

Universidade Federal de Santa Catarina  
Programa de Pós-graduação em Engenharia Química

**REINALDO FELIPONI MARCOLLA**

**Investigação e Aplicação de Técnicas de Modelagem e Controle de  
Processos Baseadas em Redes Neurais com Ajuste em Linha Utilizando  
Algoritmos Genéticos.**

**Dissertação de Mestrado apresentada ao  
Programa de Pós-graduação em Engenharia  
Química da Universidade Federal de Santa  
Catarina como requisito parcial para a  
obtenção do título de Mestre em Engenharia  
Química.**

**Orientador: Prof. Ricardo Antonio Francisco Machado, Dr. Sc.**

**Florianópolis, 2005**

**REINALDO FELIPONI MARCOLLA**

**Investigação e Aplicação de Técnicas de Modelagem e Controle de Processos Baseadas em Redes Neurais com Ajuste em Linha Utilizando Algoritmos Genéticos.**

Esta Dissertação foi julgada e aprovada como requisito parcial para a obtenção do título de **Mestre em Engenharia Química** no **Programa de Pós-graduação em Engenharia Química** da Universidade Federal de Santa Catarina, área de concentração de Processos Químicos e Biotecnológicos.  
Florianópolis, fevereiro de 2005.

---

Prof. Agenor F., Dr. Sc..  
Coordenador do Curso de Pós-Graduação em  
Engenharia Química

---

Prof. Ricardo Antonio F. Machado, Dr. Sc.  
**Orientador**

**Banca Examinadora**

---

Prof. Ariovaldo Bolzan, Dr.Sc.

---

Prof. Adriano Cancelier, Dr.Eng.

*À minha família,  
Adayr Marcolla,  
Olívia Feliponi Marcolla,  
Otávio Feliponi Marcolla  
por todos os sacrifícios que  
fizeram visando minha educação.*

*Agradecimentos*

*À Universidade Federal de Santa Catarina-UFSC,  
Ao Laboratório de Controle de Processos, LCP, do  
Departamento de Engenharia Química e  
Engenharia de Alimentos,  
pelo suporte tecnológico;  
Ao Programa de Pós-graduação em Engenharia Química;  
Ao Programa PRH-34 da Agência Nacional do Petróleo pelo apoio  
financeiro;*

*Ao Professor Ricardo Antônio Francisco Machado  
pela amizade, orientação,  
incentivo e grande oportunidade.*

*Ao amigo Carlos Alberto Claumann pelo grande apoio, incentivo  
e auxílio junto as metodologias aplicadas no trabalho.  
Ao amigo Adriano Cancelier pelo incentivo e auxílio  
na aplicação do controlador preditivo.  
Ao amigo Luis Henrique Castelan Carlson pelo  
auxílio fornecendo o software de aquisição de dados.  
Ao amigo Rafael Arioli pelo apoio na condução  
das reações de polimerização.*

*A todos os amigos do LCP que, de alguma forma, contribuíram  
para a conclusão deste trabalho.*

*A todos meus amigos que sempre me apoiaram neste caminho.*

## RESUMO

MARCOLLA, REINALDO FELIPONI. **Investigação e Aplicação de Técnicas de Modelagem e controle de Processos Baseadas em Redes Neurais com Ajuste em Linha Utilizando Algoritmos Genéticos** 2005. 200 f. Dissertação de Mestrado (Mestrado em Engenharia Química) – Programa de Pós-Graduação em Engenharia Química, UFSC, Florianópolis.

---

Neste trabalho foi realizado um estudo em torno de um sistema de controle de temperatura, o qual apresenta não-linearidades e perturbações difíceis de serem modeladas. Inicialmente foi investigada a aplicação de um controlador linear PID, ajustado segundo metodologias clássicas da pré-sintonia segundo práticas simplificadas de identificação do sistema em questão. Os resultados observados para o controlador PID não foram considerados satisfatórios devido a pouca informação a respeito da dinâmica do processo que fica agregada ao modelo utilizado, bem como o tempo morto envolvido nas ações de controle. O fato do controlador PID não apresentar bom desempenho para sistemas com dinâmica variável, como caso do sistema estudado foi demonstrado.

Para tratar a problemática do tempo morto envolvido nas ações de aquecimento e resfriamento do sistema, foram investigadas metodologias para de identificação do sistema e determinação do tempo morto do processo, através de métodos gráficos e através do método de Mínimos Quadrados.

A fim de solucionar o problema de modelagem do sistema, foi aplicado uma modelagem empírica com base em uma rede Neural Feedforward (RNA), com 3 camadas, treinada “*off-line*” a partir de um conjunto de padrões obtido do próprio sistema, com o reator contendo água; levando em consideração os tempos mortos pré-determinados, utilizando a metodologia de um algoritmo híbrido que utiliza algoritmos genéticos e mínimos quadrados. Esta RNA foi aplicada a um controlador Preditivo Baseado em Modelo (MPC) como modelo do preditor, e foram obtidos resultados satisfatórios para o sistema de controle de temperatura do reator contendo água no seu interior.

Também foi demonstrado o desempenho do controlador preditivo (MPC) com a RNA estática, frente o controle de temperatura conduzindo uma reação de polimerização em suspensão de estireno, onde o controlador apresentou desempenho bom mas devido a alterações na dinâmica do sistema quando conduzindo reação química, pois o sistema já não contém somente água e a reação pode liberar ou retirar calor do meio; apresentou-se a necessidade de aplicar uma ferramenta que possa tratar estes desvios na dinâmica do processo a fim de melhorar ainda mais o desempenho do controlador MPC.

Contudo, foi aplicada uma metodologia da adaptação, ou ajuste dos pesos e bias da camada de saída da RNA utilizada como modelo do MPC, através de uma metodologia de minimização da função objetivo do controlador MPC, via Algoritmos Genéticos com a rede sendo alimentada de forma recorrente (o valor predito é realimentado na própria rede para fins de otimização dos pesos), onde o desempenho do controlador foi demonstrado tanto para o controle de temperatura do reator contendo somente água, quanto conduzindo reação de polimerização de estireno, sendo que o desempenho foi superior com o MPC utilizando método adaptativo em comparação ao MPC utilizando a rede estática em todos os casos.

O desempenho do controlador juntamente com o método adaptativo conduzindo reação química de polimerização frente a intenso ruído no sinal da variável manipulada, representando uma perturbação grande ao sistema; onde o controlador MPC comportou-se satisfatoriamente mesmo sobre esta condição extrema de trabalho.

Por fim, pôde-se concluir que o controlador MPC utilizando modelo empírico baseado em uma RNA obteve desempenho satisfatório para a RNA estática e este desempenho pôde ser ainda melhorado com a utilização do método de adaptação on-line baseado em Algoritmos Genéticos, melhorando a transição de *set-point* do controlador, a manutenção da referência (*set-point*) e também auxiliando a tratar perturbações oriundas de ruído no sinal do sistema de aquisição de dados.

---

**Palavras Chave:** Controle Preditivo, Redes Neurais, Algoritmos Genéticos, Poliestireno, Inteligência Artificial.

## ABSTRACT

MARCOLLA, REINALDO FELIPONI. **Investigation and Application of Techniques of Modeling and control of Processes Based on Neural Network with Adjustment in Line Using Genetic Algorithms 2005.** 200 pg.. Dissertation of Master's degree (Master's degree in Chemical Engineering)–Program of Masters degree in Chemical Engineering, UFSC, Florianópolis.

---

In this work a study was accomplished around a system of temperature control, which presents non-linearity's and disturbances difficult of they be modeled. Initially the application of a lineal controller was investigated PID, adjusted according to classic methodologies of the pré-sintony according to simplified practices of identification of the system in subject. The results observed for controller PID they were not considered satisfactory due to little information regarding the dynamics of the process that is joined to the used model, as well as the dead time involved in the control actions. Controller's PID fact not to present good acting for systems with variable dynamics, as case of the studied system was demonstrated..

To treat the problem of the dead time involved in the heating actions and cooling of the system, methodologies were investigated for of identification of the system and determination of the time killed of the process, through graphic methods and through the method of Square Minima.

In order to solve the problem of modeling of the system, an empiric modeling was applied with base in a net Neural Feedfoward (RNA), with 3 layers, “off-line” starting from a group of patterns obtained of the own system, with the reactor containing water; taking in consideration the pré-certain times, using the methodology of a hybrid algorithm that uses genetic and minimum algorithms square. This RNA was applied her/it a controller Preditivo based on Model (MPC) as model of the predictor, and they were obtained satisfactory results for the system of control of temperature of the reactor containing water in your interior.

The acting of the controller predictivo was also demonstrated (MPC) with static RNA, front the temperature control leading a polymerization reaction in styrene suspension, where the controller presented good acting but due to alterations in the

dynamics of the system when leading chemical reaction, because the system no longer it contains only water and the reaction can liberate or to remove heat of the middle; he/she came the need to apply a tool that can treat these deviations in the dynamics of the process in order to get better still more controller's MPC acting.

However, a methodology of the adaptation was applied, or adjust of the weights and bias of the layer of exit of RNA used as model of MPC, through a methodology of minimization of controller's MPC function objective, through Genetic Algorithms with the net being fed in an appealing way (the predicted value is realimented in the own net for ends of optimization of the weights), where the controller's acting was demonstrated so much for the control of temperature of the reactor containing only water, as leading reaction of styrene polymerization, and the acting was superior to the MPC using method adaptative front MPC with net static front to in all the cases.

The controller's acting together with the method adaptative leading chemical reaction of polymerization front to intense noise in the sign of the manipulated variable, representing a big disturbance to the system; where controller MPC behaved satisfactorily same about this extreme condition of work.

Finally, it could be ended that the controller MPC used empiric model based on a RNA obtained satisfactory acting for static RNA and this acting could be still gotten better with the use of the method of adaptation on-line based on Genetic Algorithms, improving the transition of set-point of the controller, the maintenance of the reference (set-point) and also aiding to treat disturbances originating from of noise in the sign of the system of acquisition of data.

---

**Keywords: Predictive Control, Neural Network, Genetic Algorithms, Polystyrene, Artificial Intelligence.**



# ÍNDICE

1.Introdução.....	14
2.Revisão Bibliográfica.....	23
2.1.Considerações Gerais .....	23
2.2.Identificação da Planta .....	26
2.2.1.Métodos Determinísticos .....	27
Sistemas que podem ser considerados de primeira ordem .....	35
Sistemas de segunda ordem pouco amortecidos .....	37
2.3.Controlador PID .....	50
2.3.1.Conceitos Gerais.....	50
Ação proporcional .....	50
Ação Integral.....	51
Ação derivativa .....	51
2.3.2. Modificações no algoritmo do controlador PID.....	51
2.3.3.Implementação Digital.....	51
O Controlador PID Digital.....	52
2.4.Metodologias para obtenção dos parâmetros do Controlador PID (pré-Sintonia do controlador) .....	55
2.4.1. Método de Cohen-Coon.....	56
2.4.2. Métodos Baseados no critério da Integral do Erro .....	57
2.4.3.Método de ajuste de Ziegler e Nichols .....	59
2.5.Uma Introdução à teoria de Algoritmos Genéticos.....	60
2.5.1.Introdução .....	60
2.5.2. Terminologia empregada em Algoritmos Genéticos.....	61
2.5.3.Representação Genética .....	62
2.5.4. Estrutura dos Algoritmos Genéticos.....	63
2.5.5. Geração da População Inicial .....	64
2.5.6. Diversidade Genética da População .....	65
2.5.7. Seleção .....	66
2.5.8.Operadores Genéticos.....	68
2.5.9.Otimização de Problemas Multivariáveis .....	76
2.5.10.Resolução de Problemas de Minimização .....	77
2.5.11.Codificação Real .....	78
2.6.Redes Neurais Artificiais.....	80
2.6.1.Inspiração da Metodologia.....	80
2.6.2.Neurocomputação.....	81
2.6.3.Motivação do uso da Redes Neurais Artificiais .....	82
2.6.4.Entendendo o funcionamento das redes neurais Artificiais.....	85
2.6.5.Conjunto de dados e Treinamento de uma Rede Neural Artificial .....	95
3.Aplicações e ferramentas utilizadas.....	101
3.1.Algoritmo de CLAUMANN(1999) / CANCELIER(2004) para treinamento off-line da RNA - “Aplicação de um Algoritmo Genético em Codificação Real para o Treinamento da Rede Neural Feedforward”.....	101
3.2.Adaptação on-line dos pesos de uma Rede Neural Artificial .....	108
3.3.O controlador utilizado.....	111
4.Material e Métodos .....	112
4.1.Reator de Aço .....	112

4.2.Reação Química de Polimerização.....	116
5.Resultados e Discussões.....	117
5.1. Identificação do Sistema.....	117
5.1.1. Identificação através dos Mínimos Quadrados .....	117
5.1.2. Identificação através do método gráfico para sistemas de primeira ordem com tempo morto. ....	123
5.1.3. Parâmetros preliminares para o controlador PID .....	125
5.2.Aplicação de um controlador Preditivo Baseado em um modelo não-linear de uma Rede Neural Feedforward treinada <i>off-line</i> .....	130
5.2.1. Justificativas: Tempo Morto e não-linearidades .....	131
5.2.2. Treinamento off-line da rede neural Feedforward.....	133
5.2.3. Avaliação do desempenho do Controlador Preditivo Baseado em Modelo aplicado ao sistema de controle de Temperatura .....	138
6.Conclusões.....	153
7. Sugestões .....	156
8. Bibliografia.....	157
9.Anexos & Apêndices.....	162

# ÍNDICE DE FIGURAS

Figura 1 - Resposta degrau de sistemas de primeira ordem com diferentes constantes de tempo .....	29
Figura 2 - Comportamento de sistemas de segunda ordem a resposta degrau para $H(s) = \frac{1}{s^2 + 2\xi s + 1}$ onde $\xi = 0.1, 0.4, 0.7$ e $1.3$ respectivamente. ....	31
Figura 3 - Resposta de um sistema de segunda ordem para diferentes fatores de amortecimento – quanto menor o fator maior o tempo para regime estacionário. .	32
Figura 4 - Sistema de primeira ordem versus sistema de segunda ordem sobre amortecido .....	33
Figura 5 - Ilustração das especificações de regime transitório.....	33
Figura 6 – Respostas principais que podem ser obtidas.....	35
Figura 7 - Ilustração do desenvolvimento do método para obtenção dos parâmetros....	37
Figura 8- Aplicação do método descrito para o sistema de segunda ordem .....	39
Figura 9 – Observação aproximada do regime transitório do sistema de segunda ordem analisado .....	40
Figura 10 - Comparação do método gráfico para sistemas de segunda ordem sub amortecidos.....	40
Figura 11 – Resultado de ajuste linear – comparação do método de MQ.....	44
Figura 12 – Comparação entre um sistema em malha aberta e em malha fechada para variação degrau sobre o sistema .....	54
Figura 13 – Método da Curva de reação de Ziegler e Nichols .....	59
Figura 14 – Diagrama de fluxo para um AG.....	64
Figura 15 - Perda de diversidade em alguns genes dos indivíduos da população CLAUANN (1999).....	65
Figura 16 - Ilustração do funcionamento do <i>rolleta</i> em um população de 4 indivíduos	67
Figura 17 - Ilustração do funcionamento do <i>crossover 1-ponto</i> .....	68
Figura 18 - Mutação ocorrida em um gene da <i>string</i> .....	69
Figura 19 - Ilustração do funcionamento do escalonamento <i>linear</i> .....	72
Figura 20 – Topologia de um neurônio artificial de Warren MacCulloch.....	86
Figura 21 – Modelo de um neurônio artificial.....	86
Figura 22 – $v_k$ versus $u_k$ , em relação ao sinal das bias .....	88
Figura 23– Modelo não-linear de um neurônio .....	89
Figura 24 - <i>Rede alimentada adiante ou acíclica (feedforward)</i> .....	91
Figura 25 - <i>Representação de uma Rede Neural Artificial com múltiplas camadas</i> .....	92
Figura 26 – Rede Neural utilizada no trabalho .....	94
Figura 27 - Aprendizado Supervisionado.....	97
Figura 28 - Aprendizado não – supervisionado.....	98
Figura 29 - Fluxograma do algoritmo genético em codificação real. (Claumann, 1999) .....	103
Figura 30 - Codificação da FANN na forma de um cromossomo. CANCELIER (2004) .....	106
Figura 31 - <i>Codificação dos pesos da camada intermediária da FANN na forma de um cromossomo – CANCELIER(2004)</i> .....	107
Figura 32 - <i>Fluxograma do algoritmo para o sistema híbrido utilizado no treinamento da FANN – CANCELIER(2004)</i> .....	108
Figura 33 - FANN com a representação utilizada no treinamento recorrente CANCELIER(2004).....	109

