

Os trabalhos foram iniciados com a identificação dos dados de treinamento que podem ser gerados pela ferramenta *Generate Training Data* do NARMA-L2. Feito isso, define-se a *topologia* da rede *Network Architecture*, escolhendo o número de camadas, *delays* e intervalo de amostragem, com base no melhor desempenho nas variáveis de processo e de controle.

Onde:

$$q_1 = 0 \text{ a } 78 \text{ cm}^3/\text{s}$$

$$h_4 = 0 \text{ a } 14 \text{ cm};$$

$$k_{02}=4,31$$

$$k_{04}=4,96 \text{ cm}^{2,5}/\text{s}$$

$$k_{12} = 9,06$$

$$k_{23} = 4,64$$

$$k_{34} = 5,26 \text{ cm}^{2,5}/\text{s}$$

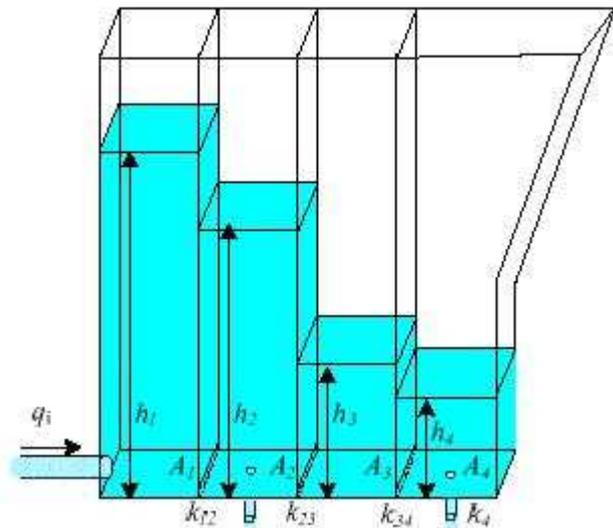


Figura 7.10 – Ilustração dos tanques acoplados

Dificuldades foram encontradas ao identificar essa planta utilizando o sistema de controle implementado da Figura 7.9, como o longo tempo envolvido para uma possível identificação baseada em dados aleatórios para a excitação da planta pelo NARMA-L2 e o intervalo de amostragem projetada.

Para obter-se sucesso, contorna-se o problema relativo ao longo tempo de experimento, recorrendo a uma estratégia de identificação do modelo da planta proposto na Figura 7.1. Então, as dificuldades apresentadas anteriormente em relação ao longo tempo envolvido nos experimentos e o intervalo de amostragem foram resolvidas.

Definiu-se a *topologia* da rede *Network Architecture*, escolhendo o número de camadas, atrasos, treinando a rede neural que seria utilizada pelo controlador NARMA-L2, e por fim realizar a simulação na planta utilizando o controlador NARMA-L2 até a obtenção de um

controle que apresenta uma melhor excursão da variável de controle com baixa saturação. Na Figura 7.11 mostramos os sinais de excitação da planta.

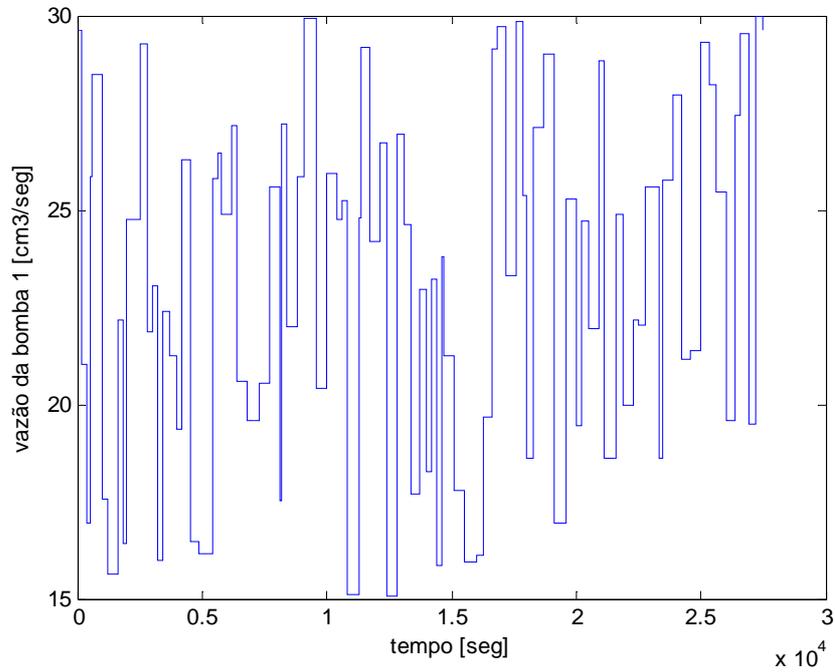


Figura 7.11 – Sinal gerado pelo NARMA-L2, para a excitação da planta.

O sinal de excitação da planta foi aplicado na planta, Figura 7.12, e obteve-se o sinal de identificação, apresentado na Figura 7.13.

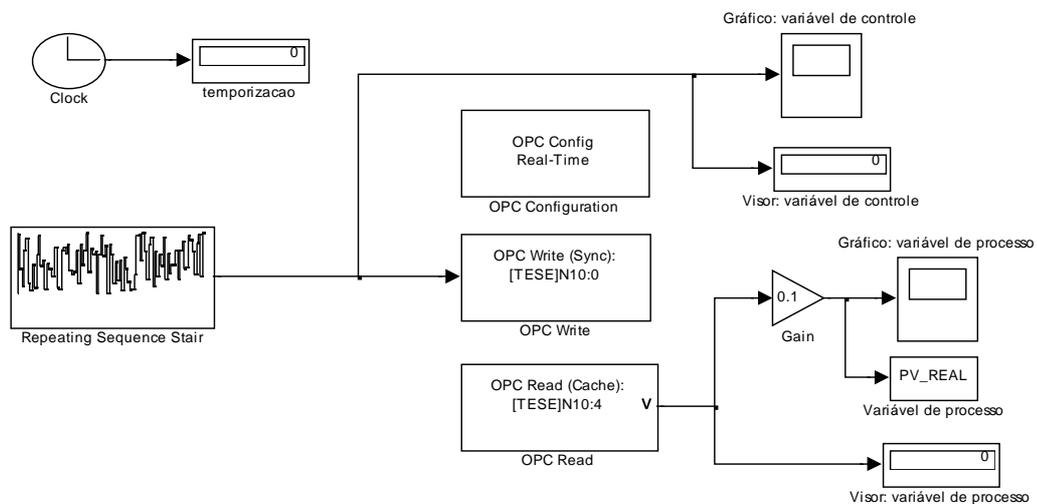


Figura 7.12 – Sistema implementado para a identificação da planta.