Figura 34 - Codificação dos pesos da camada de sída da RNA em um cromossomo..	110
Figura 35 - Esquema da unidade piloto de polimerização (Mazzucco, 2003) “com pequena alteração a respeito do trabalho de Mazzucco quanto a válvula reguladora de pressão de vapor” .....	113
Figura 36 - Detalhes do reator de polimerização (Mazzucco, 2003).....	114
Figura 37 - Tela principal do software de aquisição de dados e controle.....	116
Figura 38- Respostas obtidas para identificação do processo (Mínimos Quadrados) .	118
Figura 39 – Aberturas das válvulas para geração dos padrões de identificação .....	118
Figura 40 – Comparação entre o modelo obtido e os dados reais .....	119
Figura 41 – Comparação entre o modelo obtido e os dados reais .....	121
Figura 42 – Curva de resposta para o sistema .....	123
Figura 43 – Curva de resposta obtida .....	124
Figura 44 – Tela do programa para determinação dos parâmetros preliminares PID ..	125
Figura 45 – Resposta do Controlador PID a variações de referência a partir de ajustes clássicos de parâmetros .....	127
Figura 46– Ações de controle do controlador PID .....	127
Figura 47 – Experimento aplicando o controlador PID ajustado por métodos clássicos a uma reação de polimerização do estireno.....	128
Figura 48- Ações de controle do controlador PID para reação química ( <i>com uma breve intervenção manual na transição</i> ) .....	128
Figura 49 – Ações das válvulas tomadas na identificação do sistema.....	134
Figura 50– Sobreposição dos pontos experimentais e os pontos gerados pela rede treinada .....	136
Figura 51 - Diagrama de blocos simplificado do sistema de controle preditivo .....	139
Figura 52 - Desempenho deste controlador 1 para o caso sem adaptação com $\alpha = 0.96$ . .....	141
Figura 53 -Ações de controle tomadas pelo controlador 1 para o caso sem adaptação com $\alpha=0.96$ .....	141
Figura 54 – Resposta do controlador 1 com adaptação por GA.....	142
Figura 55 – Ações de controle tomadas pelo controlador 1 com adaptação por GA ...	142
Figura 56 - Curva de resposta do controlador MPC para controlador 2 , sem adaptação (314 amostras cerca de 52 minutos), $\alpha = 0,75$ .....	144
Figura 57 – Ações de controle tomadas pelo controlador 2 para o caso sem adaptação .....	144
Figura 58 – Curva de resposta do controlador 2 para o caso com adaptação.....	145
Figura 59 - Ações de controle tomadas pelo controlador 2 para o caso com adaptação .....	145
Figura 60 – Curva de resposta para o controlador 2 em experimento com reação de polimerização sem adaptação, somente utilizando a RNA recorrente como modelo do MPC.....	147
Figura 61– Ações de controle para o controlador 2 sem adaptação com reação química .....	147
Figura 62 – Curva de resposta do controlador 2 para experimento com reação química e com adaptação por GA. ....	148
Figura 63 – Ações de controle para experimento com reação química e com adaptação por GA para o controlador 2 .....	148
Figura 64– Curva de resposta para o caso com ruído .....	151
Figura 65- Ações de controle para o caso com ruído.....	151
Figura 66 - Variação do valor da Função objetivo para experimento com adaptação por GA.....	152

# ÍNDICE DE TABELAS

Tabela 1- Relações de projeto de Cohen-Coon .....	56
Tabela 2- Valores para aplicação do Método ITAE (b – Para mudanças de set-point, a relação de projeto para o modo integrativo é $\tau/T_i = A + B(\theta/\tau)$ ). .....	58
Tabela 3 - Esquema para obtenção dos parâmetros do controlador .....	60
Tabela 4 - Comparação entre cérebro e computador .....	82
Tabela 5– Comparativo entre computadores e Neurocomputadores .....	82
Tabela 6 - Parâmetros ajustados para controlador PID – Caso 1 .....	126
Tabela 7 - Parâmetros ajustados para controlador PID – Caso 2 .....	126
Tabela 8 - Formação dos padrões para treinamento da RNA.....	134
Tabela 9 - Pesos e bias obtidos no treinamento da Rede Neural ( <i>o arquivo esta no mesmo formato que é utilizado no software de controle</i> ).....	137

# LISTA DE SÍMBOLOS

$\xi$ -	Vetor de erros cometidos para o método de mínimos quadrados. Fator de amortecimento em funções transferência de segunda ordem.
$w_n, \omega_n$ -	Frequência natural.
$\tau_d$ -	Atraso puro no tempo.
$\tau_1, \tau_2$ -	Constantes de tempo de um sistema.
$L^{-1}$ -	Notação de transformada inversa de Laplace.
$K$ -	Ganho do sistema.
$\tau$ -	Constante de tempo do sistema.
$\theta$ -	Tempo morto o sistema. Matriz de regressores para o método de Mínimos Quadrados.
$U_1, V_2$ -	Válvula de aquecimento.
$U_2, V_1$ -	Válvula de resfriamento.
$t$ -	Tempo.
$H(s)$ -	Função transferência no domínio de Laplace.
$m_1$ -	Área sob a curva de resposta.
$X^T$ -	Matriz transposta “X”.
$[X^T X]^{-1}$ -	Matriz pseudo-inversa de “X”.
$W$ -	Matriz diagonal de pesos do estimador de Mínimos Quadrados.
$\theta_{MQ}$ -	Estimador de Mínimos Quadrados.
$\theta_{MQP}$ -	Estimador de Mínimos Quadrados ponderados.
$\xi^T$ -	Matriz transposta “ $\xi$ ”.
$J_{MQ}$ -	Função custo do estimador de Mínimos Quadrados.
$\Delta t$ -	Intervalo de tempo (amostragem).
$T_i$ -	Constante integrativa do controlador PID.
$K_c$ -	Ganho do controlador PID.
$T_d$ -	Constante derivativa do controlador PID.
$D$ -	Densidade genética da população.
$f$ -	Fitness.
$F$ -	Fitness escalonado.
$\sigma$ -	Variância.

# 1.Introdução

Processos na indústria de petróleo e gás natural são conhecidos pelos elevados custos de operação de suas plantas os quais estão associados aos alto valor agregado de seus produtos e sub-produtos. A produção em grande escala e muitas vezes em fluxo contínuo traz a necessidade de se **organizar** diferentes parâmetros dentro do processo a fim de se manter a produção dentro de margens satisfatórias de qualidade, sendo que estas margens, frequentemente se apresentam muito estreitas de lucratividade ou condições operacionais.

Organizar estes parâmetros adequadamente é uma tarefa essencial uma vez que a partir deles se pode estruturar **modelos** para o comportamento dos processos dentro da cadeia produtiva e com isto se pode estimar como uma determinada operação poderá influenciar na especificação do produto final.

Esta influência de uma determinada operação ou um conjunto destas muitas vezes se apresenta como um fator o qual traz a necessidade de monitoramento permanente do processo e eventual ação sobre o mesmo, a fim de que o produto final não seja produzido fora de especificação ou em alguns casos até mesmo não ocorra um acidente na planta.

CARDOSO (2004), descreve algumas questões envolvidas na logística do Petróleo, onde cita-se a utilização de sistemas de controle e aquisição de dados em oleodutos e gasodutos, tal como o SCADA (*Supervisory Control and Data Aquisition*); o qual permite o acompanhamento e a supervisão das operações de transferência em tempo real, favorecendo sistemas operantes em estoque de *Just-In-Time*; também a utilização do GPS (*Global Positioning System*) e CAD (*Computer Aided Design*) são citados. Tais tecnologias segundo CARDOSO (2004) têm na sua utilização “*o aumento da performance de atendimento à demanda, no caso da polidutos (dutos utilizados para mais de um produto)*”. Porém, tais conceitos e estratégias se apresentam como ferramentas essenciais em toda a cadeia produtiva, da exploração até o ponto de consumo.

Torna-se importante enfatizar que não é uma tarefa simples tal monitoramento e tão pouco econômica, mas extremamente necessária a fim de que se possa obter resultados satisfatórios de produção na indústria de petróleo e gás natural.

Este monitoramento somente pode ocorrer com o conhecimento prévio da operação a ser realizada, as propriedades que à influenciam, como temperatura,

densidade, pressão, composição; das condições de processo como tempo de residência, taxa de reação, conversão; dentre outros fatores que podem ser inerentes ao processo em questão. A partir deste conhecimento prévio se pode tomar algumas decisões como por exemplo para o caso geral de se obter gasolina e solventes a partir do petróleo (óleo cru) a faixa de temperatura necessária é, geralmente, 40 a 175° C. Em uma bateria de destilação de óleo cru, não somente são separados os hidrocarbonetos para produção de gasolina (C<sub>5</sub> – C<sub>10</sub>) mas também querosene (C<sub>11</sub> a C<sub>12</sub>) na faixa de temperatura de 175 a 235° C ; gasóleo leve (C<sub>13</sub> a C<sub>17</sub>) na faixa de temperatura de 235 a 305° C e pesado (C<sub>18</sub> a C<sub>25</sub>) na faixa de temperatura de 305 a 400° C, gás liquefeito de petróleo (GLP, C<sub>3</sub> a C<sub>4</sub>) na faixa de até 40° C, bem como os lubrificantes (C<sub>26</sub> a C<sub>38</sub>) na faixa de temperatura de 400 a 510° C. Estas faixas de temperatura são gerais e dependem exclusivamente das características dos hidrocarbonetos os quais se quer extrair. O que pode variar nisto tudo é a composição do óleo cru, pois esta composição que representa o percentual de cada um dos hidrocarbonetos para fracionamento varia de acordo com o reservatório do qual foi extraído o óleo.

Inicialmente pode não parecer importante esta informação, mas tanto é que no Brasil, as refinarias de petróleo estão dimensionadas para trabalhar com o conhecido óleo leve. Quando da extração em reservatórios brasileiros de óleo pesado o mesmo é negociado com países do oriente médio pois, o mesmo não se apresenta economicamente viável para ser processado em nossas refinarias.

Esta variável “composição dos óleos” torna-se importante para ser monitorada dentro de um processo, por exemplo, de fracionamento de óleo cru, já que esta operação esta baseada em relações de equilíbrio de fases, temperatura e pressão, onde as condições de operação podem mudar substancialmente. Ocorre que no caso da composição, torna-se muito difícil este monitoramento, podendo ser realizado com o emprego de cromatógrafos on-line nos fluxos de entrada. Mas no caso de alguma mudança significativa de composição, o tempo de análise necessário poderia representar um problema pois então, influenciando em perda de energia, por exemplo, dos fornos utilizados para vaporização das frações de óleo cru. A razão de refluxo e ponto de alimentação da carga também são parâmetros influenciados pela mudança na composição do óleo, já que com a mudança nas relações de equilíbrio, as composições nos estágios somente poderão ser conhecidas quando as temperaturas assim o forem.

Neste ponto também notamos que a variável temperatura é determinante em processos da indústria de petróleo e pode influenciar diretamente na especificação dos

produtos derivados do petróleo. Em uma operação como a destilação, a manutenção da temperatura da coluna e de seus estágios é determinante para um bom rendimento da operação, e o monitoramento desta variável é significativamente importante para obtenção do produto esperado.

Partindo-se da premissa de que a temperatura é uma variável significativa em uma operação dentro da cadeia produtiva torna-se necessário monitorá-la e muitas vezes, tomar uma ação dentro do processo para que esta se mantenha dentro de uma faixa pré-estabelecida.

Para tanto é necessário se obter um modelo matemático que represente a variação da especificação que se quer manter em função da propriedade que se apresenta relevante na operação, a temperatura por exemplo. Este modelo geralmente não é obtido de maneira simples, muitas vezes sendo representado por equações diferenciais, ordinárias ou parciais, o que vai depender da especificação a ser monitorada, que pode não depender somente de uma única propriedade física por exemplo. Supondo-se que o modelo matemático obtido pode representar bem o sistema, agora possuímos uma ferramenta muito eficiente para que possamos monitorar a operação e verificar se tudo ocorre “normalmente”, ou seja, conforme o esperado para as condições do produto a ser processado e as especificações do sub-produto desejado. Mas, no caso de ocorrer alguma anomalia ainda não se pode “intervir” no processo a fim de corrigir o problema, até então somente se pode verificar que algo está errado e se calcular os prejuízos, que vão depender do problema que se apresenta. Para o caso da temperatura em uma coluna de destilação por exemplo, uma redução desta pode comprometer toda a operação, causando grandes prejuízos.

O leitor pode se perguntar como então se pode intervir satisfatoriamente em uma operação unitária de engenharia a fim de se manter as condições de operação dentro do necessário para satisfazer as especificações do produto e manter o processo monitorado sempre produtivo? A resposta pode ser resumida em uma única palavra: **Controle**.

Ao conjunto destes fatores:

- modelagem,
- monitoramento,
- intervenção ou ação sobre um processo ou operação industrial;

poderia ser resumido como **Controle de processos**.



O controle de processos apresenta-se como uma ferramenta muito útil para solucionar problemas como os citados anteriormente, e possui vasta aplicação na indústria de petróleo e gás natural.

Esta ferramenta se ocupa de monitorar certas variáveis de um sistema a fim de mantê-las em níveis desejados conforme a especificação de produto esperada. Neste caso, a partir de um modelo matemático de um processo se pode realizar a predição de saída do processo real e corrigi-la caso ocorra algum desvio do valor desejável. Caso se perceba que a saída do processo real se encontra fora da faixa de operação pode-se então se tomar uma **ação** sobre o sistema a fim de se manter estas dentro da faixa esperada.

As diferentes formas de se manipular ou agir sobre o processo a fim de se corrigir os desvios existentes são as conhecidas como **estratégias de controle**, e podem ser aplicadas diferentemente dependendo do processo a ser controlado; devem ser entendidas rústicamente como formas de se descrever e representar o processo bem como metodologias de se estruturar as equações e métodos matemáticos de que se alça mão com o objetivo de a partir de variáveis manipuladas se manter variáveis controladas dentro de faixas de operação esperadas no processo.

Os modelos do processo podem ser obtidos de muitas formas e fazem parte de uma etapa conhecida como **identificação do processo ou planta**. Esta identificação visa obter uma representação para o modelo do processo em questão com variáveis manipuladas (de entrada) em função de controladas (de saída), e podem ser obtidos de muitas maneiras como por exemplo discretização de equações diferenciais, funções de transferência no domínio de Laplace, funções transferência no domínio da frequência, métodos determinísticos, métodos estocásticos; estes métodos representam uma gama grande de metodologias para se obter uma representação para o processo; metodologias inerentes a este trabalho serão descritas melhor no capítulo posterior.

É importante considerar que a etapa de definição da **estratégia de controle** a ser utilizada pode depender da etapa citada anteriormente, e considerando-se que tudo esteja de acordo teoricamente o sistema deve ser controlado dentro das faixas requeridas.

Dentro destas estratégias de controle é muito comum na indústria se utilizar o controlador PID ( Proporcional, Integral , Derivativo), o qual têm como função principal calcular a ação de controle que deverá ser realizada em uma planta industrial com base em dados das variáveis controladas visando alterar as variáveis manipuladas no sentido

de minimizar o erro entre o valor lido e o calculado. Maiores detalhes sobre o controlador PID serão apresentados no próximo capítulo. O controlador PID é um conjunto de equações as quais são dependentes de fatores conhecidos como parâmetros do controlador e são determinados a partir do modelo estimado do processo, considerando seu comportamento com relação as variáveis em questão previamente conhecido, ou seja; sua dinâmica conhecida e invariável para aquele caso.

Mas, nos casos em que não há possibilidade de se obter o modelo fenomenológico do processo, por falta de informações ou pela complexidade do sistema, tanto a tarefa de identificação do processo quanto a de controle tornam-se muito difíceis. Consideremos ainda que o processo sofra alterações em sua **dinâmica** de comportamento como por exemplo a mudança da composição de óleo alimentada na coluna de fracionamento, ou variações na pressão de vapor de aquecimento de um forno, mudanças estas não previstas no modelo, que até então não estavam sendo levadas em conta. Quando isso ocorre torna-se necessário se atualizar os parâmetros do controlador PID por exemplo, a fim de que este possa tomar ações de controle que sejam representativas no processo, dentro dos parâmetros esperados.

Este tipo de problema é comum na indústria de petróleo e gás e algumas técnicas para contorná-lo serão aprofundados neste trabalho, como a utilização de **algoritmos genéticos**.

Um outro problema, o qual pode ser identificado é o **tempo morto**, que nada mais é do que tempo que um sistema leva para responder a um estímulo, como por exemplo a manipulação de uma variável de entrada. Este é problema comum em processos de troca de calor por exemplo. O estudo de estratégias que visem reduzir a influência do tempo morto no desempenho de um sistema de controle é um campo vasto de pesquisas, onde podemos citar algumas estratégias utilizadas como aplicação do Preditor de Smith por exemplo.

Com o objetivo de se explorar maneiras de se tratar estes problemas de identificação e controle de processos apresentando dinâmica variável e tempo morto é a que nos propomos como objeto neste trabalho. Como objetivo final pretende-se propor uma metodologia para ajuste automático de controladores PID em processos com as características citadas anteriormente, e um breve comentário a respeito do uso e aplicação de controladores e técnicas de ajuste de parâmetros segue posteriormente.

Sabe-se que a indústria de petróleo têm tido grande dedicação ao desenvolvimento de técnicas de controle de processos que sejam cada vez mais

eficientes, de maneira a satisfazer necessidades de operação de plantas as quais operam em condições das mais variadas possíveis e com níveis de risco muito grandes. Estes “riscos” devem ser reduzidos ao mínimo para a parcela humana necessária na operação destas plantas e com isto torna-se necessário o aprimoramento das técnicas de controle automatizado nas mesmas, o que pode além de reduzir os riscos de exposição de pessoal em áreas críticas, também pode reduzir a possibilidade de falhas humanas em processos altamente complexos a sistemas acoplados os quais podem convergir a estados críticos muito rapidamente comprometendo a operação, equipamentos, colocando em risco vidas e causando prejuízos tanto financeiros quanto humanos e ambientais.

Com base nestes objetivos, têm-se desenvolvido ao longo dos anos métodos de controle de processos automáticos, os quais são desenvolvidos nas mais diferentes plataformas e com os mais variados objetivos. Estes métodos estão intimamente ligados a protocolos de transmissão de dados, como tecnologias de FieldBus Foundation, ProfiBus; dentre outros, os quais definem protocolos e procedimentos de transmissão de dados em ambientes de risco, com o objetivo de controlar processos industriais a partir de sensores e atuadores analógicos e/ou digitais, os quais utilizam teorias de controle clássico ou digital para atuar nos processos a fim de manter determinados parâmetros e condições durante a operação.

Também outra abordagem amplamente utilizada na indústria é o desenvolvimento de sistemas de controle baseados em CLP's, controladores lógico programáveis, o uso de Controladores de Modelo Preditivo (MPC), de Modelo Preditivo Não-Linear (NLMPC), Henson & Seborg (1997) descrevem a aplicação de controladores MPC e NLMPC para fins de controle de um Reator de Polimerização, apresentando resultados interessantes; sendo que a indústria Petroquímica têm um grande gama de aplicações na área de derivados como Polímeros, que representam uma área importante de pesquisa e desenvolvimento de metodologias de controle nesta área; cabe ressaltar que a indústria Petroquímica, partindo geralmente ou da nafta, que é uma fração líquida do refino do petróleo, ou do próprio gás natural tratado, a partir de sofisticados processos petroquímicos são capazes de quebrar, recombinar e transformar as moléculas originais dos hidrocarbonetos presentes no petróleo ou no gás, gerando, em grande escala, uma diversidade de produtos, os quais, por sua vez, irão constituir a base química dos mais diferentes segmentos da indústria em geral. Atualmente, é possível identificar produtos de origem petroquímica na quase totalidade dos itens industriais consumidos pela população tais como embalagens e utilidades domésticas de

plástico, tecidos, calçados, alimentos, brinquedos, materiais de limpeza, pneus, tintas, eletro-eletrônicos, materiais descartáveis e muitos outros.