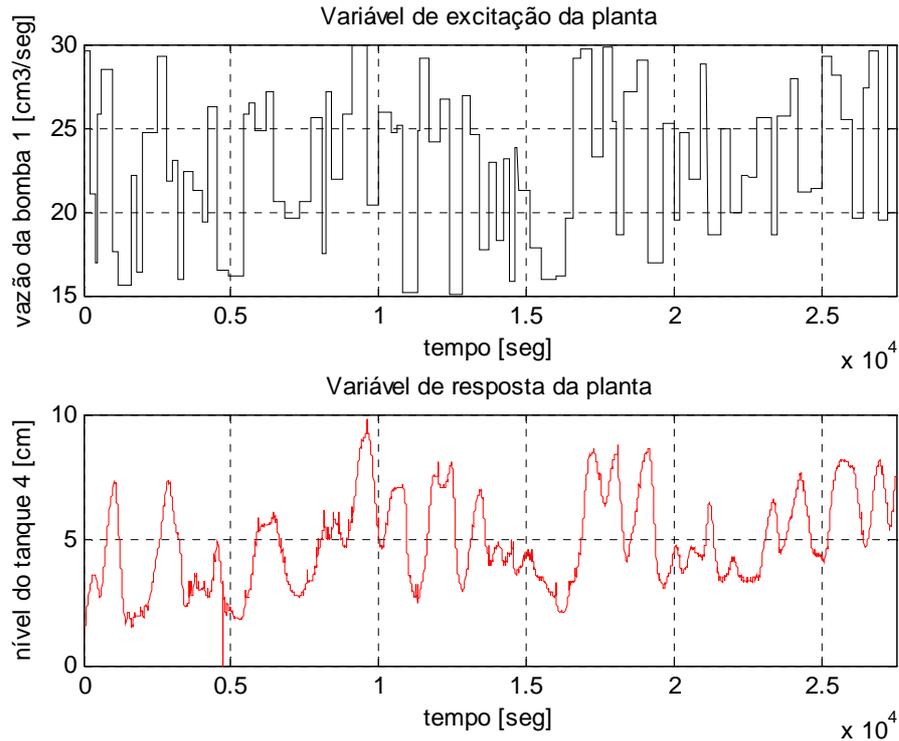


Figura 7.13 – Sinal identificado na planta

7.7 - TREINAMENTO DA RNA UTILIZANDO O CONTROLADOR NARMA-L2.

Nesse item, de posse dos vetores de entrada (*Input*) e de saída (*Planta Output*) da planta de tanques acoplados, demonstrado na Figura 7.14, parte-se para o treinamento utilizando o modelo do sistema controlado exposto na Figura 7.1, através do NARMA-L2.

O conjunto acoplado de quatro tanques de água (modelo do processo) foi identificado através do *Plant Identification* - NARMA-L2: Nas Tabelas 7.7 a 7.9, apresentam-se a arquitetura de rede, dados do treinamento e o erro, respectivamente. Estes são os parâmetros de treinamento da rede neural para o controlador NARMA-L2. Uma melhor rede é aquela que fornece o menor erro no conjunto de validação e aquela que fornece o mínimo erro para o conjunto de teste. A partir de certo número de passos, o desempenho do conjunto de teste pára de decair.

Na Figura 7.14, temos a visão dos gráficos gerados, durante o treinamento da rede neural no ambiente Matlab e Simulink.

Tabela 7.7 - Arquitetura da rede - identificação

Arquitetura da Rede	
Tamanho da camada escondida	5
Entradas atrasadas da planta	4
Saídas atrasadas da planta	4
Intervalo de amostragem (seg)	10

Tabela 7.8 - Dados do treinamento - identificação

Dados do treinamento	
Amostragem	180
Máxima entrada na planta (cm ³ /s)	40
Mínima entrada na planta (cm ³ /s)	0
Intervalo máximo	200
Mínimo intervalo	199
Épocas de treinamento	300
Função de treinamento	<i>traincgf</i>
Critério de parada do treinamento	número de épocas alcançado

Tabela 7.9 – Respostas obtidas pela configuração do processo - identificação

Número de neurônios	Atrasos na entrada	Atrasos na saída	Erro (rms)
5	4	4	0.47765

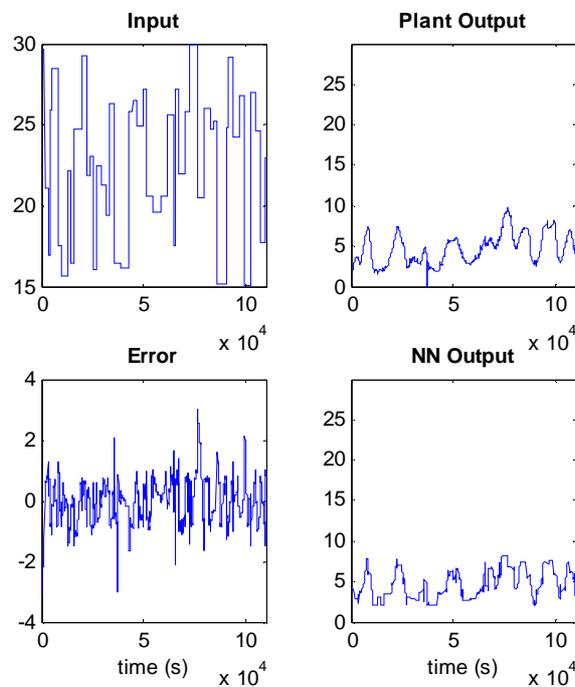


Figura 7.14 – Visão dos gráficos gerados no treinamento da rede neural

7.8 - NARMA-L2: SIMULAÇÃO E CONTROLE DA PLANTA

Conforme descrito no item anterior, os parâmetros para as configurações do conjunto acoplado de quatro reservatórios de águas foram obtidos, o controlador NARMA-L2 foram então implementado para controlar a planta. Na planta foram usados os seguintes parâmetros:

$$q_i = 78 \text{ cm}^3/\text{s}; \quad h_4 = 14 \text{ cm};$$

$$k_{02}=4,31 \text{ e } k_{04}=4,96 \text{ cm}^{2,5}/\text{s};$$

$$k_{12} = 9,06, \quad k_{23} = 4,64 \text{ e } \quad k_{34} = 5,26 \text{ cm}^{2,5}/\text{s}$$

Na Figura 7.15, apresenta-se uma simulação do controle NARMA-L2, com a rede neural treinada, e na Figura 7.16 apresenta-se o gráfico resultante dessa simulação.