Podemos descrever a aplicação de controladores lógicos onde utiliza-se métodos de o controle com base em uma lógica de circuitos elétricos os quais determinam pontos de processo com base em dispositivos on-off; e com isto se pode ter um controle robusto de processos mas nem sempre com condições ótimas de operação; os controladores com base em modelos preditivos são também estudados para processos industriais os quais possuem muitas entradas e saídas do processo caracterizando uma outra abordagem da teoria de controle, os mesmos podem ser utilizados com sucesso para uma gama de processos industriais mas o ajuste de seus parâmetros traz a necessidade de uma base computacional muito grande, o que torna difícil seu uso com boa eficiência em processos altamente dinâmicos.

O outro enfoque que é o uso de controladores PID (Proporcional Integrativo Derivativo) têm se mostrado uma solução muito prática para os mais variados sistemas industriais, devido à necessidade de recursos computacionais bem menor que os modelos preditivos e facilidade de ajuste manual de seus parâmetros. Com a utilização destes controladores são desenvolvidos muitos sistemas de controle dedicados ou específicos para determinados processos na indústria, que têm demonstrado boa receptividade no meio. Com a utilização da teoria de controle, pode-se desenvolver muitos métodos e estratégias de controle para se utilizar estes controladores de maneira satisfatória na indústria mas, este grau de satisfação vai depender muito de alguns fatores como:

- 1 – Abordagem do problema e desenvolvimento do modelo do processo;
- 2 – Definição dos parâmetros ótimos de operação;
- 3 – Definição das variáveis controladas e manipuladas;
- 4 – Determinação dos parâmetros do(s) controlador(es) utilizados no sistema de controle;
- 5 – Testes operacionais a fim de determinar se as condições de tempo de resposta do sistema, over-shoot, under-shoot, fator de amortecimento e limites de operação estão sendo satisfatoriamente satisfeitos para o processo em questão;
- 6 – Fixação dos parâmetros de operação.

Neste procedimento, temos que notar que a determinação dos parâmetros do controlador é o ponto determinante do sucesso ou fracasso do sistema de controle em

questão. Esta etapa do processo de desenvolvimento do sistema de controle vem sendo amplamente pesquisada a fim de se determinar melhores resultados para estes parâmetros.

Em processos muito dinâmicos e com condições de operação extremas pode não ser possível se permitir ao usuário (operador) o ajuste manual destes parâmetros, então uma técnica de ajuste automático se torna necessária.

Técnicas de ajuste automático vem sido desenvolvidas ao longo dos anos, com as mais diferentes estratégias de ajuste utilizadas a partir de softwares desenvolvidos especialmente para os sistemas de controle em questão. Uma abordagem utilizada é o uso de métodos heurísticos para a determinação destes parâmetros, então ao longo desta dissertação se procura propor a utilização de um método de auto-ajuste on-line para o controlador MPC baseado em um modelo de rede neural artificial em um processo da indústria de Petróleo e gás Natural, mais especificamente da indústria Petroquímica com a aplicação de técnicas de Inteligência Artificial (Algoritmos Genéticos). Esta técnica é amplamente utilizada em diversos setores e sua utilização para este fim parece ser uma alternativa interessante para esta abordagem.

É importante descrever o que descreve CANCELIER (2004) sobre as unidades petroquímicas as quais *“abrange processos físico-químicos de transformação que apresentam fortes não-linearidades, além de múltiplos estados de equilíbrio. Ainda, para processos contínuos, alguns destes estados de equilíbrio são instáveis em malha aberta. Mesmo assim, grande parte dos estudos realizados em controle de processos de aplicações industriais são tipicamente feitos por controladores lineares baseados em modelos linearizados, ou ainda em controladores clássicos, como o PID. Contudo, sabe-se que teorias de controle linear são válidas somente para as vizinhanças de um ponto operacional e, para sistemas com fortes não-linearidades, como é o caso da maioria dos processos químicos, a fronteira de aplicação do modelo linear em torno de um ponto torna-se muito pequena, ou seja, métodos baseados em controle linear podem não apresentar bons resultados”*.

O método de auto ajuste on-line significa repetir os passos citados anteriormente de maneira cíclica durante o processo dependendo de variações de set-point, ou de condição de operação; que podem ser produzidas por perturbações diversas no sistema, por fatores econômicos ou simplesmente pela dinâmica do processo visando a minimização da função objetivo como citado anteriormente.

O documento apresenta-se estruturado como segue: no **Capítulo 2** apresenta-se algumas considerações sobre sistemas de controle, métodos de identificação de sistemas, controlador PID e pré-sintonia de controladores PID, métodos de ajuste clássicos para controladores PID Uma introdução à teoria de Algoritmos Genéticos, características, operadores, aplicação de GA à problemas de otimização (minimização). Conceitos de Redes Neurais Artificiais, representação, funcionamento, treinamento e aplicações.

No **Capítulo 3** apresenta-se as aplicações e ferramentas utilizadas neste trabalho, Algoritmo de CLAUMANN(1999)/CANCELIER(2004) para treinamento off-line de uma RNA-Feedforward, adaptação on-line de uma RNA, os controladores utilizados no trabalho.

No **Capítulo 4** apresentam-se os Materiais e Métodos utilizados, descrição da unidade experimental utilizada e outras considerações.

No **Capítulo 5** apresentam-se os Resultados e Discussões obtidos como trabalho, identificação do sistema (tempo morto envolvido), Aplicação de um controlador PID ajustado por metodologias clássicas, Aplicação de um controlador preditivo baseado em um modelo não linear de uma RNA treinada off-line via algoritmos genéticos e Mínimos Quadrados e avaliações dos controladores aplicados ao sistema.

No **Capítulo 6** apresentam-se as conclusões, no **Capítulo 7** sugestões para trabalhos futuros, no **Capítulo 8** a bibliografia utilizada neste estudo, no **Capítulo 9** os anexos e apêndices para consulta.

## 2.Revisão Bibliográfica

Este capítulo apresenta alguns conceitos utilizados no trabalho, apresentando uma descrição das metodologias utilizadas, de forma resumida a encontrada na literatura clássica. Embora muitos conceitos apresentados sejam bastante conhecidos e várias das metodologias sejam relativamente antigas, o texto visa auxiliar a leitura sem que haja a necessidade de se buscar literatura auxiliar e facilitar a compreensão deste trabalho.

Os itens 2.3 e 2.4 que tratam do controlador PID e metodologias de pré-sintonia do PID; em controle de processos e foram inseridos a fim de seguir uma ordem de acordo com os trabalhos que realizados experimentalmente. Já o item 2.5 trata da teoria de Algoritmos Genéticos e o item 2.6 trata da teoria sobre Redes Neurais Artificiais.

### 2.1.Considerações Gerais

Sistemas de controle como citado anteriormente têm como objetivo a manutenção de condições de operação de uma determinada propriedade em um processo. A fim de se atuar em um sistema de maneira a podermos manter esta propriedade ou variável dentro de uma faixa de operação desejada lançamos mão desta ferramenta; o controle. Esta ferramenta pode ser apresentada como sendo um conjunto de componentes; computador, sensores, atuadores, placas de aquisição de dados, software supervisorio, **controlador (digital ou analógico)** bem como o próprio operador que também pode ser considerado um componente desta ferramenta.

Dentro deste contexto podemos visualizar o diagrama a seguir:

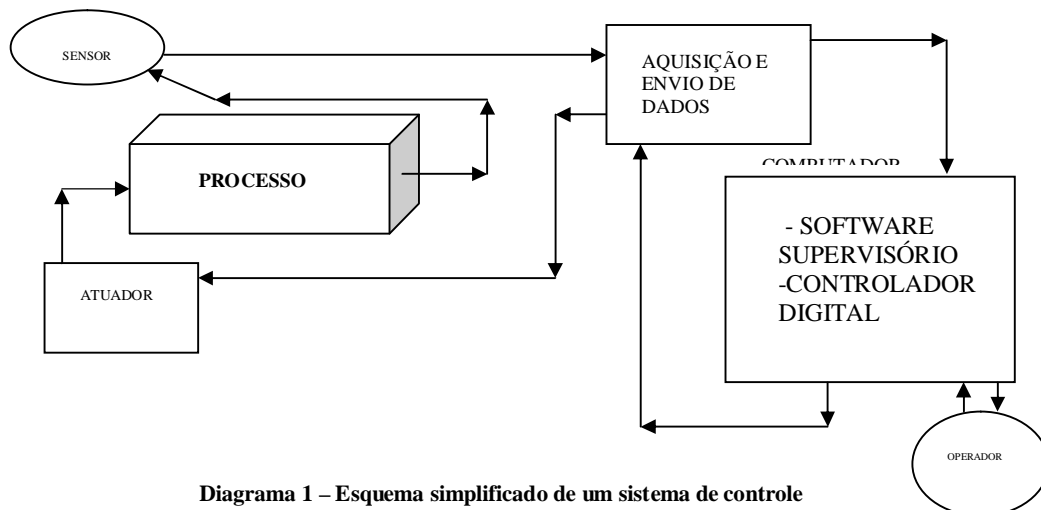


Diagrama 1 – Esquema simplificado de um sistema de controle

Observando o diagrama notamos que o papel do operador pode ser tanto de observador quanto de atuador no sistema de controle. Quanto mais acurado é o sistema de controle, menores as necessidades de que o operador interfira neste diagrama. Um sistema de controle ideal seria aquele o qual não necessitasse de nenhuma intervenção do operador.

Note que neste caso estamos apresentando um sistema de controle digital, implementado em um Computador Digital, o qual tem funções distintas como a de realizar a leitura dos dados coletados pelos sensores e apresentá-los ao operador na forma de um sistema supervisório. No sistema supervisório temos inserido nosso controlador digital que está recebendo as informações do processo e com base em um ponto de referência está calculando a ação de controle a ser tomada pelos atuadores e enviando o sinal para os mesmos, e repetindo este processo ciclicamente por meio automático **pré-programado** no computador.

Esta conformação do sistema pode variar conforme as relações entre o valor de referência e as variáveis do processo. É comum termos como referência para os sistemas de controle a diferença entre o valor lido para uma determinada variável e o valor desejado e com base nesta diferença se lança mão de um conjunto de equações para se calcular a ação de controle que deverá ser enviada para os atuadores (válvulas, etc.), a este conjunto de equações podemos denominar como **controlador digital**.

*OGATA (1997)* faz uma distinção entre diferentes tipos de sistemas de controle, sendo eles como segue:

1.1 – Sistemas de controle **com retroação**: Sistema que mantém uma relação preestabelecida entre a grandeza de saída e a grandeza de referência, comparando-as e utilizando esta diferença como meio de controle.

1.2 – Sistemas de controle **a malha fechada**: Os sistemas de controle com retroação são freqüentemente referidos assim. Na prática podemos dizer que não existe distinção entre eles. No sistema em malha fechada o sinal atuante de erro, que é a diferença entre o sinal de entrada e o sinal de retroação (que pode ser o próprio sinal de saída ou uma função do sinal de saída e de suas derivadas e/ou integrais), excita o controlador de modo a reduzir o erro e trazer o valor do sinal de saída para o valor desejado. Ou seja, a expressão controle a malha fechada acarreta sempre no uso da retroação a fim de reduzir o erro do sistema.



2 – Sistemas de controle **em malha aberta**: Os sistemas nos quais o sinal de saída não afeta a ação de controle são conhecidos assim. Este tipo de sistema de controle não mede sinais de saída e nem atua no sistema conforme estes valores, geralmente atua com uma seqüência pré-programada ao longo do tempo, sem preocupar-se com o que ocorre no processo.

É importante observarmos que os sistemas de controle em malha fechada apresentam algumas vantagens sobre os em malha aberta pelo fato de utilizarem a retroação, fato este que torna o sistema pouco sensível a perturbações externas do sistema ou mudanças internas nos parâmetros do sistema; tornando possível que alcançando mão de componentes baratos e sem muito trabalho se obter um controle relativamente preciso de uma determinada variável em um processo, o que se torna pouco provável em um sistema em malha aberta pois este não teria a capacidade de “perceber” estas mudanças no sistema, sendo o controle em malha aberta recomendado apenas para sistemas onde as entradas são conhecidas antecipadamente no tempo e não há a ocorrência de distúrbios sobre o sistema.

Neste trabalho vamos observar o uso de sistemas em malha fechada em função das características do processo a ser estudado.

Neste momento pode parecer ao leitor que aplicar uma ferramenta de controle pode ser um tanto simples, e realmente o princípio desta aplicação é bem simples como se ode notar anteriormente.

Convém observar que num contexto geral, as etapas envolvidas a fim de se **pré-programar** os eventos a serem realizados pelo computador digital são muitas e vão desde a programação do sistema de aquisição de dados do processo, passam por toda a montagem de hardware e dos sistemas de sensores e atuadores para o sistema, até a aplicação do sistema do controle ao processo; mas, a aplicação deste sistema requer um conhecimento prévio do processo envolvido, observando as necessidades e determinando as variáveis envolvidas neste contexto para fins de aplicação de um sistema de controle. Também pode não haver este conhecimento prévio do processo então apresentam-se algumas opções para o desenvolvedor do sistema representar o mesmo matematicamente a fim de estudá-lo.

É certo que pode ser necessário se ter este conhecimento do sistema a fim de se poder organizar aquele conjunto de equações as quais representam o controlador, por meio dos **parâmetros do controlador**, de maneira adequada a fornecer valores de

respostas aos atuadores que satisfaçam as necessidades em questão; maiores detalhes a este respeito serão observados posteriormente.

Torna-se importante para o leitor ater-se ao fato de que uma representação do sistema é necessária para se desenvolver a ferramenta de controle.

**SEBORG (1989)** apresenta duas distinções gerais a serem observadas neste ponto; onde fala das filosofias de controle e modelagem:

1 – **Aproximação tradicional:** A estratégia de controle, bem como o sistema de hardware são selecionados baseados em conhecimento do processo, experiência ou intuição do desenvolvedor, após o sistema de controle ser instalado na planta, os parâmetros do controlador são determinados e o controlador é ajustado.

2 – **Aproximação baseada em modelo:** O modelo do processo é desenvolvido seguindo-se basicamente três caminhos: **(i)** podem ser usados métodos de design com base em controle clássico; **(ii)** pode ser incorporado diretamente na lei de controle, aproximando-se a partir de um ponto inicial por técnicas de controle avançadas; **(iii)** pode ser usada simulação computacional do processo a fim de se explorar a melhor estratégia de controle a ser usada e valores iniciais para os parâmetros do controlador a ser utilizado.

A aproximação baseada em modelo é inicialmente mais vantajosa por algumas razões. Primeiro, plantas modernas são geralmente altamente acopladas com relação aos fluxos de matéria e energia e este acoplamento torna a operação da planta muito difícil. Segundo, devido a satisfazer melhor às necessidades econômicas das plantas que atualmente são exigidas de trabalhar o mais próximo possível dos limites de restrição visando um desempenho econômico melhor.

De uma maneira geral o objetivo do sistema de controle é obter quatro características principais como nos apresenta **MARLIN (2000)**: (1) sinal praticamente constante, (2) resposta rápida à perturbações ou mudança de set-point (ponto de operação), (3) atraso mínimo, (4) pouca sensibilidade a perturbações do processo.

## **2.2. Identificação da Planta**

Como já introduzido anteriormente, um conhecimento a respeito do processo e/ou planta a ser controlado(a) é importante a fim de se desenvolver a ferramenta de

controle. Para tanto existem diversas técnicas e métodos que podem ser utilizados e que podem ser aproveitados para controle.

Esta identificação pode ser feita através de **modelagem matemática** fenomenológica, com base em conhecimento prévio do sistema, lançando mão de técnicas matemáticas e balanços diferenciais de massa e energia com base em engenharia de processos, físico química, reatores; enfim, baseado em métodos clássicos de modelagem matemática ou também via outras metodologias que podem ser distinguidas como identificação de sistemas.

A identificação de sistemas por meios físicos é também conhecida como modelagem conceitual, e este tipo de modelagem requer muitas vezes tempo demasiado e também extenso conhecimento dos fenômenos envolvidos no processo tornando sua aplicação muito dificultada. Este tipo de modelagem não é objeto de estudo neste trabalho.

A **Identificação de Sistemas** segundo **AGUIRRE (2000)** “*é uma área do conhecimento que estuda técnicas alternativas de modelagem matemática*”.

Uma das características destas técnicas é que pouco ou nenhum conhecimento prévio do sistema é necessário e, conseqüentemente, tais métodos são também referidos como modelagem (ou identificação) caixa preta ou modelagem empírica. Em outros casos pode ser preferível usar técnicas de identificação de sistemas para se obter modelos. Neste caso, o tipo dos modelos, as técnicas usadas e os requisitos necessários são bastante distintos dos análogos a modelagem através da natureza do processo.

Contudo, a identificação da planta é uma tentativa de representar matematicamente o processo envolvido a fim de se desenvolver e analisar a melhor estratégia de controle a ser empregada no caso.

A seguir tratamos de algumas formas ou métodos de identificação da planta relevantes a este trabalho.

### **2.2.1.Métodos Determinísticos**

Neste momento procura-se introduzir as técnicas utilizadas para determinar um modelo para o processo a ser controlado sem se considerar que possa existir ruído acoplado ao sinal enviado pelos sensores do sistema que são conhecidos como determinísticos.

Para tanto existem muitos métodos, os quais procuram representar os modelos para os processos na forma de funções transferência no domínio de Laplace (*maiores detalhes sobre Transformada de Laplace podem ser obtidos em “Process, Dynamics and Control”, Seborg, Edgar, Mellichamp, 1989, John Wiley & Sons Inc*) por exemplo, ou em outras formas que procuram representar sistemas os quais são definidos por equações diferenciais lineares, que são comuns em alguns problemas de engenharia.

Basicamente uma função transferência é uma função que modela o comportamento dinâmico de um par entrada – saída de um sistema, ou seja, descreve como uma determinada entrada é dinamicamente “transferida” para a saída de um sistema.

Um problema típico de modelagem de sistemas lineares é obter a função transferência do sistema de interesse, e como já citado anteriormente; uma das maneiras de se obter esta função é aplicando-se a Transformada de Laplace à equação diferencial que descreve o sistema. Esta equação diferencial pode ser obtida a partir de modelagem fenomenológica do sistema.

Funções transferência também podem ser obtidas diretamente a partir de dados produzidos pelo sistema usando-se métodos de identificação dos quais alguns serão abordados a seguir.

É importante lembrar o leitor que a obtenção destas funções para o sistema é importante para podermos definir as estratégias de controle e parâmetros do controlador a ser utilizado para aplicar a ferramenta de controle.

Para fins didáticos podemos considerar as funções transferência como sendo de três grupos globais:

(i) **Primeira ordem:** São funções transferência do tipo

$$H(s) = \frac{K}{\tau s + 1} \quad \text{Eq.(1),}$$

Onde **K** é o ganho, e  $\tau$  a constante de tempo. Temos que  $\tau$  caracteriza a velocidade com que o sistema responde a uma certa entrada. Muitos sistemas podem ser descritos ou aproximados por funções deste tipo. Para elucidar melhor considere um sistema  $Y(t)$  variante no tempo de primeira ordem, com  $K = 1$ , e  $\tau = 1, 20, 80$  e  $100$  respectivamente. A resposta em malha aberta para estes sistemas pode ser observada na **Figura 1** desenvolvida a partir do software **Matlab 6.0™** utilizando o Simulink™.

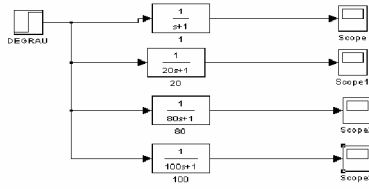


Diagrama 2 – Esquema do Simulink para obtenção da Figura 1.

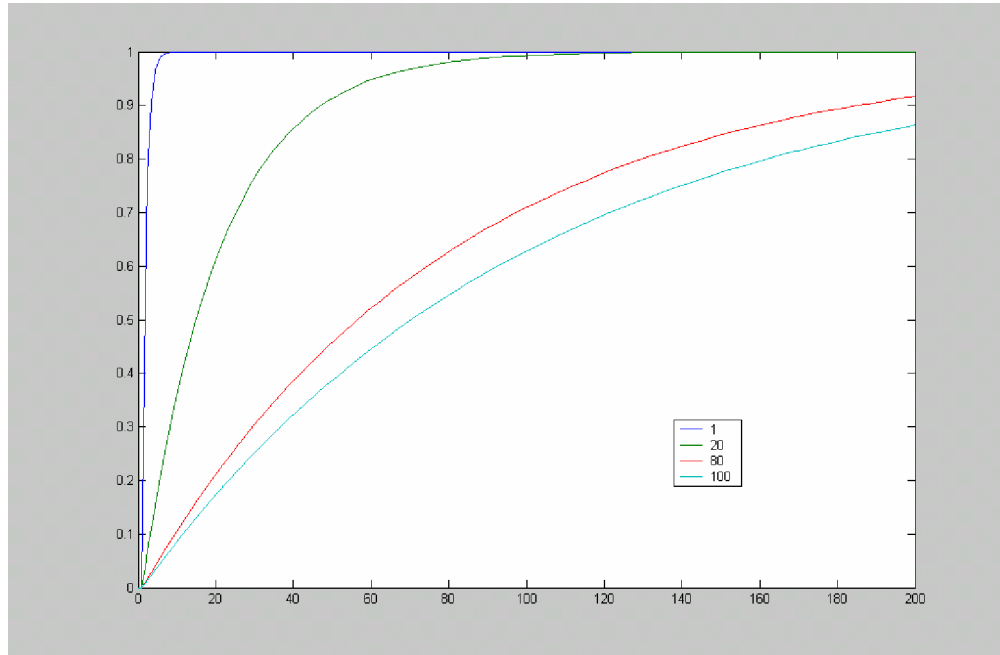


Figura 1 - Resposta degrau de sistemas de primeira ordem com diferentes constantes de tempo

Note que conforme aumenta a constante de tempo mais lenta fica a resposta do sistema.

Podemos desenvolver esta transformada a fim de se obter a função contínua no tempo ou seja a transformada inversa de Laplace para uma resposta degrau neste sistema, donde obtemos que um degrau unitário no domínio de Laplace pode ser representado por  $1/s$  logo  $\mathbf{H}(s)/s$  pode ser considerada a resposta degrau de uma função de primeira ordem, escrevemos então:  $\frac{H(s)}{s} = \frac{K/\tau}{s(s+1/\tau)}$  e decompondo em frações parciais obteremos que a Transformada Inversa de Laplace  $\mathbf{L}^{-1} = \mathbf{y}(t) = \mathbf{K} (1 - e^{-t/\tau})$ .

(ii) **Segunda ordem:** São funções transferência do tipo

$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad \text{Eq.(2).}$$

Sendo que a frequência natural  $\omega_n$  e o fator de amortecimento  $\xi$  são os parâmetros que determinam as características dinâmicas do sistema em estudo. Notemos que esta função é normalmente representada no domínio de Laplace na forma da frequência. Este tipo de função representa normalmente sistemas oscilatórios onde se  $\xi < 1$ , teremos na análise de resposta da frequência a ocorrência de pólos complexos conjugados, e existem técnicas para analisar a estabilidade do sistema e parâmetros dos controladores para este tipo de sistemas. Por outro lado, se  $\xi \geq 1$ , temos um sistema de segunda ordem dito com pólos reais e estes podem ser analisados utilizando-se o princípio da superposição.

O fator de amortecimento  $\xi$  determina o grau de oscilação do sistema como podemos notar na **Figura 2** desenvolvida a partir do software **Matlab 6.0™** utilizando o Simulink™, quanto maior o termo de  $\xi$  mais oscilatório se apresenta o sistema.

Para este tipo de funções temos alguns novos termos os quais surgem que são dependentes do valor de  $\xi$ : sub amortecido ( $0 < \xi < 1$ ), criticamente amortecido ( $\xi = 1$ ) e sobre amortecido ( $\xi > 1$ ). O esquema para obtenção da Figura 2 é apresentado abaixo.

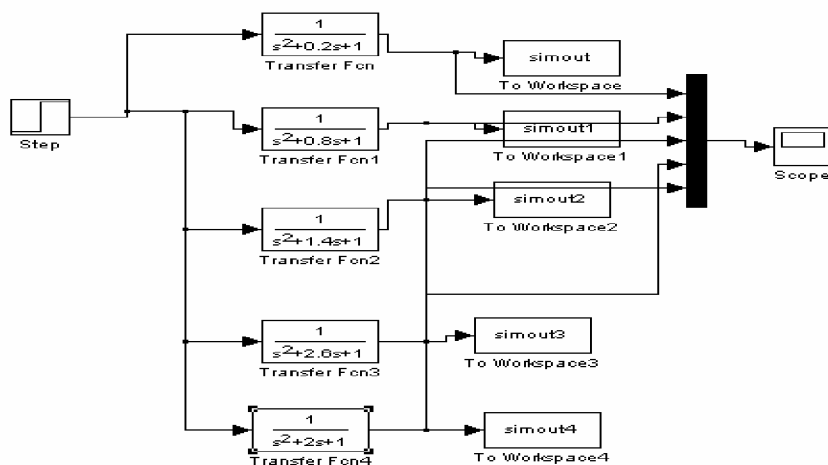


Diagrama 3 – Esquema para obtenção da Figura 2.

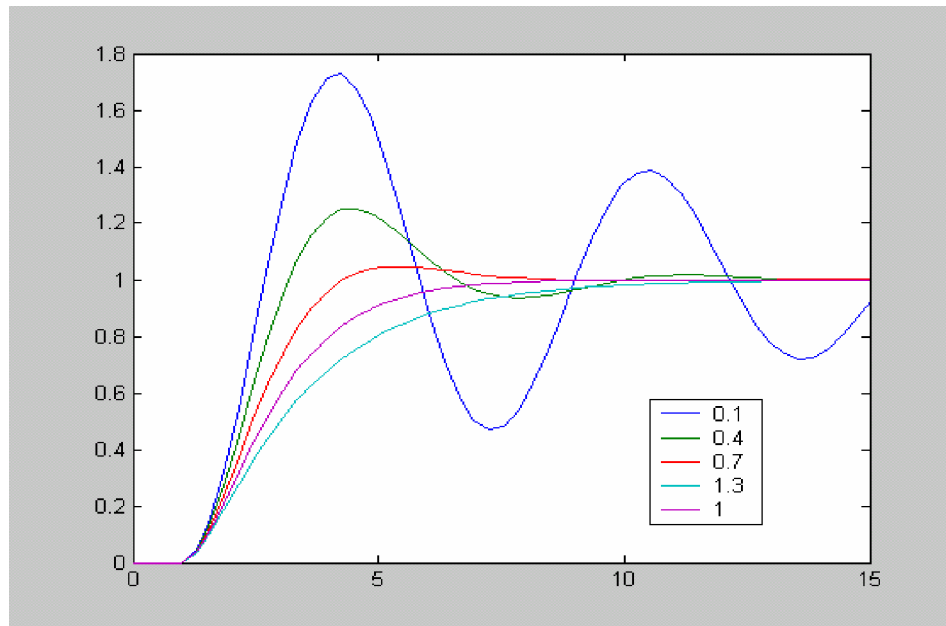


Figura 2 - Comportamento de sistemas de segunda ordem a resposta degrau para

$$H(s) = \frac{1}{s^2 + 2\xi s + 1} \text{ onde } \xi = 0.1, 0.4, 0.7 \text{ e } 1.3 \text{ respectivamente.}$$

Até que o sistema entre em regime estacionário dizemos que este encontra-se em regime estacionário. Alguns sistemas podem não atingir estado estacionário e são ditos sistemas instáveis. Observemos que analisamos apenas um pequeno intervalo de tempo sendo que se passarmos para uma amostragem maior poderemos ver como se comportam os sistemas com diferentes fatores de amortecimento como segue na **Figura 3**.

Podemos notar que quanto menor o valor de  $\xi$  maior o tempo para o sistema atingir o estado estacionário e vemos também que um sistema de segunda ordem sobre amortecido  $\xi > 1$  poderia se aproximar de um sistema de primeira ordem como vemos na **Figura 4**, para um valor grande de  $\xi = 10$  e no sistema de primeira ordem  $\tau = 20$ .

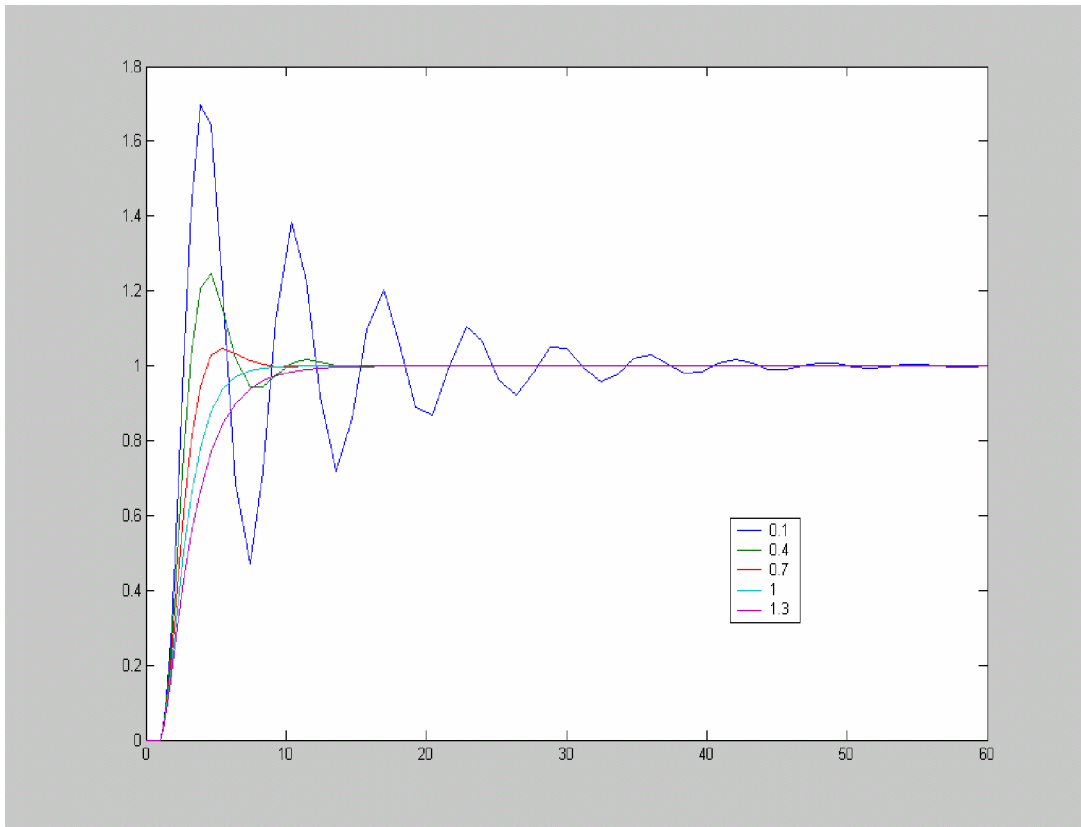


Figura 3 - Resposta de um sistema de segunda ordem para diferentes fatores de amortecimento – quanto menor o fator maior o tempo para regime estacionário.

O esquema para obtenção da Figura 4 é apresentado a seguir:

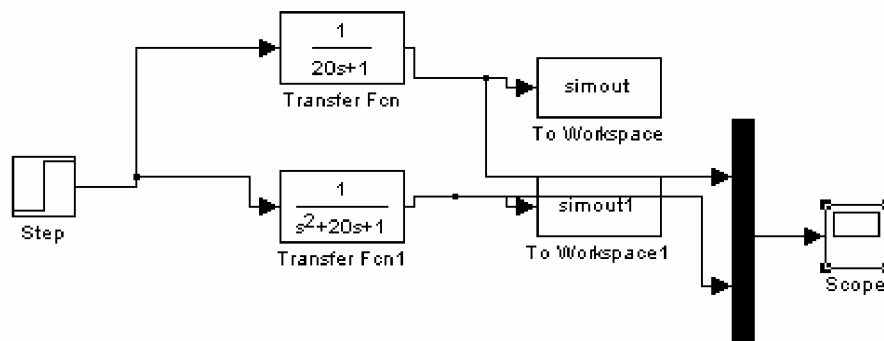


Diagrama 4 – Esquema para obtenção da Figura 4.



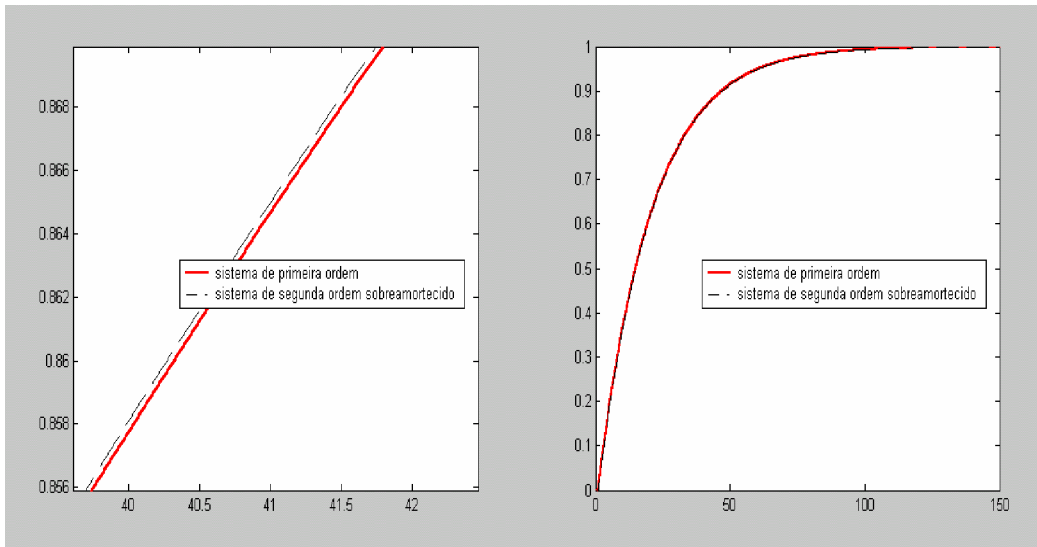


Figura 4 - Sistema de primeira ordem versus sistema de segunda ordem sobre amortecido

Este tipo de sistemas também podem ser chamados de sistemas com resposta transitória; dentro deste regime transitório, podemos identificar alguns termos que podem auxiliar na identificação do sistema e no seu ajuste, são eles: tempo de atraso ( $t_d$ ), tempo de subida ( $t_r$ ), instante de pico ( $t_p$ ), máximo valor de passagem ( $M_p$ ), tempo de acomodação ( $t_s$ ). Podemos observar na **Figura 5** estes valores.