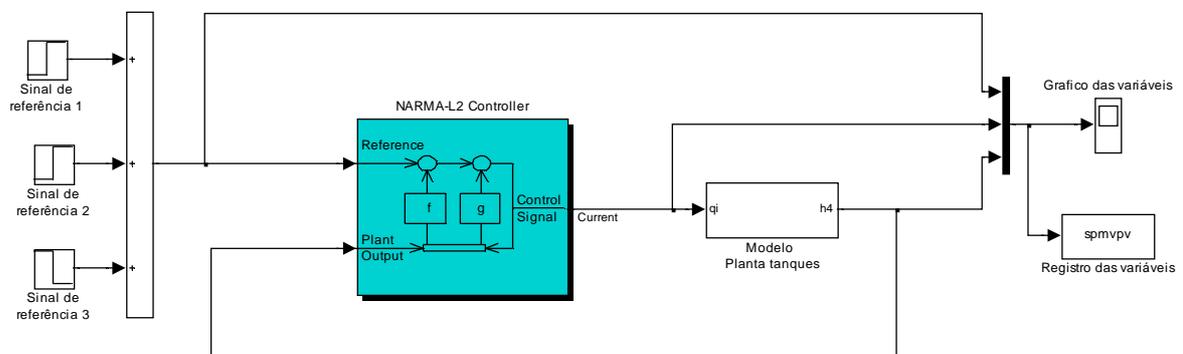


Figura 7.15 - Controlador NARMA-L2 (com a rede neural treinada) implementado no modelo da planta.

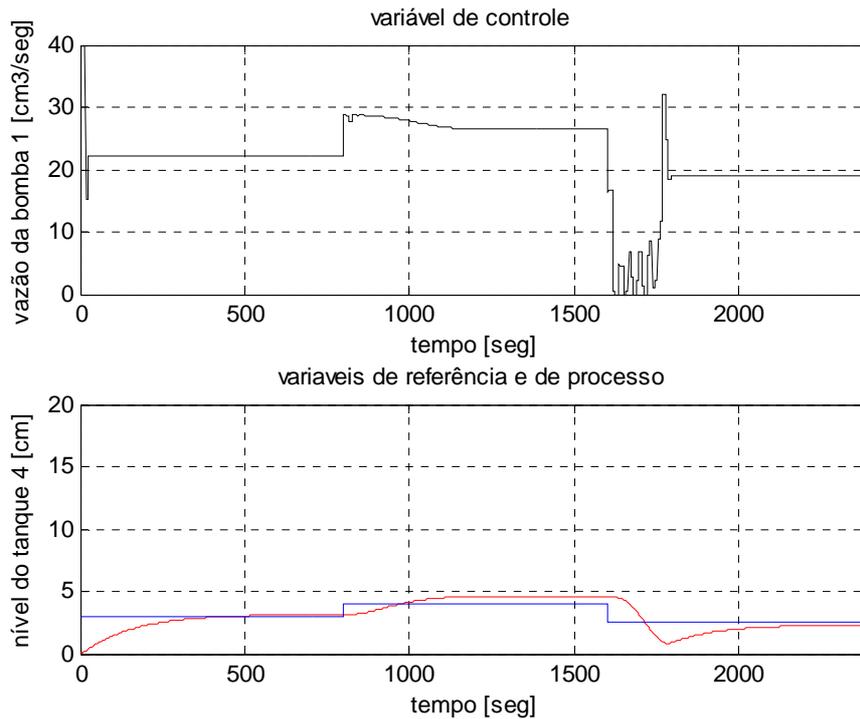


Figura 7.16 – Simulação do controlador NARMA-L2 treinado na planta (referências: 3 cm, 4 cm e 2,5 cm)

Na Tabela 7.10, mostra-se o tempo de resposta da simulação para o controlador NARMA-L2, realizados no ambiente Matlab e Simulink, para essa implementação. Apresentam-se as mudanças do valor desejado, os tempos de subida/descida, tempos de pico, tempos de sobresinal/subsinal e o tempo de acomodação.

Tabela 7.10 - Tempo de resposta da simulação do controlador NARMA-L2.

Valor desejado h_4 (cm)	Tempo (seg): Subida/descida (0 a 100%)	Tempo (seg): Pico	Sobresinal/ Subsinal (%)	Tempo (seg): acomodação (+/-2%)
3 cm	462	798	4,11	798
4 cm	176	415	14,54	798
2,5 cm	114	184	65,92	798

Agora, implementa-se o controlador NARMA-L2, descrito no item anterior, na planta, com o objetivo de verificar e analisar o controle já com a rede neural treinada. Apresenta-se na Figura 7.17 a implementação do sistema de controle NARMA-L2 (com a rede neural treinada) implementado na planta. Na Figura 7.18 apresenta-se o experimento com o valor desejado de 4 cm-5 cm-4 cm.

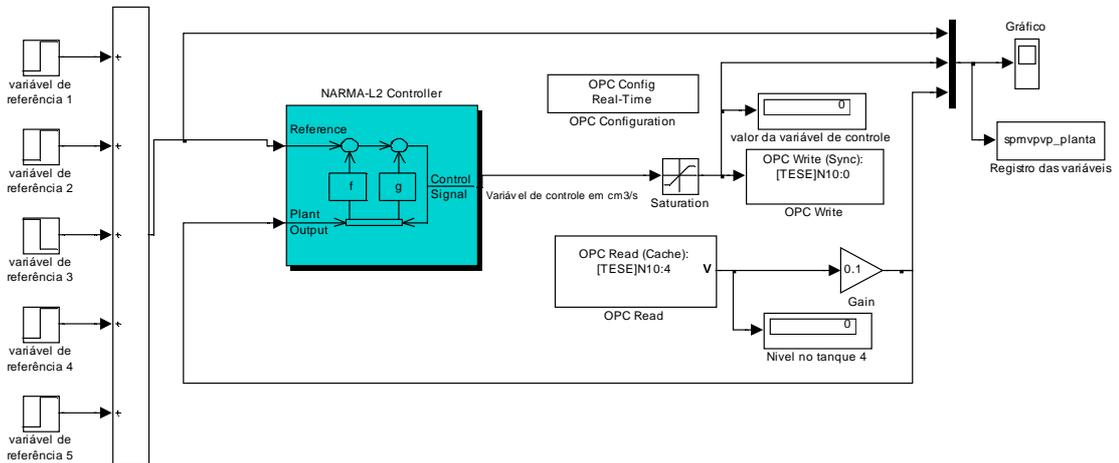


Figura 7.17 – Esquema de montagem para o controlador NARMA-L2, para a validação na planta de tanques.