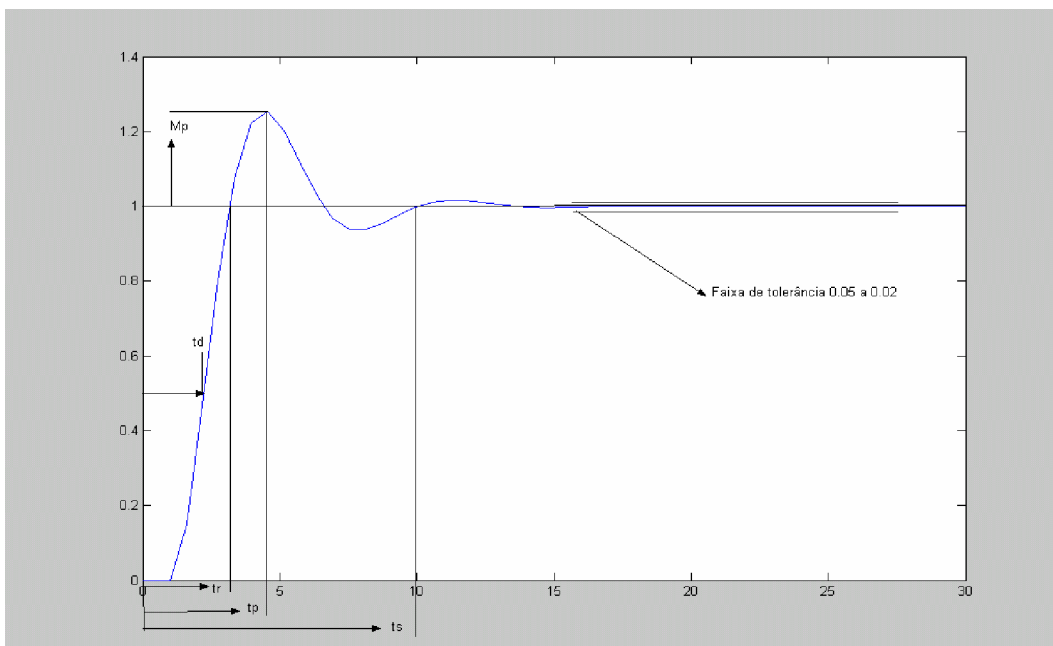


Figura 5 - Ilustração das especificações de regime transitório

Podemos definir estas especificações como: (i) tempo de atraso  $t_d$ : é o tempo necessário para que a resposta alcance, pela primeira vez; a metade do valor final; (ii) tempo de subida  $t_r$ : é o tempo necessário para que resposta passe de 10 % a 90 %, de 5% a 95 % ou de 0% a 100% de seu valor final. Para sistemas de segunda ordem sub amortecidos normalmente se usa o tempo de 0% a 100% e para sistemas sobre amortecidos normalmente de 10% a 90 %; (iii) Instante de pico  $t_p$ : é o tempo necessário para que a resposta alcance o primeiro pico; (iv) Máxima ultrapassagem (percentual)  $M_p$ : é o máximo valor de pico da curva de resposta medido. Quando o valor do regime estacionário difere da unidade é comum usarmos o valor percentual que é o valor no pico subtraído do valor no estado estacionário e tudo dividido pelo valor no estado estacionário; (v) Tempo de acomodação ( $t_s$ ): é o tempo necessário para que a curva de resposta alcance valores em torno de um valor dentro de uma faixa de 2% a 5% deste valor e aí permaneça.

É importante citar que para o caso de um sistema de segunda ordem com  $\xi = 0$  a resposta transitório não decai.

Note que todas estas especificações estão no domínio do tempo e com isto podemos ter uma idéia de qual a ordem do sistema apenas observando a curva de resposta no domínio do tempo.

(iii) **Ordem maior:** São sistemas que apresentam funções transferência maiores que 2; podem usualmente ser decompostos em frações parciais em uma soma de sistemas de ordens menores a então analisados separadamente. Não são objeto deste estudo portanto somente são citados aqui.

Segundo **AGUIRRE (2000)**, “a motivação básica para decompor a função transferência é poder separa-la em módulos básicos cuja resposta pode ser facilmente obtida nas tabelas de transformada de Laplace”. Maiores detalhes sobre esta prática podem ser encontrados em AGUIRRE (2000) – pg. 79.

A seguir procura-se apresentar os métodos explorados neste trabalho para identificação da planta de controle utilizada, para registro que existem um número de metodologias grande e neste trabalho somente são abordados os métodos os quais procurou-se explorar como ferramentas para obtenção de uma função transferência satisfatória para o processo.

### Sistemas que podem ser considerados de primeira ordem

Neste ponto podemos observar uma maneira mais direta de se obter a função transferência de um processo. Com base no conhecimento da conformação das respostas dos sistemas a uma perturbação degrau; o desenvolvedor pode ter uma boa idéia da ordem do sistema a ser controlado.

Observemos um caso em que somente temos a série temporal de uma variável para uma perturbação degrau em um sistema hipotético qualquer. Podemos ter duas respostas principais as quais podem nos dar uma boa idéia da ordem do sistema como segue na **Figura 6 (a,b)** abaixo.

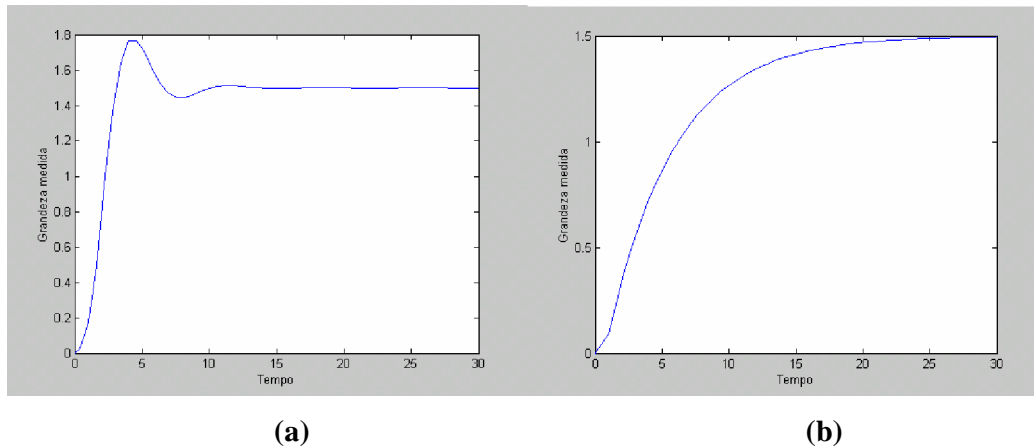


Figura 6 – Respostas principais que podem ser obtidas

A **Fig.6.a** é a resposta a um sistema de segunda ordem e observando-se a curva de resposta o desenvolvedor; sem nenhum conhecimento do processo pode ter uma idéia disto somente por observação da resposta. A **Fig.6.b** é a resposta a um sistema de primeira ordem e isto fica visível a um desenvolvedor que já tem algum conhecimento da forma das respostas deste tipo de sistema.

Com base nisto, como já foi colocado anteriormente podemos obter para o caso da Fig.6.b a função transferência do sistema considerando que este é de primeira ordem com base na resposta apresentada.

Os sistemas de primeira ordem são da forma  $H(s) = \frac{K}{\tau s + 1}$ , sabemos que estamos tratando de um método determinístico (não leva em consideração o ruído do sistema) e sendo assim; podemos observar que o ganho  $K$  é o valor absoluto da variação

da grandeza medida no instante zero até o instante de estado estacionário e esta diferença dividida pela amplitude do degrau aplicado no sistema. Considerando a equação temporal do sistema de primeira ordem  $y(t) = K (1 - e^{-t/\tau})$ , podemos notar que se considerarmos que o tempo  $t = \tau$ , ou seja, estamos considerando que o tempo decorrido é igual a uma constante de tempo, o valor da resposta do sistema sai do valor inicial (antes da aplicação degrau) e atinge 63,2 % da variação total relativa ao degrau, temos  $y(t) = K (1 - e^{-1}) = K (1 - 0,367879) \approx K (0.6321)$ , vemos que  $y(t)$  para este caso é igual a 63,2 % do valor de  $y(t)$  no estado estacionário frente à perturbação degrau ou seja; para obtermos a constante de tempo  $\tau$  basta obtermos o valor do intervalo de tempo necessário para o sistema atingir este valor frente a uma perturbação degrau. Estamos considerando um degrau de amplitude unitária.

Caso a amplitude do degrau não seja unitária é importante ressaltar que para se obter o ganho do sistema devemos dividir o valor pela amplitude do degrau aplicado.

Se sistema possuir atraso puro no tempo ou também conhecido com tempo morto; o desenvolvimento e análogo mais consideramos  $t = 0$  o tempo em que sistema começa a responder a perturbação. Maiores detalhes sobre tempo morto são citados posteriormente.

A **Figura 7** ilustra este procedimento para uma função transferência  $H(s) = \frac{10}{5s+1}$  aplicando-se um degrau de 15 unidades após decorridos 15 unidades de tempo, notamos que a resposta do sistema nos retorna exatamente os valores dos parâmetros já conhecidos  $K = 10$  e  $\tau = 5$  como ilustra a figura.

Este procedimento foi utilizado neste trabalho procurando-se considerar o sistema de primeira ordem com tempo morto a fim de se encontrar uma estimativa da função transferência para o processo em questão.

A partir desta função transferência o desenvolvedor pode obter a função contínua no tempo para avaliar melhor este processo ou utilizar a função no domínio de Laplace simuladores como Simulink<sup>®</sup>.

Esta função é utilizada também para se definir os parâmetros do controlador PID a ser utilizado o qual leva em conta a dinâmica do processo, neste caso de primeira ordem.

Maiores informações sobre o controlador PID e os métodos utilizados neste trabalho são introduzidas posteriormente.

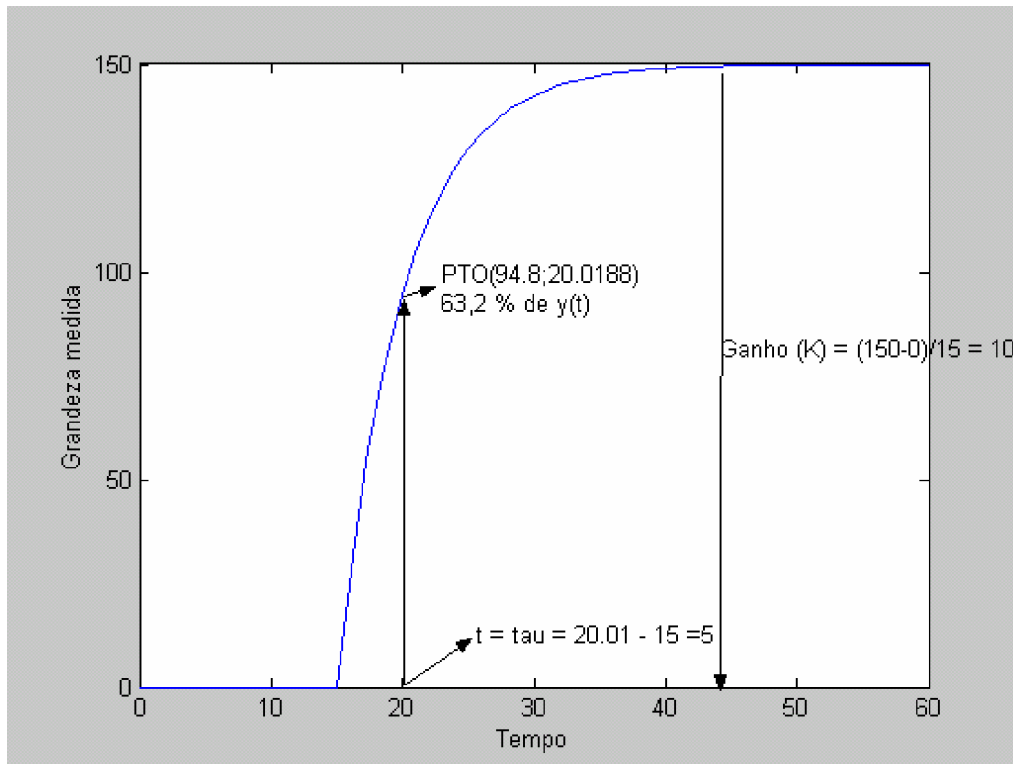


Figura 7 - Ilustração do desenvolvimento do método para obtenção dos parâmetros

Note que para este tipo de identificação de sistemas, o conhecimento prévio do desenvolvedor a respeito do tipo de resposta é essencial, como no caso do sistema estudado o sistema apresentou em alguns casos este tipo de resposta ou uma resposta muito próxima a esta; como já apresentado; “sistemas de segunda ordem podem ser aproximados a sistemas de primeira ordem para certos casos”; optou-se por utilizar este procedimento e desenvolver parâmetros de controlador PID para os sistemas encontrados.

Posteriormente resultados a este respeito são apresentados e discutidos.

### **Sistemas de segunda ordem pouco amortecidos**

Se o sistema a ser identificado apresentar uma resposta semelhante a da Fig.6.a; podemos considerar que este seria um sistema de segunda ordem pouco amortecido.

Para este caso existe um procedimento também simples para obtenção dos parâmetros do sistema (PHILLIPS & PARR, 1991); que se encontra descrito em AGUIRRE (2000) como segue.

O sistema apresenta resposta de um sistema de segunda ordem pouco amortecido (sub amortecido); sendo que  $\xi^2 \ll 1$  é uma consideração para a sistemática descrita.

A resposta de um sistema deste tipo a um degrau unitário é:

$$y(t) = 1 - \frac{1}{\beta} * e^{-\xi w_n t} \text{sen}(\beta w_n t + \phi) \quad \text{Eq.(3).}$$

Sendo que  $\beta = \sqrt{1 - \xi^2}$  e temos;

$$\phi = \tan^{-1}(\beta / \xi). \quad \text{Eq.(4).}$$

A frequência amortecida é  $w = w_n \beta$ , e o número de oscilações por segundo, ou seja; a frequência do sinal amortecido é em hertz e vale  $w/2\pi$ . Dado que o **número de ciclos do sinal amortecido no período de uma constante de tempo** é

$$: \frac{w/2\pi}{\xi w_n} = \frac{w_n \beta}{2\pi} \frac{1}{\xi w_n} = \frac{\sqrt{1 - \xi^2}}{2\pi \xi}.$$

Supomos que após **N** constantes de tempo, as oscilações não são mais perceptíveis, então o **número de ciclos visíveis na resposta do sistema a uma entrada degrau** é dado por:

$$N \frac{\sqrt{1 - \xi^2}}{2\pi \xi} \quad \text{Eq.(5)..}$$

Agora consideramos que para as funções transferência as quais queremos estimar; a resposta ao degrau se mantém numa faixa de cerca de 2% quando em “regime estacionário” após quatro constantes de tempo (caso sem tempo morto), ou seja;  $N = 4$ . Assumindo que o sistema é sub amortecido de forma a se ter  $\xi^2 \ll 1$ , o **número de ciclos visíveis** pode ser aproximado por  $0,6/\xi$ .

A exatidão desta metodologia depende da aproximação de  $\sqrt{1 - \xi^2} \approx 1$ . A frequência  $w$  pode ser estimada diretamente da série temporal. Como se assumiu que  $\sqrt{1 - \xi^2} \approx 1$ , então  $w_n \approx w$ . finalmente o ganho **K** é obtido de forma análoga a processos de primeira ordem.

Para esclarecer melhor considera a **Figura 8** que apresenta a resposta a um degrau em um sistema de segunda ordem sub amortecido com a seguinte função de

transferência conhecida  $H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$  com  $\xi = 0,2$ ;  $\omega_n = 1$ .

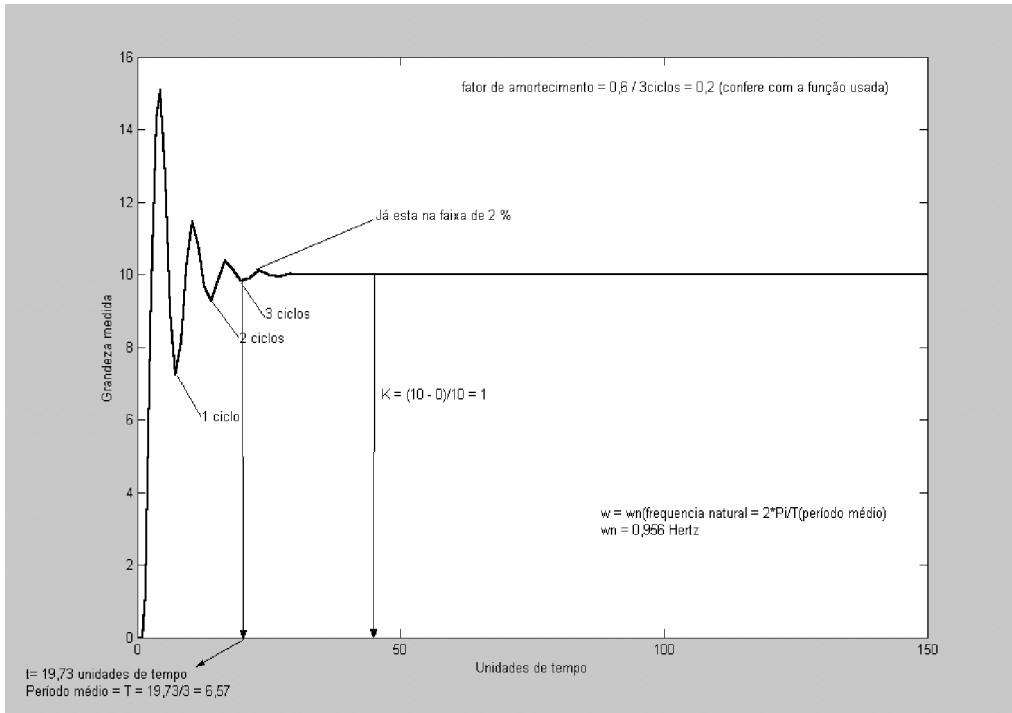


Figura 8- Aplicação do método descrito para o sistema de segunda ordem

Note que a função transferência encontrada pelo método foi:

$$H(s) = \frac{1}{0.914s^2 + 0.3824s + 0.914} \quad \text{sendo a função utilizada } H(s) = \frac{1}{s^2 + 0.4s + 1};$$

observamos as duas respostas na **Figura 10** e vemos que a forma das respostas é semelhante mais a resposta encontrada pelo método é levemente diferente que a da função transferência original; isto se deve ao fato de que a consideração de  $\sqrt{1-\xi^2} \approx 1$  não ter sido satisfeita, pois no caso  $\sqrt{1-\xi^2} = \sqrt{1-(0.2)^2} = \sqrt{0.96} \approx 0.97979$  e não 1 como deveria este é um problema desta metodologia. Mesmo assim a função transferência encontrada seria útil para se ter uma idéia do comportamento do sistema.

Na **Figura 9** podemos ter uma visão melhor do comportamento do sistema pouco antes de entrar em regime estacionário.

Existem outros métodos para identificação de sistemas de segunda ordem, apenas foi citado este método gráfico para fins didáticos sendo outros métodos determinísticos (sem considerar o ruído) para sistemas de segunda ordem abordados a seguir.