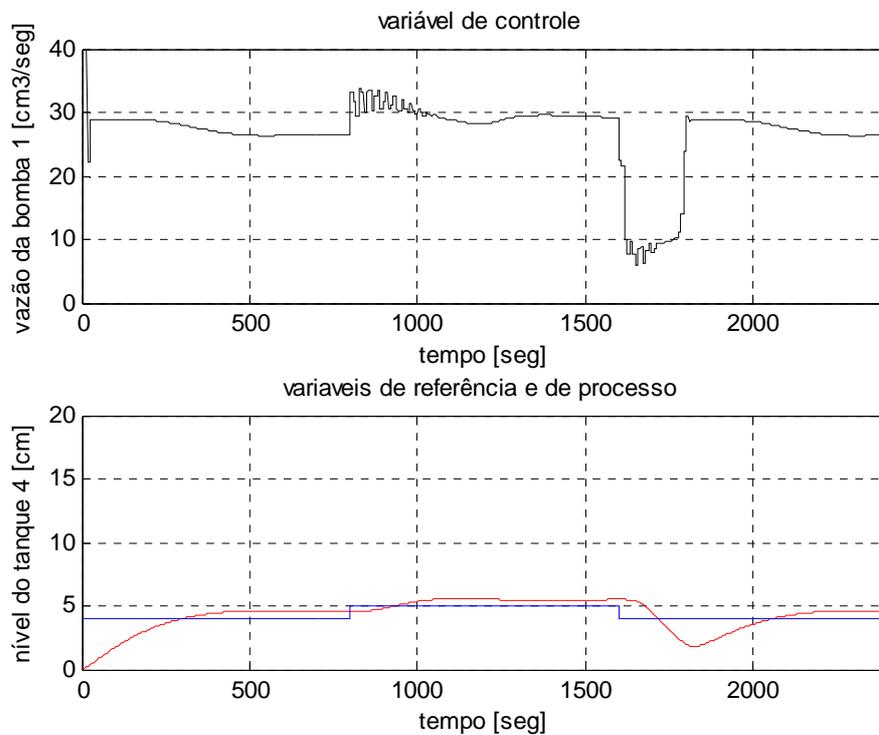


Figura 7.18 – Sinais obtidos na planta (referências: 4 cm, 5 cm e 4 cm)

Na Tabela 7.11, mostra-se o tempo de resposta para o controle realizado pelo controlador NARMA-L2, apresentados na Figuras 7.18, realizados no ambiente Matlab e Simulink, para a planta. Nesta tabela apresentam-se as mudanças do valor desejado e dos tempos de subida/descida, tempos de sobresinal/subsinal e o tempo de acomodação para o processo de nível de líquidos.

Tabela 7.11 - Tempo de resposta da planta – identificada e treinada.

Valor desejado h_d (cm)	Tempo (seg): Subida/descida (0 a 100%)	Tempo (seg): Pico	Sobresinal/ Subsinal (%)	Tempo (seg): acomodação (+/-2%)
4 cm	301	528	14,69	798
5 cm	134	336	12,24	798
4 cm	118	226	54,20	798

Utilizando novos parâmetros, como número de neurônios e atrasos de entradas e saída, conforme apresentados na tabela 7.12, a rede neural foi novamente treinada.

Tabela 7.12 – Respostas obtidas pela configuração para novos parâmetros

Número de neurônios	Atrasos na entrada	atrasos na saída	Erro
10	4	4	0.0531424

A simulação teve como objetivo o de verificar e analisar o controle já com a rede neural treinada. Na Figura 7.19, apresenta-se o gráfico resultante dessa simulação.

Na Tabela 7.13, mostra-se o tempo de resposta para o controlador NARMA-L2, realizados no ambiente Matlab e Simulink, para a simulação apresentada na Figura 7.19. Apresentam-se as mudanças do valor desejado, tempo de subida/descida, tempo de sobresinal/subsinal e tempo de acomodação.

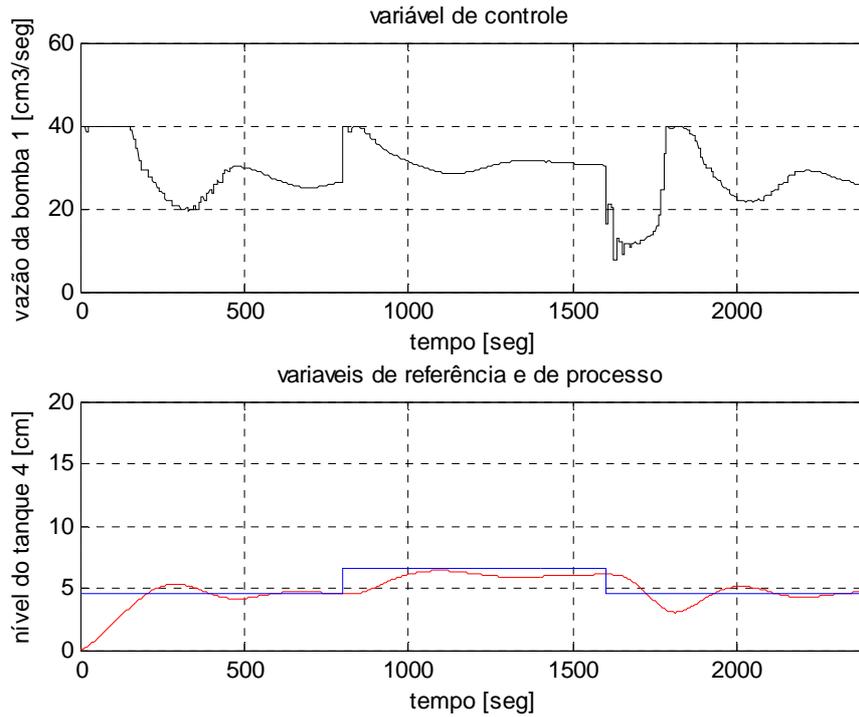


Figura 7.19 - NARMA-L2, treinado na planta e implementado no modelo da planta (referências: 4,6 cm, 6,6 cm e 4,6 cm)

Tabela 7.13 - Tempo de resposta da simulação para novos parâmetros.

Valor desejado h_4 (cm)	Tempo (seg): Subida/descida (0 a 100%)	Tempo (seg): Pico	Sobresinal/ Subsinal (%)	Tempo (seg): acomodação (+/-2%)
4,6 cm	209	291	15,56	707
6,6 cm	187	293	6,65	411
4,6 cm	124	212	33,96	701

Implementando-se o controlador NARMA-L2 na planta, apresenta-se na Figura 7.20 o gráfico obtido para os valores desejado de 5 cm- 6 cm- 5 cm. Na Tabela 7.14, mostrou-se os tempos de respostas para o controle realizado pelo controlador NARMA-L2. Nesta tabela apresentam-se as mudanças do valor desejado, tempos de subida/descida, tempos de sobresinal/subsinal e os tempo de acomodação para o processo de nível de líquidos.

Tabela 7.14 - Tempo de resposta da planta para novos parâmetros.