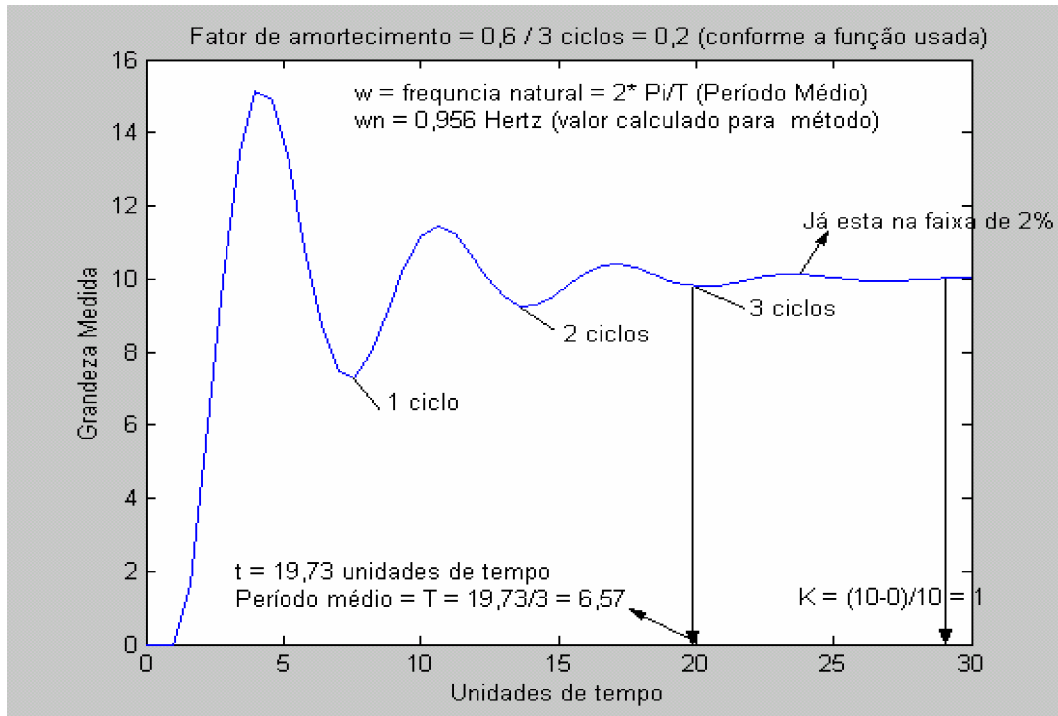


Figura 9 – Observação aproximada do regime transitório do sistema de segunda ordem analisado

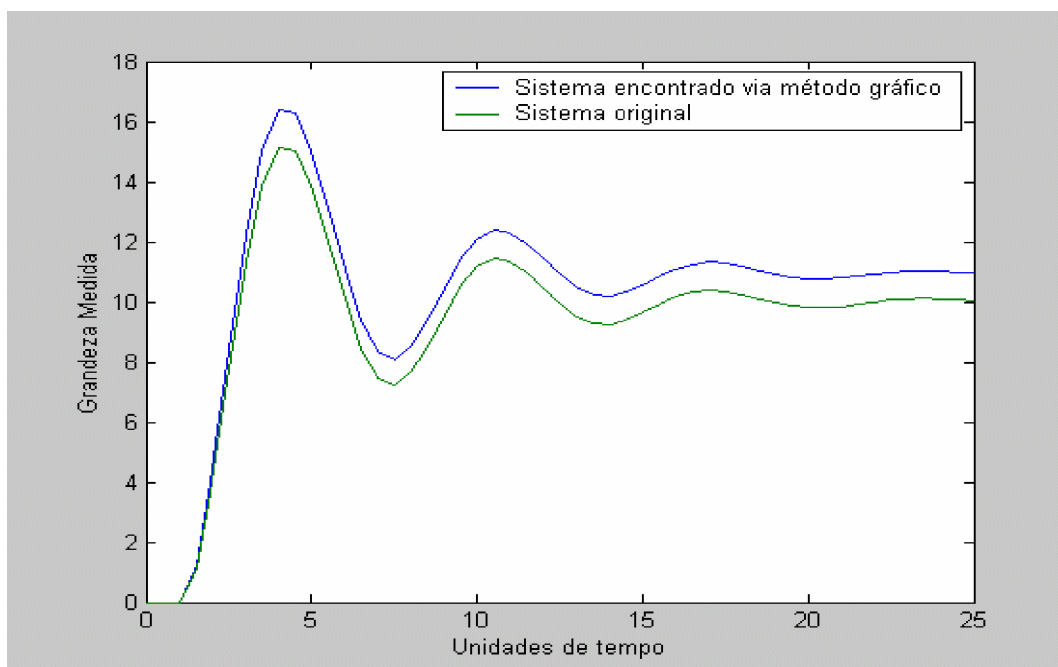


Figura 10 - Comparação do método gráfico para sistemas de segunda ordem sub amortecidos.



### **Método de Sundaesan**

Este método pressupõe que o sistema a ser analisado pode ser aproximado a funções transferência de segunda ordem do tipo:

$$H(s) = \frac{e^{-\tau_d s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad \text{Eq.(6).}$$

$$H(s) = \frac{e^{-\tau_d s} \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad \text{Eq.(7).}$$

Onde o parâmetro  $\tau_d$  representa o atraso puro no tempo,  $\tau_1$  e  $\tau_2$  são constantes de tempo fixas e os demais parâmetros já foram introduzidos anteriormente.

Este método tem como objetivo a partir de uma série de pontos os quais representam a resposta de um sistema em malha aberta a uma perturbação degrau; identificar uma função transferência que pode representar este sistema satisfatoriamente a partir de uma idéia inicial de que o processo possa ser representado por funções transferência de segunda ordem como as acima apresentadas. Este método utiliza metodologias gráficas aladas a equações matemáticas a fim de se obter os parâmetros;

O **Apêndice A** traz uma descrição deste método para o caso sobreamortecido na forma de dois exemplos aplicativos.

Pode-se ver pelos exemplos do apêndice que este método apresenta-se como um bom método para estimação de parâmetros de sistemas com este tipo de resposta. Mas pode-se encontrar dificuldades em identificar o ponto de inflexão da curva, o que pode levar a problemas de aplicação do método.

Para o caso de sistemas com resposta do tipo sub amortecido o método é similar com algumas mudanças pequenas nas equações envolvidas e no tratamento da área  $m_1$  da curva. Maiores detalhes podem ser encontrados em Aguirre(2000).

### **Mínimos Quadrados**

O método dos mínimos quadrados apresenta uma metodologia eficaz para representação dos processos. Nesta metodologia utiliza-se de um sistema de equações matricial a fim de determinar valores de coeficientes (ou estimar os parâmetros) os quais satisfaçam este sistema que é montado a partir de dados experimentais de uma

série temporal de temperatura versus tempo, ou temperatura versus abertura de válvula; por exemplo. O sistema de equações será montado conforme a aproximação a qual se deseja utilizar para o conjunto de dados.

Podemos observar o método tomando-se um conjunto de dados experimentais e procedendo o ajuste destes pontos a uma reta. Tomando-se um conjunto de pontos queremos escrever uma equação da forma  $y = f(x)$  de modo a representar da melhor maneira este conjunto de pontos experimentais. Para o caso da reta, a expressão a ser considerada é  $Y = A + B \cdot X$  donde  $A$  e  $B$  são os coeficientes ou parâmetros do modelo a ser estimado. A seguir se desenvolverá esta metodologia de maneira que se poderá perceber a mecânica do método, sendo que a expressão a ser considerada somente mudará a maneira como se monta o sistema matricial e não altera o procedimento envolvido, apenas é importante ressaltar que o caso da reta é um problema de solução única, podemos, dependendo do problema pode-se obter um sistema dito sobredeterminado, onde temos mais equações que incógnitas, visto que torna-se necessário uma pequena adequação ao procedimento.

Dado uma série de pontos ( $\{Y_1, Y_2, Y_3, \dots, Y_n\}; \{X_1, X_2, X_3, \dots, X_n\}$ ); podemos representar como um conjunto de equações  $Y_i = f(X_i)$ . Se considerarmos que  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  pode ser parametrizada conforme um conjunto de parâmetros de um vetor dito  $\theta \in \mathbb{R}$ , podemos então representar a função  $f(x)$  como  $Y = f(x, \theta)$ , e podemos escrever o sistema de equações com  $\theta_i$  incógnitas.

Para tanto são necessárias algumas considerações:

- a) A função  $f$  e o vetor  $\theta$  não variam de uma restrição para outra, ou seja, a equação é invariante no tempo;
- b) Podemos escrever  $Y = f(X, \theta)$  como  $Y = X^T \theta$ .
- c) Neste caso o número de elementos a determinar é igual ao número de restrições (ou pontos experimentais).

Com tudo, podemos ilustrar o problema com segue:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \cdot \\ \cdot \\ \cdot \\ Y_n \end{bmatrix} = \begin{bmatrix} X_1 & X_2 & X_3 & \dots & X_n \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \theta_n \end{bmatrix} \quad \text{ou seja ; } Y = X \cdot \theta.$$

Podemos notar que neste caso,  $Y$  é a variável dependente dos regressores de  $X$ , e  $\theta$  representa o vetor de parâmetros a determinar. Desde que a matriz  $X$  seja não singular ( $\text{Det} \neq 0$ ), é possível determinarmos o vetor de parâmetros  $\theta$ , invertendo a matriz de regressores, ou seja:  $\theta = X^{-1} \cdot Y$ .

Para o caso do sistema ser sobredeterminado; ou seja, o número de equações ser maior que o de incógnitas, temos que a matriz de regressores não é quadrada e portanto não pode ser invertida. Para tanto procede-se multiplicando-se todo o conjunto de matrizes por  $X^T$  (matriz transposta de  $X$ ), e vamos calcular a matriz dita pseudo-inversa de  $X$  que é:  $[X^T X]^{-1} X^T$ , e a matriz de coeficientes  $\theta$  pode ser determinada multiplicando-se a matriz pseudo-inversa de  $X$  pela matriz  $Y$ ; ou seja;  $\theta = [X^T X]^{-1} X^T \cdot Y$ . Note que a matriz  $X$  deve ser não singular.

Para exemplificar, vamos tomar um conjunto de pontos  $X = \{1 \ 2 \ 3 \ 5\}$ ,  $Y = \{0 \ 3 \ 9 \ 10\}$ , e vamos procurar fazer um ajuste destes pontos utilizando a metodologia descrita para uma reta do tipo  $Y = A + B \cdot X$ .

Queremos obter o ajuste linear acima, para tanto devemos organizar o conjunto de matrizes de forma a obtermos a equação pretendida.  $f(X) = \theta_1 + \theta_2 X$ .

O sistema é montado como segue, utilizando o software Matlab®:

```
[Ycal] = [Xcal] .[teta];
```

```
xcal = [1 1
        1 2
        1 3
        1 5]
```

```
ycal = [0
        3
        9
        10]
```

```
teta = ((inv((xcal.')(xcal)))*(xcal.')(ycal))
```

```
teta = [ $\theta_1$ 
         $\theta_2$ ]
```

```
teta = -1.57142857142857
        2.57142857142857
```

A Figura 11 abaixo pode ilustrar o resultado do ajuste linear realizado:

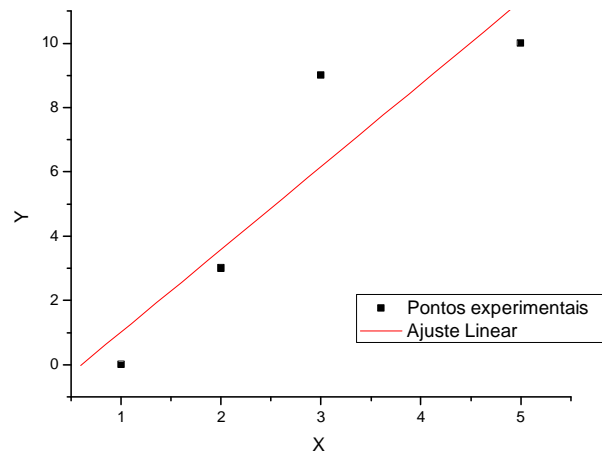


Figura 11 – Resultado de ajuste linear – comparação do método de MQ

O exemplo anterior pode ser caracterizado como uma aplicação simples da metodologia a qual envolve o método de mínimos quadrados. Para o sistema determinado sabemos que o conjunto de equações pode satisfazer simultaneamente o conjunto de restrições. Para o caso do sistema sobredeterminado, notemos que a solução obtida é apenas uma dentre as infinitas soluções para o sistema.

O **estimador de mínimos quadrados** leva em consideração que para determinada estimação a qual se procede esta implícito um erro de estimação  $\xi$  que é gerado ao se tentar aproximar os pontos experimentais a uma determinada expressão.

Podemos representar então o sistema de equações como:  $\mathbf{Y} = \mathbf{X} \cdot \boldsymbol{\theta} + \boldsymbol{\xi}$ . Sendo que  $\boldsymbol{\xi}$  é o vetor de erros cometidos ao se aproximar o conjunto de pontos da expressão desejada como  $\mathbf{X} \cdot \boldsymbol{\theta}$ .

O método de mínimos quadrados objetiva encontrar uma solução para o conjunto de equações a qual possa minimizar o erro  $\boldsymbol{\xi}$  obtido.

Para tornar a resposta mais precisa, podemos tomar o somatório do quadrado dos erros:

$$J_{MQ} = \sum_{i=1}^N \xi^T \xi = \|\boldsymbol{\xi}\|^2 \quad \text{Eq.(8).}$$

O qual representa o índice que mensura a qualidade do ajuste de  $\mathbf{X} \cdot \boldsymbol{\theta}$  ao vetor de dos Y. Portanto o estimador de mínimos quadrados visa minimizar este índice. Podemos

isolar  $\xi = \mathbf{Y} - \mathbf{X} \cdot \boldsymbol{\theta}$  e reescrevermos a expressão  $\mathbf{J}_{MQ} = (\mathbf{Y} - \mathbf{X} \cdot \boldsymbol{\theta})^T (\mathbf{Y} - \mathbf{X} \cdot \boldsymbol{\theta}) = \mathbf{Y}^T \mathbf{Y} - \mathbf{Y}^T \mathbf{X} \boldsymbol{\theta} - \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{Y} + \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\theta}$ . Problema de minimização de  $\mathbf{J}_{MQ}$  com relação a  $\boldsymbol{\theta}$  nos compele a que a seguinte expressão seja verdadeira:  $\partial \mathbf{J}_{MQ} / \partial \boldsymbol{\theta} = 0$ . Com isso vamos ter que:  $\partial \mathbf{J}_{MQ} / \partial \boldsymbol{\theta} = -(\mathbf{Y}^T \mathbf{X})^T - \mathbf{X}^T \mathbf{Y} + (\mathbf{X}^T \mathbf{X} + \mathbf{X}^T \mathbf{X}) \boldsymbol{\theta} = -\mathbf{X}^T \mathbf{Y} - \mathbf{X}^T \mathbf{Y} + 2 \mathbf{X}^T \mathbf{X} \boldsymbol{\theta}$ ; aplicando-se algumas relações de matrizes e igualando-se a expressão anterior a zero vamos obter:  $\boldsymbol{\theta} = [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{Y}$ . Note-se que esta expressão se iguala a aplicada no exemplo anterior para estimarmos os parâmetros da reta. Mas ainda, para que  $\boldsymbol{\theta}$  encontrado seja mínimo é necessário que a derivada segunda em relação a  $\boldsymbol{\theta}$  seja maior que zero, ou seja;  $2 \mathbf{X}^T \mathbf{X} > 0$ . Esta expressão é por definição positiva; portanto:

$$\boldsymbol{\theta} = [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{Y} \quad \text{Eq.(9).}$$

Representa o estimador o qual fornece o valor de  $\boldsymbol{\theta}$  e minimiza o somatório do quadrado dos erros. Este representa o estimador de mínimos quadrados.

Podemos escrever:

$$\boldsymbol{\theta}_{MQ} = \arg_{\boldsymbol{\theta}} \min \mathbf{J}_{MQ} = [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{Y} \quad \text{Eq.(10).}$$

Esta expressão representa o estimador de mínimos quadrados ordinário, sendo que existem formas diferenciadas para este estimador; como; MQP – mínimos quadrados ponderados; GMQ – estimador generalizado de mínimos quadrados; EMQ – estimador estendido de mínimos quadrados.

### **O estimador de mínimos quadrados ponderados (MQP)**

Anteriormente apresentou-se uma forma de estimador de mínimos quadrados o qual fornece a cada ponto regressor uma parcela igual do erro, no caso do MQP pode haver interesse em se definir pesos diferentes para certos valores de erro. Dependendo do grau de incerteza das medições utilizadas para estimar uma expressão podemos dar maior ou menor ênfase ao erro cometido utilizando-se estes valores, conforme a representatividade esperada das medições.

Portanto, podemos reescrever a função “custo” para o estimador de mínimos quadrados ponderados:

$$J_{MQ} = \sum_{i=1}^N \xi(i) \omega_i \xi(i) = \xi^T W \xi \quad \text{Eq.(11).}$$

Sendo que  $W$  é uma matriz diagonal cujos elementos são pesos  $w_i$ , ou seja,  $W = \text{diag}\{w_1 \ w_2 \ \dots \ w_N\}$ . E por fim vamos obter, a fim de minimizarmos a função “custo” que:

$$\theta_{MQP} = [X^T W X]^{-1} X^T W Y \quad \text{Eq.(12).}$$

Se a matriz  $W$  for definida positiva, o conjunto  $X^T W X$  também será e existirá inversa.

Cabe ressaltar que o estimador de mínimos quadrados representa uma prática ferramenta a ser utilizada a fim de se prever expressões que representem um conjunto de pontos experimentais. Para tanto, dependendo da aplicação pode se apresentar de diferentes formas, sempre mantendo a idéia geral até então apresentada.

Podemos notar no exemplo a seguir a aplicação do estimador de mínimos quadrados para um modelo **ARX-Auto Regressivo com Entrada Exógena** o qual é um tipo de modelo com características as quais a matriz de regressores vai ser gerada partir de sinais temporais em seqüências de números.

Neste caso, se pode representar o modelo de maneira geral como:  $Y(k | \theta) = \psi^T(k-1) \theta + \xi(k)$ ; sendo que  $k$  representa o instante considerado e  $\psi(k-1)$  é um vetor variáveis regressoras tomadas até o instante  $k - 1$ ; tal que  $\psi(k-1) = [\psi_1 \ \psi_2 \ \dots \ \psi_{n0}]^T$ . Neste caso temos um modelo dinâmico, mas análogo ao do caso anterior, onde temos na forma matricial:  $Y = \psi \theta + \xi$ .

Contudo, podemos aplicar a mesma metodologia aplicada anteriormente, e a função “custo” para os sistemas ARX pode ser expressa da forma:

$$J_{MQ} = \sum_{k=1}^N \xi(k | k-1, \theta)^2 = \xi^T \xi = \|\xi\|^2 \quad \text{Eq.(13).}$$

Sendo que  $\xi(k | k-1, \theta)$  é o erro de predição cometido no instante  $k$  ao fazer a predição com informação apenas até o instante  $k-1$ .

Do mesmo modo podemos representar os estimadores MQ e MQP como segue:

$$\theta_{MQ} = [\psi^T \psi]^{-1} \psi^T Y \quad \text{Eq.(14).}$$

$$\theta_{MQP} = [\psi^T W \psi]^{-1} \psi^T W Y \quad \text{Eq.(15).}$$

Pode-se reescrever estas equações da seguinte forma:

$$\theta_{MQ} = \left[ \frac{1}{N} \sum_{k=1}^N \psi(k-1) \psi^T(k-1) \right]^{-1} \left[ \frac{1}{N} \sum_{k=1}^N \psi(k-1) Y(k) \right]^{-1}. \quad \text{Eq.(16).}$$

$$\theta_{MQP} = \left[ \frac{1}{N} \sum_{k=1}^N \omega_k \psi(k-1) \psi^T(k-1) \right]^{-1} \left[ \frac{1}{N} \sum_{k=1}^N \omega_k \psi(k-1) Y(k) \right]^{-1}. \quad \text{Eq.(17),}$$

Para o caso da matriz de pesos ser diagonal, para o caso de não ser diagonal a implementação é diferente.

Para fins de implementação numérica estas últimas equações podem ser mais fáceis de serem utilizadas.

No exemplo a seguir poderemos visualizar a aplicação desta metodologia a fim de estimar os parâmetros de uma série temporal segundo um modelo ARX definido.

### Exemplo

A partir do seguinte modelo ARX:  $\psi Y(k-1) = [\theta_1 Y(k-1) \theta_2 Y(k-2) \theta_3 u(k-3) \theta_4 u(k-1) \theta_5 u(k-2)]$ , e dado o conjunto  $[u, Y]$ , determinar os coeficientes segundo o método dos mínimos quadrados.

$$Y(k) = [2 \ 5 \ 6 \ 9 \ 10 \ 18 \ 20 \ 60 \ 62 \ 66] \text{ e } u(k) = [1,5 \ 1,5 \ 1,5 \ 2 \ 2,5 \ 2,5 \ 4 \ 4 \ 4].$$

$$\text{O vetor de regressores é: } \psi(k-1) = [Y(k-1) \ Y(k-2) \ u(k-3) \ u(k-1) \ u(k-2)]^T$$

Vamos ter o seguinte sistema matricial (ou conjunto de restrições) a partir dos dados acima:

$$\begin{bmatrix} 9 \\ 10 \\ 18 \\ 20 \\ 60 \\ 62 \\ 66 \end{bmatrix} = \begin{bmatrix} 6 & 5 & 1.5 & 1.5 & 1.5 \\ 9 & 6 & 1.5 & 1.5 & 1.5 \\ 10 & 9 & 1.5 & 2.5 & 2 \\ 18 & 10 & 2 & 2.5 & 2.5 \\ 20 & 18 & 2.5 & 4 & 2.5 \\ 60 & 20 & 2.5 & 4 & 4 \\ 62 & 60 & 4 & 4 & 4 \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix}$$

Vamos calcular utilizando o software Matlab® e vamos obter os valores dos coeficientes para o modelo.

ycal =

9

10

18

20

60

62

66

fical =

6.0 5.0 1.5 1.5 1.5

9.0 6.0 1.5 1.5 1.5

10.0 9.0 1.5 2.5 2.0

18.0 10.0 2.0 2.5 2.5

20.0 18.0 2.5 4.0 2.5

60.0 20.0 2.5 4.0 4.0

62.0 60.0 4.0 4.0 4.0

>> teta = ((inv((fical.)\*(fical)))\*(fical.)\*(ycal))

teta = 1.36557593190705

-0.40857502627486

11.62450133909213

24.00949149542188

-34.16928708564466 ; estes são os coeficientes do modelo para o exemplo.



E o vetor de resíduos para este caso seria:

$$\text{erro} = (\text{ycal} - (\text{fical} * \text{teta}))$$

erro =

0.65236091662797  
 -2.03579185281833  
 -1.10049065850035  
 1.65586978579437  
 0.36683021950026  
 -0.18512637576509  
 -0.01002919722302

E o produto interno de cada regressor com o vetor de resíduos é:

$$\text{verifica} = (\text{fical}' * \text{erro})$$

verifica =

1.0e-010 \*  
 -0.29722002636845  
 -0.08000711204659  
 -0.01495692458775  
 -0.01914912672873  
 -0.02128075493602 .

Vemos que o vetor verifica é muito pequeno da ordem  $10^{-10}$  o que confirma uma característica de ortogonalidade o estimador de MQ. Já este produto vetorial representa o vetor ortogonal entre o vetor de dados e o do estimador e quanto mais próximo de zero melhor o ajuste.

## 2.3. Controlador PID

### 2.3.1. Conceitos Gerais

O controlador PID é muito utilizado em controle de processos em função de ser facilmente programável, necessitar de poucos parâmetros de ajuste e não muito suporte de recursos para operação.

Este controlador trata-se de um conjunto de equações as quais são aplicadas a fim de se produzir ações de controle através de atuadores em um processo, a partir de medições efetuadas por sensores visando um ponto ou uma faixa de operação (*set-point*).

Uma versão clássica do controlador PID pode ser apresentada como segue:

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad \text{Eq.(18).}$$

Onde “**e**” representa o erro entre a variável medida e a variável controlada (erro de rastreamento) e “**u**” é a variável de controle que é a soma dos termos: proporcional, integral do erro e derivada do erro. Os parâmetros de controle considerados para este tipo de controlador são o ganho do controlador  $K_p$ , a constante de tempo integrativa  $T_i$  e a constante de tempo derivativa  $T_d$ .

Esta equação correlaciona os termos os quais executam as ditas ações de controle, proporcional, integral e derivativa.

#### Ação proporcional

A ação proporcional é representada pelo termo  $K_p e(t)$  que implementa a ação básica de realimentação (negativa) proporcional ao erro de rastreamento. Desta forma, só é ativa enquanto o erro for diferente de zero. Conseqüentemente, para reduzir o erro de regime é preciso um valor alto do ganho  $K_p$ . Entretanto, na prática não é possível aumentar arbitrariamente o ganho tendo em vista a saturação dos atuadores, a amplificação do ruído e a estabilidade do sistema em malha fechada.

### **Ação Integral**

Para o caso em que se quer rastrear um sinal tipo degrau, a função principal da ação integral é assegurar que o erro estacionário seja sempre zero. Com este termo, um valor de  $e$  diferente de zero, mesmo sendo muito pequeno, sempre levará a uma mudança na variável de controle  $u$ , que se manterá constante se o erro for nulo. Assim, se o sistema for estável, o erro em regime a um comando em degrau será necessariamente zero.

### **Ação derivativa**

O propósito usual da ação derivativa é melhorar a estabilidade transitória do sistema em malha fechada, agindo como um termo de amortecimento. A derivada do erro pode ser interpretada como uma predição do erro no tempo  $(t + T_d)$  observado no tempo  $t$ , como pode-se notar na expansão em série de Taylor de  $(e+T_d)$ :

$$e(t+T_d) = e(t) + T_d \frac{de(t)}{dt} + \dots \quad \text{Eq.(19).}$$

### **2.3.2. Modificações no algoritmo do controlador PID**

Algumas modificações utilizadas na aplicação do controlador PID são descritas no **Apêndice B**.

### **2.3.3. Implementação Digital**

A expressão do controlador PID apresentada anteriormente esta baseada no domínio do tempo contínuo. Para sua implementação em microprocessadores é preciso expressar este controlador na sua forma discreta, entre outros: a amostragem e a discretização.

## Amostragem

Quando um computador é empregado para a implementação de uma lei de controle (expressão matemática a qual representa o controlador), todo o processamento de sinais é realizado entre instantes discretos do tempo. A seqüência de operações é a seguinte:

1. Esperar pela interrupção do relógio;
2. Ler a variável do processo (saída da planta) na porta de entrada;
3. Calcular o sinal de controle;
4. Escrever o sinal de controle na porta de saída;
5. Ir ao passo um.

O período  $h$  com que o sinal é amostrado deve ser suficientemente pequeno de modo a satisfazer o teorema de Nyquist, que estabelece que a frequência mínima de amostragem deve ser superior ao dobro da frequência máxima do sinal amostrado.

## Discretização

A discretização do controlador PID é feita para cada um dos seus componentes, existindo diferentes aproximações possíveis para cada um deles. Uma descrição de metodologias utilizadas para esta discretização podem ser encontradas no **Apêndice C**.

### O Controlador PID Digital

Considere-se a expressão do controlador PID para o domínio contínuo:

$$u(t) = \bar{u} + Kp \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad \text{Eq.(20).}$$

Para convertermos esta expressão a uma expressão no domínio discreto (digital equivalente), seguiremos a aproximação por diferenças finitas:  $\int_0^t e(t') dt' \approx \sum_{k=1}^n e_k \Delta t$ , para

o termo derivativo vamos utilizar:  $\frac{de}{dt} \approx \frac{e_n - e_{n-1}}{\Delta t}$ . Onde  $\bar{u}$  é o termo de bias.

A forma digital do controlador PID pode ser escrita de duas formas; a forma da **posição** e da **velocidade**.

Substituindo-se as aproximações por diferenças finitas na expressão geral do controlador PID, obtemos a **forma de posição**:

$$u(t) = \bar{u} + Kp \left[ e_n + \frac{\Delta t}{T_i} \sum_{k=1}^n e_k + \frac{T_d}{\Delta t} (e_n - e_{n-1}) \right] \quad \text{Eq.(21).}$$

Podemos escrever esta expressão em termos da **Transformada z** – vide *SEBORG* (1989).

$$D(z) = \frac{P'(z)}{E(z)} = K_c \left[ 1 + \frac{\Delta t}{T_i} \left( \frac{1}{1-z^{-1}} \right) + \frac{T_d}{\Delta t} (1-z^{-1}) \right] \quad \text{Eq.(22)}$$

Esta expressão representa o controlador PID digital em termos da transformada z conforme o algoritmo da posição, neste caso ele fornece o valor da saída do controlador diretamente. Esta expressão pode ser também derivada usando-se a aproximação por diferenças em atraso.

A **forma da velocidade** é uma alternativa de boa aplicação prática pois não é necessário se selecionar o termo de bias, nem de se comutar o somatório do erro. Desde que o estado estável nominal (ou bias d controlador) é constante, a mudança na saída do controlador  $\Delta p_n$  é dada por:  $\Delta p_n = p_n - p_{n-1} = p_n' - p_{n-1}'$ .

Isolando-se  $p_{n-1}$  e por rearranjo matemático com a expressão da posição podemos obter:

$$\Delta p_n = K_c \left[ (e_n - e_{n-1}) + \frac{\Delta t}{T_i} e_n + \frac{T_d}{\Delta t} (e_n - 2e_{n-1} + e_{n-2}) \right] \quad \text{Eq.(23).}$$

E podemos escrever esta expressão na forma da transformada z como segue:

$$\Delta p(z) = K_c \left[ (1-z^{-1})E(z) + \frac{\Delta t}{T_i} E(z) + \frac{T_d}{\Delta t} (1-2z^{-1} + z^{-2})E(z) \right] \quad \text{Eq.(24).}$$

Note que simplesmente se multiplicou a expressão anterior por  $(1-z^{-1})$ .

Notemos que o controlador apresenta apenas três parâmetros a serem ajustados,  $K_c$ ,  $T_i$ ,  $T_d$ .

A seguir na **Figura 12** podemos observar um exemplo de um processo hipotético sendo controlado por um controlador PID.

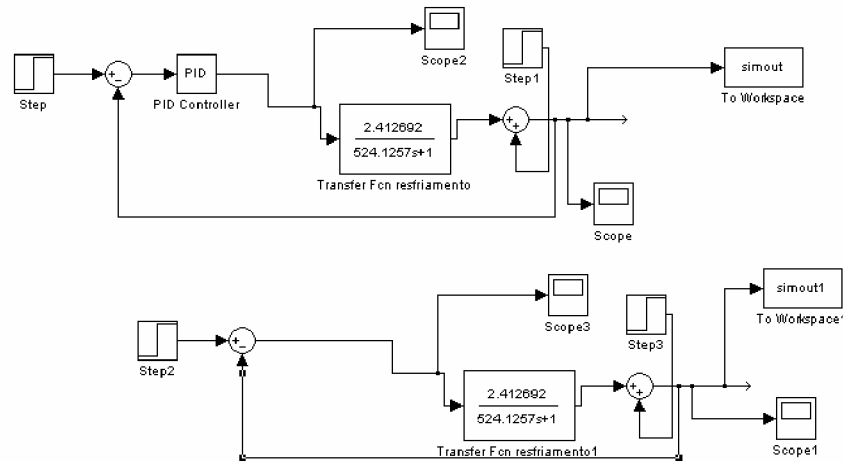


Diagrama 5 – Esquema para obtenção da simulação no Simulink®.

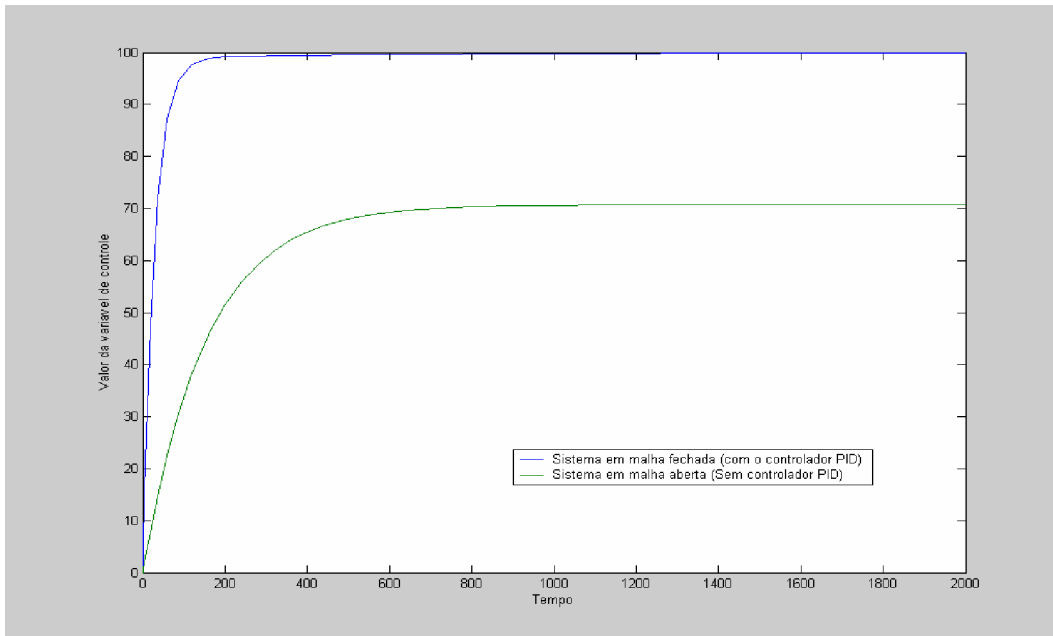


Figura 12 – Comparação entre um sistema em malha aberta e em malha fechada para variação degrau sobre o sistema

Notemos que com o controlador PID o sistema vai para o *set-point* (100 unidades), e já em malha aberta apesar de demorar mais para estabilizar, o sistema não consegue atingir o *set-point* desejado.

Neste caso utilizamos apenas para fins ilustrativos sendo que existem metodologias de ajuste de parâmetros do controlador PID as quais dependem da identificação do sistema a ser controlado, se sistema de primeira ou maior ordem (usualmente se utiliza segunda ordem); qual o valor dos parâmetros do sistema em questão, observar critérios de estabilidade, são características inerentes a determinação dos parâmetros do controlador.

Se existem não linearidades e quais as influências externas aplicadas ao sistema, podem ser perguntas a surgir. Neste caso, o controlador PID é um controlador comumente conhecido para aplicação em problemas lineares, para o caso de sistemas não-lineares, ou com dinâmica variável, pode não ser possível aplicar o controlador PID, então se lança mão de outras metodologias de controle como controladores baseados em modelo, baseados em inteligência artificial (Redes Neurais Artificiais), dentre outras metodologias existentes.

## **2.4. Metodologias para obtenção dos parâmetros do Controlador PID (pré-Sintonia do controlador)**

Neste ponto vamos abordar algumas metodologias utilizadas para determinação de parâmetros do controlador PID, são metodologias amplamente utilizadas e foram aplicadas neste trabalho com o objetivo de investigar sua funcionalidade para o sistema discutido experimentalmente.

Basicamente vamos se abordar metodologias de design de controladores baseados no critério da resposta transiente, que de uma forma geral, a partir de uma identificação preliminar do sistema a ser controlado, com os valores dos parâmetros do processo, e analisando a resposta do sistema em malha aberta a uma perturbação (degrau, rampa, etc), pode-se obter parâmetros para o controlador PID os quais forneçam um comportamento satisfatório ou ao menos um ajuste inicial bom para determinados processos, também convém ressaltar a simplicidade da aplicação destas metodologias. A seguir vamos abordar algumas metodologias as quais foram aplicadas neste trabalho.

Em suma; vamos abordar metodologias as quais se aplicam para pré-sintonia de controladores PID a partir de um modelo de processo específico e já conhecido ou identificado.