Valor desejado h_4 (cm)	Tempo (seg): Subida/descida (0 a 100%)	Tempo (seg): Pico	Sobresinal/ Subsinal (%)	Tempo (seg): acomodação (+/-2%)
5 cm	346	425	15,21	798
6 cm	168	168	1,78	798
5 cm	125	199	13,04	798

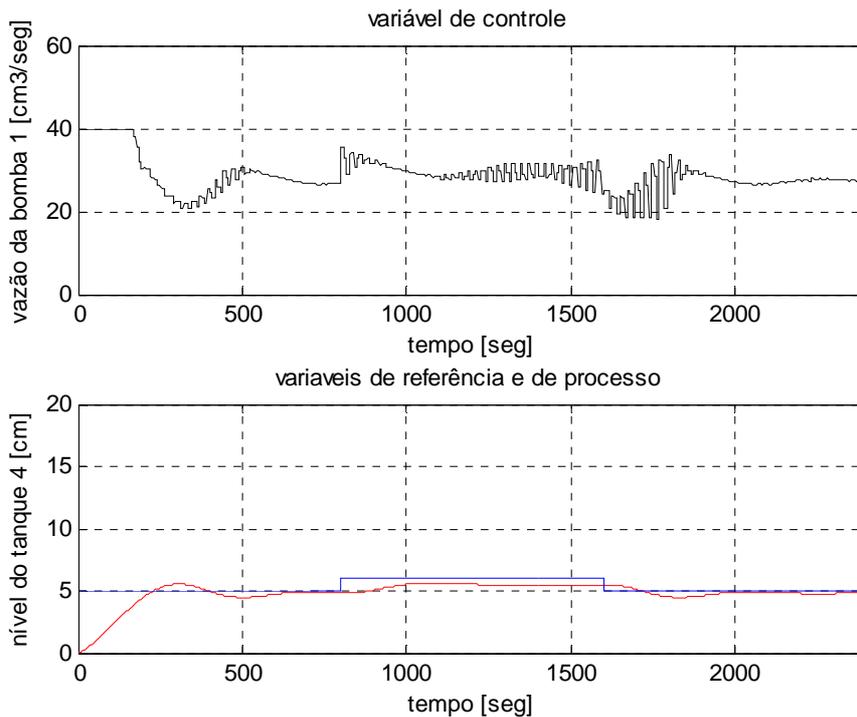


Figura 7.20 – Sinais obtidos na planta (referências: 5 cm, 6 cm e 5 cm)

7.9 - COMENTÁRIOS

Apresentamos neste capítulo, o experimento de um controlador NARMA-L2 controlando um processo de quatro tanques acoplados. A planta foi identificada, treinada a rede neural para o controlador NARMA-L2 e foi controlada através do controlador NARMA-L2.

Após a identificação do modelo da planta e treinamento da rede neural para o controlador NARMA-L2, via controlador NARMA-L2, no ambiente Matlab® e Simulink®, as informações extraídas dos resultados obtidos apresentaram desempenho pouco satisfatório (Figuras 7.4, 7.7 e 7.8). Os sinais obtidos pela simulação, Figura 7.16 e

os sinais obtidos pelo controle, Figura 7.18 não foram satisfatórios. A capacidade de manter a variável controlada próximo ao valor desejado foi insuficiente, mesmo em presença de alteração do valor de referência, Tabelas 7.10 e 7.11. Observa-se que o comportamento da variável de controle é bem suave e sem alterações bruscas, este comportamento é positivo para o processo, pois preserva a moto-bomba, não comprometendo a durabilidade e a manutenção preventiva dessa bomba.

O propósito desta tese de doutorado foi a de projetar e implementar uma rede neural no processo de quatro tanques acoplados de forma a controlar o nível do tanque 4 em toda a sua faixa de operação. Em alguns casos, o projeto do controlador neural funcionou satisfatoriamente, conforme apresentados nesta tese e em artigos técnicos publicados, mas em outros casos o controlador neural apresentou desempenho pouco satisfatório, sendo inviável a aplicação do controlador neural nesse processo, pois a rede neural utilizada não teve a capacidade de levar a variável controlada para próximo do valor desejado.

8 - CONCLUSÕES

Neste trabalho apresentam-se técnicas de identificação, treinamento e implementação de controle neural, utilizando redes neurais, para um processo de quatro tanques de água acoplados, realizado no SIMULINK[®] do software MATLAB[®], em um computador comercial de uso geral, cuja transferência de dados foi realizada pela toolbox OPC para os módulos analógicos.

Destacaram-se os aspectos do *hardware* do processo envolvidos para o controle. Estes aspectos são importantes para as várias fases que ocorrem durante o desenvolvimento deste trabalho. Vimos que se podem controlar os níveis dos tanques acoplados, por meio da variação da tensão na moto-bomba. Também, por meio do modelo não-linear da planta de tanques acoplados, podemos realizar simulações do processo reduzindo o tempo de bancada. Com o objetivo da linearização desse processo de quatro tanques acoplados, utilizamos as redes neurais, principalmente devido aos pólos do processo linearizado ser muito diferentes para os diferentes pontos de operação, que é uma característica intrínseca deste sistema não linear.

Destacaram-se, principalmente, os aspectos de aplicações experimentais de técnicas de controle para um processo de quatro tanques acoplados: o controlador NARMA-L2, identificando e controlando a planta. Esta aplicação foi implementada com o objetivo de controlar a altura de nível do quarto tanque desse processo descrito por modelo dinâmico não-linear.

Observou-se que, para o processo de quatro tanques acoplados, o controle neural utilizado apresentou desempenho pouco satisfatório, fato que pode ser atribuído à complexidade, oscilações do tipo *stick-slip* da planta e à implementação do NARMA-L2 na toolbox do Matlab que utiliza a rede neural com apenas uma camada escondida.

Como principal contribuição desta tese de doutorado destaca-se o estudo de avaliação de controle neural de um processo não linear composto de quatro tanques de água acoplados entre si. Vários experimentos de controladores são implementados, objetivando a pesquisa, atuação, validação e demonstrações de controle neural ainda não realizado no meio acadêmico, conforme pesquisa bibliográfica extensa realizada, para esse tipo de

planta. Também, destacam-se as publicações de artigos durante o doutorado, contendo a descrição de contribuições em controle neural.

Ressalta-se a avaliação de controle neural de processo físico real, servindo de referência para estudos futuros. Sendo de grande valia acerca da formulação empregada na modelagem da planta, e também com os gráficos demonstrando as situações de controle.

Com os resultados deste trabalho, as técnicas de controle e os experimentos realizados em um processo de quatro tanques, que requer uma implementação avançada de controle inteligente, revelam-se bastante interessantes, mostrando os diversos tipos de controladores neurais realizados, podendo ser uma referência para trabalhos futuros mais avançados na área de controle de processos.

Como sugestões para continuidade da pesquisa apresentada neste trabalho, pode-se proceder a um estudo mais aprofundado das técnicas de controle e metodologias aplicadas neste trabalho no tocante aos seguintes itens:

a) Controladores utilizados: pode-se investigar melhor a lei de controle utilizada nesta aplicação a fim de se encontrar uma expressão que possibilite uma melhor convergência da variável de controle; teste de implementação do NARMA-L2 utilizando a rede neural com mais de uma camada escondida;

b) Metodologia de ajuste de parâmetros: podem-se investigar outras estratégias de ajuste de parâmetros da RNA, ou utilizando mais recursos computacionais avançados de forma a ampliar os limites utilizados pelo algoritmo;

c) Utilizar outras topologias e outros algoritmos de aprendizado de redes neurais diferentes das implementadas nessa tese, que faça o treinamento e modele bem esse processo e ao montar a estrutura de controle faça a realimentação funcionar bem, obtendo um melhor desempenho;

d) Alteração do processo de quatro tanques acoplados: a altura do tanque 1 influencia no controle dos demais tanques, no que tange ao volume de água transferido aos demais tanques e a abertura entre os tanques, o que limita a vazão de água entre os mesmos. Para um sistema de uma entrada e uma saída, um determinado nível de água no tanque 4 exige uma altura bem maior no tanque 3, por seguinte no tanque 2, e refletindo no tanque 1. Segue na Figura 8.1(a), o processo atual e na Figura 8.1(b) uma sugestão bem interessante

de um novo processo. Esse mesmo processo sugerido poderia ser usado normalmente para a variante MIMO, ou seja, duas entradas juntamente com duas saídas.

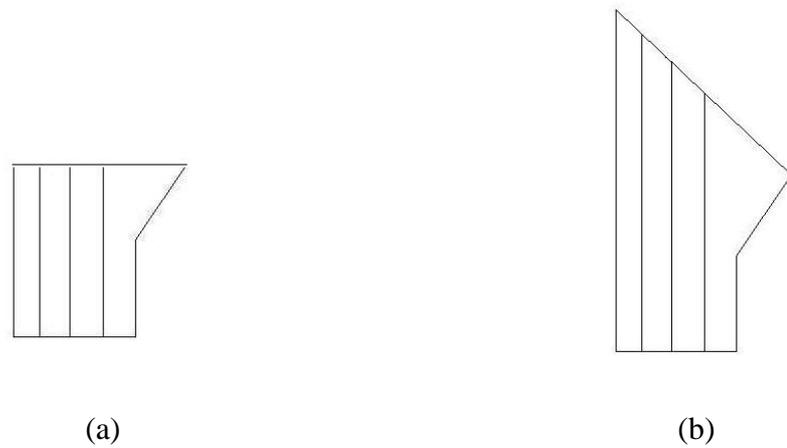


Figura 8.1 – Processo de quatro tanques acoplados:
(a) processo atual; (b) processo sugerido

Os resultados apresentados mostram que as redes neurais utilizadas podem ser consideradas controladores dinâmicos não-lineares. Apesar de o controle via rede neural utilizado apresentar um desempenho pouco satisfatório, a rede neural é uma ferramenta válida para a identificação de um processo a partir de dados experimentais. A adoção de estratégias avançadas de controle, ainda que apresente vantagens, é trabalhosa e requer muitos procedimentos prévios e um planejamento detalhado para levar a uma implementação exitosa do controle neural.

REFERÊNCIAS BIBLIOGRÁFICAS

- Aguirre, L. A., “Introdução à Identificação de Sistemas: Técnicas lineares e não-lineares aplicadas a sistemas reais”, Ed. UFMG, 2000.
- Andre L. Maitelli, Francisco G. Júnior, José S. B. Lopes. E Luiz A. Guedes, “Metodologia para Re-Sintonia de Controladores PID Industriais”, Salvador-BA, CBA 2006.
- Battiti, R.; “First- and second-order methods for learning: between steepest descent and Newton's method.”, *Neural Computation*, v.4 n.2, p.141-166, March 1992.
- Bauchspiess, A., “Notas de aula da disciplina Tópicos em Controle e Automação”, UNB, 2006.
- Bazaraa, M.S., Sherali, H.D., Shetty, C.M., “Nonlinear Programming: Theory and algorithms.” 2.ed. New York: John Wiley & Sons, 638p, 1993.
- Billings, S.A. “Introduction to Nonlinear Systems Analysis and Identification”, in K. R. Godfrey and R. P. Jones (eds) *Lecture Notes in Control Systems, Signal Processing for Control*, SERC Vacation School, Springer-Verlag, pp. 262-294, 1985.
- Billings, S.A. “An Overview of Nonlinear Systems Identification”, in *Proc. Of the 7 th IFAC Symposium in System Identification and System Parameter Estimation*, pp, 1985.
- Bishop, C. M., “Exact calculation of the Hessian matrix for the multilayer perceptron”, *Neural Computation*, 4(4), 494–501, 1992.
- Braga, A.; Carvalho, A.; Ludermir, T. "Redes Neurais, Artificiais: Teoria e Aplicações", Livro Técnico e Científico, Rio de Janeiro, 2000.
- Carlos A.R.P., "Redes Neurais para prevenção de Inadimplência em Operadoras de Telefonia", tese de doutorado, COPPE, 2005.
- Chen, F.C. e Khalil, H.K., “Adaptive Control of a Class of Nonlinear Discrete-Time Systems Using Neural Networks”, *IEEE Trans. on Automat. Control*, 40, pp. 791-801, 1995.
- Cybenko, G., “Approximation by a superpositions of a sigmoidal function”, *Math Control Signal Systems*, Vol.2, pp. 303-314, 1989.
- Doebelin, E. O., “Measurement Systems - Application and Design”, McGraw Hill, 1990.
- Freitas, A.A., “Linearização Exata Adaptativa Aplicada ao Controle Preditivo de Processo de Nível Multivariável”, dissertação de mestrado, UNB, 2007.
- Freeman, J.A., Skapura, D., “Neural Networks: Algorithms, Applications and programming techniques”, Addison-Wesley, 1991.

- Gambier, A., Unbehauen, H., "Adaptive predictive state-space control of a multivariable 3-tank system". Em 38th Conference on Decision & Control, páginas 1234–1239, Phoenix, Arizona, dezembro de 1999.
- Garcia, C.E., Morari, M. , "Internal model control 1: a unifying review and some new results", *Ind. Eng. Chem. Process Des. Dev.*, Vol.21, pp. 308-323, 1982.
- Gosmann, H. L., "Um Sistema Multivariável de Tanques Acoplados para avaliação de Técnicas de Controle". Dissertação de Mestrado, ENE/UNB, 2002.
- Grega, W. e Maciejczyk, "A. Digital control of a tank system". *IEEE Transactions on Education*, 37(3):271–276, 1994.
- Grega, W., "Hardware-in-the-loop simulation and its application in controleducation" *Frontiers in Education Conference*, 1999, 29th Annual. Volume 2, Issue Page(s):12B6/7 - 12B612 vol.2, 1999.
- Guimarães, B., Souza, A.S., Gosmann,H.L., Bauchspiess A., "Internet Based Remote Laboratory: The Level Control of Three Coupled Water Reservoirs". XV ACCA, Santiago de Chile, 2002.
- Gupta, M. M., Sinha, N. K., "Intelligent control systems: Theory and applications", IEEE Press, 1996.
- Haykin, S. "Neural Networks: A Comprehensive Foundation". Second Edition, Prentice Hall, 1999.
- Hunt, K.J.; Sbarbaro, D. Neural Networks for nonlinear internal model control, *IEE Proc.*, Part D, 138, pp. 431-438, 1991.
- Jesus, P. A., Hagan, O.;"Smoothing the control action for NARMA-L2 controllers" *Circuits and Systems*, The 2002 45th Midwest Symposium on. Volume 3, Page(s): III-37 - III-40 vol.3, Issue , 4-7, MWSCAS-2002.
- Johansson, K. H., Horch, A., Wijk, O., e Hansson, "A. Teaching multivariable control using the quadruple-tank process". Em 38th Conference on Decision & Control, páginas 807–812, Phoenix, Arizona, 1999.
- Kamil Š vancara, Petr Pivo ka, Brno, " The real time communication between Matlab and the real process controlled by PLC", 7 th International Research/Expert Conference, "Trends in the Development of Machinery and Associated Technology", TMT 2003, Lloret de Mar, Barcelona, Spain, 15-16 September, 2003.
- Ko, C. C., Chen, B. M., Chen, J., Zhuang, Y., e Tan, K. C. "Development of a web-based laboratory for control experiments on a coupled tank apparatus". *IEEE Transactions on Education*, 44(1):76–86, fevereiro de 2001.

- Lippmann, R. P., "An introduction to computing with neural nets", IEEE ASSP Mag, Vol 4, nº 2, pp 4-22, 1987.
- Luo, F., Unbehauen, R., "Applied Neural Networks for Signal Processing", Cambridge University Press, 1998.
- Lopes, J. S. B.; Popoff, L. H. G.; Vale, M. R. B. G.; Gabriel Filho, O.; Araújo, F. M. U. e Maitelli, A. L. "Um Esquema de Controle Neural Aplicado a um Sistema de Tanques Acoplados Utilizando Controlador Lógico Programável." XVI Congresso Brasileiro de Automática, pp. 2916-2919, Salvador-BA, outubro 2006.
- Lopes, J. S. B. ; Araújo, F. M. U. ; Maitelli, A. L. ; Gabriel F. "Controle avançado de processos industriais de petróleo utilizando um sistema de inferência neuro-fuzzy para geração dinâmica de set-points." In: VII International Conference on Industrial Applications, 2006, Recife. Anais em CDROM do VII Induscom, 2006.
- Ljung, L. (1987). "System Identification: Theory for the User", Prentice-Hall, Englewood Cliffs, NJ.
- Lucena, P. B.; "Análise de um Controlador Baseado no Jacobiano Estimado da Planta Através de uma Rede Neural", UFRN, 2005.
- Maia, C. A., "Controle Neural de Plantas Não-Lineares via Linearização", Dissertação de Mestrado, PPGEE/UFGM, 1996.
- Maia, C.A. e Resende, P.; "Projeto de controladores lineares para plantas não-lineares via linearização", XI Congresso Brasileiro de Automática, S. Paulo, 1996.
- Marvin M., Seymour P. "Perceptrons: An Introduction to Computational Geometry", Cambridge, 1998.
- Mathworks, Inc., "Neural Networks Toolbox for use with MATLAB", 2006.
- Melo, G.A.F e Bernardes, M. C., "Instrumentação e Controle de uma Maquete de Nível de Líquido com Quatro Tanques Interligados", trabalho de graduação, ENE/UNB, 2006.
- Metal Leve, "Controlador Programável", Ref. 1770-6.5.16. - Allen Bradley, 1987.
- Metal Leve, "Controlador Programável", Programação - Ref. 1785, 1988.
- Mohammad-Hoseini, M. Farrokhi, "Discrete sliding control of nonlinear nonaffine system using neural network and approximate models," 14th Iranian Conference on Electrical Engineering, Tehran, Iran, 2006.
- Moller, M.F., "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning." Neural Networks, 6 , 525-533, 1993.

- Mokri, S.S., Husain, H, Martono, W., Shafie, A. "Real Time Implementation of NARMA-L2 Control of a Single Link Manipulator"; American Journal of Applied Sciences, 5(12): 1642-1649, 2008.
- Narendra, K.S., S. Mukhopadhyay, "Adaptive Control Using Neural Networks and Approximate Models", IEEE Transactions on Neural Networks, 1997.
- Narendra, K.S. and Parthasarathy, K., "Identification and control of dynamical systems using neural networks", IEEE Trans. Neural Networks, vol. 1, no.1, 1990.
- Narendra K. and Boskovic, J. D., "Comparison of linear, nonlinear and neural network based adaptive controllers for a class of fed-batch fermentation processes", Automatica, Vol. 31, No. 6, pp. 817-840, 1995.
- Normey-Rico, J. E., "Projeto de Controladores PID industriais". Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, 2001.
- Ogata, K., "Engenharia de Controle Moderno". Editora Prentice Hall do Brasil Ltda., Rio de Janeiro, 2a. edição, 2003.
- Oliveira, J. C. P.; Livro técnico "Controlador Programável", Makron Books, 1993.
- Oliveira, J. C. P., Dissertação de Mestrado "Projeto e Implementação de Técnicas Neurais em Controlador Programável", UFMG, 1997.
- Oliveira, J. C.P.; Resende, P., Maia, C. A., "Controle Neural Gain Scheduling de Tanques Interativos", 3º SBAI, Vitória, pp. 224-228, 1997.
- Oliveira, J. C. P., Bauchspiess, A., "Neural Control of a Non Linear Process with Gain Scheduling, São Luís, SBRN 2004.
- Oliveira, J. C. P., Bauchspiess, A., "Controle Neural com Pesos Seleccionáveis pela Faixa de Operação Aplicado a um Processo de Nível de Líquidos de 4ª Ordem", XVI Congresso Brasileiro de Automática, Salvador, Bahia, 2006.
- Oliveira, J. C. P.; Bauchspiess, A., "Controle Neural de um Processo de Nível 4ª Ordem", VIII SBAI Simpósio Brasileiro de Automação Inteligente, Florianópolis, SC, 2007.
- Oliveira, J. C. P., Bauchspiess, A., "Neural Control Of a 4rd Order Level Process", COBEM 19º Congresso de Engenharia Mecânica, Brasília, DF, 2007.
- Oliveira, J. C. P., Bauchspiess, A., "Neural Control Of a 4rd Order Level Process", ABCM Symposium Series in Mechatronics - Vol. 3 - pp.197-206, Copyright 2008 by ABCM.
- Pearlmutter,. "Fast exact multiplication by the Hessian". Neural Computation, 6(1), pp.147-160, 1994.

- Pitarque, Alfonso ; Roy, Juan Francisco; Ruiz, Juan Carlos. “Las redes neuronales como herramientas estadísticas no paramétricas de clasificación.”, Universidad de Valencia, España. *Psicothema*, 2000 - vol12, Supl. nº 2, pp 459-463.
- Prapong Klysubun, P., Netsai, C., Ratchasima, N., "Development of Matlab-Based Data Logging System at Siam Photon Source", Proceedings of EPAC (European Particle Accelerator Conference), Edinburgh, Scotland, 2006.
- Pukrittayakamee A., Jesus O., Hagan M., “Smoothing the Control Action for NARMA-L2 Controllers”, 45th IEEE International Midwest, Oklchoma. 2002.
- R. Sekhon, H. Bassily, J. Wagner, P.E., and J. Gaddis, “Stationary Gas Turbines – A Real Time Dynamic Model with Experimental Validation”. American Control Conference, Minneapolis, Minnesota, USA, 2006.
- Ramos, A. P. e Wense, G. L. B. “Laboratório Didático de Controle de Processo de Nível”, Trabalho de Graduação, Engenharia Mecatrônica, Universidade de Brasília, 2008.
- Reed, R. D., Marks, R. J.; “Neural Smothing Supervised Learning in Feedforward Artificial Neural Networks”, The Massachusetts Institute of Technology Press, London, England, 1999.
- Rumelhart, D .E., Hinton, G. E. and Williams, R.J., “Parallel Distributed Processing.” Cambridge, MA., 1986.
- Sekhon, R.. Bassily, H. Wagner, J. Gaddis, J., “Stationary gas turbines - a real time dynamic model with experimental validation”, Dept. of Mech. Eng., Clemson Univ., SC, USA; American Control Conference, 2006.
- Shamma, J. S., Athans, M., “Analisy of gain scheduled control for nonlinear plants”, IEEE Trans. Automat. Contr., vol. 35, pp. 898-907, 1990.
- Slotine, J.J.E.; Weiping, L., “Applied Nonlinear Control.”, Prentice Hall, New Jersey, 1991
- Simas, H., Bruciapaglia, A. H., e Coelho, A. A. R., “Advanced control using three low-cost laboratory models: Experiments and design issues”. Em XII CBA – Brazilian Automatic Control Conference, páginas 431–436, Uberlândia, MG, setembro de 1998.
- Souza, E., “Método de Interpolação Polinomial para o Problema de Linearização Exata”, Tese de doutorado, Universidade Federal de Uberlândia, Uberlândia, MG, 1998.
- Sharda, R., Wang, J., "Neural networks and operations research/management science", *European Journal of Operational Research*, Vol. 93 pp.227-9, 1996.
- Soares, P.P.S.; Nadal, J., “Aplicação de uma Rede Neural Feedforward com Algoritmo de Levenberg-Marquardt para Classificação de Alterações no Segmento ST do

- Eletrocardiograma”; IV Congresso Brasileiro de Redes Neurais; São José dos Campos; BRASIL; Publicado em CD-ROM, 1999.
- Tamayo, S. e Mendes, R.G.R., “Sistema de Nível de Líquidos de 2ª Ordem Didático” Trabalho de Graduação, Engenharia Mecatrônica, Universidade de Brasília, 2004.
- Wang, H.; Brown, H., Harris, C.J., “Neural network modelling of unknown nonlinear systems subject to immeasurable disturbances”, IEE Proc. Control Theory Appl., vol.141, no. 4, pp. 216-222, 1994.
- Zurada, J. M., “Introduction to Artificial Neural Systems” Publisher: West Pub. Co, US, 1992.

APÊNDICE

Apresentação de um trecho de treinamento da rede neural discutida no item 5.4 e os parâmetros finais (w_1 , w_2 , w_3 , b_1 , b_2 e b_3) obtidos que foram utilizados para o controlador neural de três camadas, utilizado para a simulação e o controle da planta.

[w1,b1,w2,b2,w3,b3]=initff(PZt,7,'tansig',14,'tansig',Tt,'logsig')

TRAINBPX: 1400/100000 epochs, lr = 0.170237, SSE = 0.020076.

TRAINBPX: 1402/100000 epochs, lr = 0.187686, SSE = 0.0199748.

w ₁ = [
	-9.8240	10.5384	-1.3422	4.7562	-2.7104		
	0.9777	-6.8041	2.4498	-9.8403	8.4739		
	-12.9111	4.7459	6.2318	2.8125	-2.4431		
	-6.7226	8.1266	9.3906	0.4391	-5.6741		
	0.8590	-9.7947	9.0968	2.5673	6.4767		
	-1.8057	8.0082	-10.1647	-7.4322	-1.5321		
	-6.9936	-5.7113	-8.5098	-3.4208	-7.9019];		
w ₂ = [
	0.4268	-0.7989	0.3422	0.0673	0.0011	0.3624	0.1225
	-0.0487	-0.3967	-0.2055	-0.4173	0.2973	-0.5586	-0.5616
	-0.4756	-0.2397	0.4168	0.1376	0.2187	0.5853	0.2533
	0.5625	0.3078	0.1009	-0.1206	-0.4784	0.8177	0.1967
	-0.2129	0.4046	-0.1789	-0.4442	-0.2956	0.6845	0.1374
	-0.1281	0.8187	-0.4201	-0.3409	-0.4147	0.7595	-0.0219
	0.4208	0.2106	0.5673	-0.3195	-0.2859	-0.4795	0.3711
	-0.2545	-0.2596	0.2367	-0.7241	-0.4922	-0.0841	-0.1341
	-0.2726	0.2219	0.0218	-0.2614	-0.5257	0.1158	-0.6006
	0.0604	-1.0741	-0.1732	0.3176	-0.6139	-0.7370	-0.0990
	-0.3798	0.3382	0.1088	0.2496	-0.1505	0.5102	0.7117
	-0.1402	-0.3767	0.4724	0.3017	-0.4436	0.0465	-0.3154
	0.5204	0.1232	-0.5450	0.2890	-0.0626	0.5396	-0.1269
	0.3648	-0.0822	-0.5121	-0.6315	-0.4634	-0.0981	-0.3801];
w ₃ = [0.8070 0.3141 -0.2862 -1.3009 -0.1304 -1.0927 -0.7032-1.3620 0.6378							
1.7976 -0.9649 -0.0286 0.1484 -0.9564];							
b ₁ = [b ₂ = [b ₃ = [1.7400].				
0.1570	0.3773						
0.0355	0.0161						
-0.9516	0.6635						
-0.8563	-0.4611						
-0.6716	0.6214						
-0.3812	-0.8104						
1.0074];	0.5163						
	1.2367						
	-0.5745						
	-0.3891						
	0.2198						
	0.5692						
	-0.4503						
	0.8022];						