### 2.4.1. Método de Cohen-Coon

Esta metodologia é aplicável para sistemas os quais podem ser descritos como de primeira ordem e primeira ordem com tempo morto e de segunda ordem. Em 1953, Cohen e Coon apresentaram relações desenvolvidas empiricamente para sistemas os quais apresentem  $\frac{1}{4}$  de taxa de decaimento. A taxa de decaimento é a razão entre dois picos sucessivos da oscilação amortecida, como ilustrado na Figura 8. Para um sistema de segunda ordem,  $\frac{1}{4}$  da taxa de decaimento corresponde ao coeficiente de amortecimento  $\zeta = 0,2$  e 50% de overshoot para mudanças no set-point. As relações de Cohen-Coon são apresentadas na Tabela 1 a seguir.

Tabela 1- Relações de projeto de Cohen-Coon

<i>Controlador</i>	<i>Parâmetros</i>	<i>Cohen-Coon</i>
<b>P</b>	$K_c$	$\frac{1}{K} \frac{\tau}{\theta} \left[ 1 + \frac{\theta}{3\tau} \right]$
<b>PI</b>	$K_c$	$\frac{1}{K} \frac{\tau}{\theta} \left[ 0,9 + \frac{\theta}{12\tau} \right]$
	$T_i$	$\frac{\theta \left[ 30 + 3 \frac{\theta}{\tau} \right]}{\left[ 9 + 20 \frac{\theta}{\tau} \right]}$
<b>PID</b>	$K_c$	$\frac{1}{K} \frac{\tau}{\theta} \left[ \frac{16\tau + 3\theta}{12\tau} \right]$
	$T_i$	$\frac{\theta \left[ 32 + 6 \frac{\theta}{\tau} \right]}{\left[ 13 + 8 \frac{\theta}{\tau} \right]}$
	$T_d$	$\frac{4\theta}{11 + 2 \left( \frac{\theta}{\tau} \right)}$



A partir deste conjunto relações é possível se encontrar os parâmetros para o controlador PID, PI, P; que oferecem um bom ajuste inicial para o sistema a ser controlado.

No entanto esta metodologia apresenta algumas desvantagens pois considera somente os dois primeiros picos da resposta do sistema e presume que o sistema tenha taxa de decaimento de  $\frac{1}{4}$ . O controlador pode se apresentar em *loop-fechado* com características oscilatórias.

#### 2.4.2. Métodos Baseados no critério da Integral do Erro

Esta metodologia apresenta-se como uma alternativa de aproximar as relações de design do controlador baseadas em um índice que considera todas as respostas em *loop-fechado*.

Dentro deste princípio apresentam-se três índices popularmente utilizados.

Integral do valor absoluto do erro(IAE)

$$IAE = \int_0^{\infty} |e(t)| dt \quad \text{Eq.(25).}$$

Onde o erro  $e(t)$  é a diferença entre o *set-point* e o valor da variável controlada lido no instante.

Integral do quadrado do erro(ISE)

$$ISE = \int_0^{\infty} |e(t)|^2 dt \quad \text{Eq.(26).}$$

Integral do erro do tempo absoluto(ITAE)

$$ITAE = \int_0^{\infty} t |e(t)| dt \quad \text{Eq.(27).}$$

Relações de design baseadas em minimizar o critério da integral do erro foram desenvolvidas para modelos de processos simples, em particular para mudanças de *set-point* e perturbações de carga.

Em especial o critério ITAE penaliza erros que persistem por longos períodos de tempo. Em geral, ITAE é o método preferencial baseado no critério da integral do erro a fim de manter os resultados conservativos para os ajustes do controlador.

Os valores das constantes para utilização nas relações de design para minimizar o critério da integral do erro baseadas no índice ITAE são mostradas na Tabela 2. Estas relações são baseadas em um sistema de primeira ordem com tempo morto. Note que os parâmetros ótimos do controlador são diferentes dependendo se as repostas são referentes a uma mudança de *set-point* ou uma perturbação de carga. Para perturbações de carga, a função de perturbação e a função transferência do processo são consideradas de forma idêntica.

Tabela 2- Valores para aplicação do Método ITAE (b – Para mudanças de set-point, a relação de projeto para o modo integrativo é  $\tau/T_i = A + B(\theta/\tau)$ ).

<i>Tipo de entrada</i>	<i>Tipo de controlador</i>	<i>Modo</i>	<i>A</i>	<i>B</i>
<b>Carga</b>	<b>PI</b>	<b><u>P</u></b>	0,859	-0,977
		<b><u>I</u></b>	0,674	-0,680
<b>Carga</b>	<b>PID</b>	<b><u>P</u></b>	1,357	-0,947
		<b><u>I</u></b>	0,842	-0,738
		<b><u>D</u></b>	0,381	0,995
<b>Set-Point</b>	<b>PI</b>	<b><u>P</u></b>	0,586	-0,916
		<b><u>I</u></b>	1,03 <sup>b</sup>	-0,165 <sup>b</sup>
<b>Set-point</b>	<b>PID</b>	<b><u>P</u></b>	0,965	-0,85
		<b><u>I</u></b>	0,796 <sup>b</sup>	-0,1465 <sup>b</sup>
		<b><u>D</u></b>	0,308	0,929

As relações de design utilizadas são:  $Y = A(\theta/\tau)^B$  onde  $Y = K.K_c$  para o modo proporcional,  $\tau/T_i$  para o modo integrativo e  $T_d/\tau$  para o modo derivativo.

A partir dos valores pré-definidos das constantes do processo, e com os valores das constantes do método, isola-se nas equações os parâmetros do controlador e obtemos os valores dos parâmetros do controlador procurados.

### 2.4.3. Método de ajuste de Ziegler e Nichols

Este método está baseado na análise da resposta em malha aberta do sistema quando é aplicado um degrau na sua entrada.

Os ganhos do controlador são obtidos em função do tempo morto  $L$  e da taxa de reação  $R$ . A taxa de reação  $R$  é o valor da tangente no ponto de máxima derivada da saída  $y$ . A Figura 12 ilustra este caso.

Os ganhos do controlador são obtidos com a ajuda da Tabela 3.

O método da curva de reação está baseado na característica de diversos processos comumente usados na indústria que podem ser aproximadamente modelados por um sistema descrito como de primeira ordem com tempo morto; e cuja a resposta em malha aberta ao degrau têm aproximadamente a forma ilustrada na Figura 13.

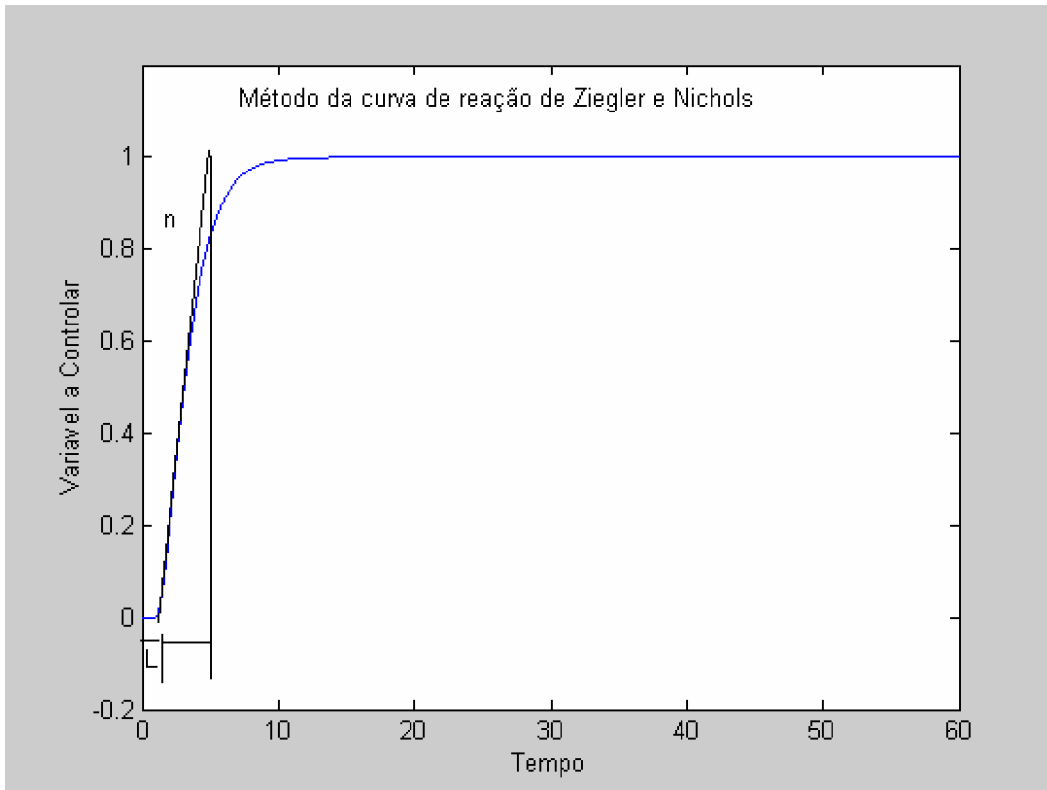


Figura 13 – Método da Curva de reação de Ziegler e Nichols

Tabela 3 - Esquema para obtenção dos parâmetros do controlador

	$K_p$	$T_i$	$T_d$
<i>P</i>	1/R L	_____	_____
<i>PI</i>	0,9/ R L	L / 0,3	_____
<i>PID</i>	1,2 / R L	L / 0,5	0,5 L

Outras metodologias podem ser encontradas no **Apêndice D**.

## 2.5.Uma Introdução à teoria de Algoritmos Genéticos

A seguir procura-se abordar a teoria que envolve basicamente uma metodologia de otimização conhecida como algoritmos genéticos. Abordar esta metodologia torna-se necessário em função de que foi utilizada como ferramenta para execução deste trabalho, a fim de avaliarmos o comportamento de um controlador baseado em um modelo de rede neural feedforward. O algoritmo genético foi utilizado a fim de se otimizar e estimar os pesos da camada escondida da rede neural em um treinamento off-line da mesma e em outra oportunidade para atualizar valores dos pesos da rede neural utilizada como modelo para o controlador preditivo não linear utilizado em um processo de controle de temperatura.

### 2.5.1.Introdução

Os Algoritmos Genéticos (AG) foram desenvolvidos por JOHN HOLLAND (1975). Tais algoritmos modificam repetidamente uma população de estruturas artificiais chamados cromossomos aplicando operadores genéticos, inspirados nos mecanismos naturais de seleção, reprodução, cruzamento e mutação.

Os cromossomos são geralmente representados por cadeias de bits que contém a informação codificada das variáveis do problema que se deseja resolver.

A etapa de seleção escolhe os melhores cromossomos na população para em seguida reproduzi-los. Na operação de cruzamento, os indivíduos selecionados se combinam por pares para trocar as suas características genéticas. O operador de mutação aplica-se eventualmente a alguns indivíduos para mudar aleatoriamente uma parte do seu material genético e introduzir diversidade na população.

Já que os indivíduos selecionados transmitem suas características aos seus descendentes, a informação histórica contida na população guia do GA para regiões mais promissoras do espaço de busca, caracterizando-se como uma metodologia de otimização.

Os GA apresentam algumas características que os tornam interessantes:

- São robustos e aplicáveis em grande variedade de problemas;

São eficazes e eficientes, já que encontram soluções ótimas para o problema em tempo razoável;

- Trabalham usando uma codificação das variáveis a serem otimizadas e não das variáveis em si;

- Ao invés de avaliar soluções individuais avaliam uma população de soluções a cada iteração;

- Necessitam somente informação sobre a função objetivo do problema, não necessitando de valores de derivada destas funções ou outra informação adicional;

- Usam regras de transição probabilísticas e não determinísticas.

Enfim; os GA requerem muito pouca informação do problema e podem otimizar funções com muitas variáveis, descontínuas ou com ruído. A única parte do GA que é dependente do problema específico que se pretende resolver é a função objetivo que se deseja otimizar. Esta é utilizada durante a etapa de seleção.

### **2.5.2. Terminologia empregada em Algoritmos Genéticos**

Sistemas naturais apresentam um ou mais cromossomos combinam-se para formar a informação genética necessária à formação de um indivíduo. As *strings*, nos sistemas genéticos artificiais, são análogas aos cromossomos nos sistemas biológicos.

A interação entre um organismo e seu ambiente, em sistemas naturais, é conhecida como fenótipo. O equivalente ao fenótipo, nos algoritmos genéticos, são as variáveis decodificadas, também chamadas de conjunto de parâmetros, solução alternativa ou ponto no espaço de solução.

Na genética, os cromossomos dividem-se em genes que são caracterizados por duas propriedades: O alelo (função do gene) e o locus (posição do gene no cromossomo). Na pesquisa genética artificial, as *strings* são formadas por um conjunto de *detetores* ou *características* que são diferenciados entre si pela função que

desempenham e pela posição que ocupam na *string*. Em codificação binária os *bits* são os detetores.

Em populações naturais, o *fitness* é a capacidade das criaturas sobreviverem aos predadores, pestes e outros obstáculos à reprodução. No ambiente artificial, o *fitness* é uma medida relativa de desempenho entre os elementos da população. O *fitness* pode ser determinado diretamente a partir do valor da função objetivo ou de algum critério baseado nela.

Em GA, a função objetivo é a função que se quer otimizar. Esta função contém a informação numérica do desempenho de cada cromossomo na população; nela estão representadas todas as características do problema que o GA necessita para realizar o seu objetivo. De uma forma geral; a função objetivo  $J$  é representada da forma:  $J = J(f(x_1, x_2, \dots, x_n))$ .

### 2.5.3. Representação Genética

Comumente é utilizada a representação binária, mas podem também ser utilizadas outras codificações. A maior parte dos trabalhos usando GA é realizada utilizando codificação binária. Nesta codificação cada cromossomo é um vetor contendo um e zero, onde cada bit representa um gen. Em problemas de otimização onde as variáveis a otimizar são números reais, a codificação binária é implementada convertendo-se o valor numérico da variável para uma expressão binária.

Ara o cálculo de cada cromossomo da população é necessário se realizar o procedimento inverso, ou seja, converter a cadeia de bits ao seu equivalente em número real. A precisão da solução vai depender da quantidade de bits utilizada para representar estes números reais.

No caso em que o problema de otimização apresenta mais de uma variável, o cromossomo da população é construído justaposto a representação genética de cada variável, formando assim uma grande e única cadeia de bits.

A desvantagem da codificação binária é que se requer constantes conversões entre expressões em ponto flutuante e estruturas binárias, o que significa um custo computacional adicional.

#### 2.5.4. Estrutura dos Algoritmos Genéticos

A estrutura básica de um GA é mostrada na Figura 14, onde podemos observar que os passos principais são:

1. Iniciar o número da geração.
2. Gerar uma população aleatória de cromossomos  $P_i$ .
3. Calcular a Função Objetivo de cada cromossomo e a sua probabilidade de sobrevivência (probabilidade de seleção).
4. Se for alcançado o objetivo final ou número máximo de gerações (iterações), terminar o processo.
5. Baseado na probabilidade de seleção, realizar a seleção e reprodução dos melhores indivíduos gerando a população  $P_{i1}$ .
6. Aplicar o operador de Cruzamento à população  $P_{i1}$  gerando a população  $P_{i2}$ .
7. Aplicar o operador de Mutação à população  $P_{i2}$  gerando a população  $P_{i+1}$ .
8. Incrementar  $i$  e voltar ao passo 3.

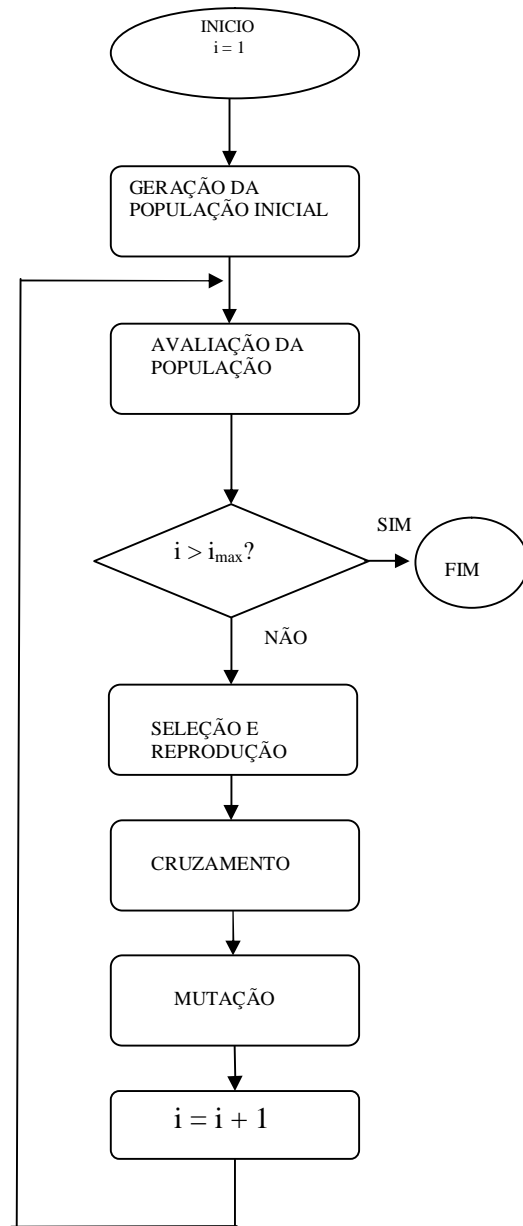


Figura 14 – Diagrama de fluxo para um AG

### 2.5.5. Geração da População Inicial

Esta é gerada de forma aleatória, porém pode ser encontrado casos em que é determinada a partir de uma seleção heurística. O importante é conter uma boa diversidade.



### 2.5.6. Diversidade Genética da População

Representa uma medida da diferença, em termos de *bits*, entre as *strings* da população. A forma mais simples de calcular a diversidade, em codificação binária, é verificar o valor de cada detetor em todas as *strings*. Caso algum detetor contenha apenas zeros ou uns (Figura 15) diz-se que houve perda de diversidade genética naquela posição. A diversidade neste caso pode ser descrita pela Eq. (28).

$$D = \frac{N_d}{N_t} \quad \text{Eq.(28).}$$

Onde: D = diversidade;

$N_d$  = número de detetores sem perda de diversidade genética;

$N_t$  = número total de detetores = comprimento da *string*.

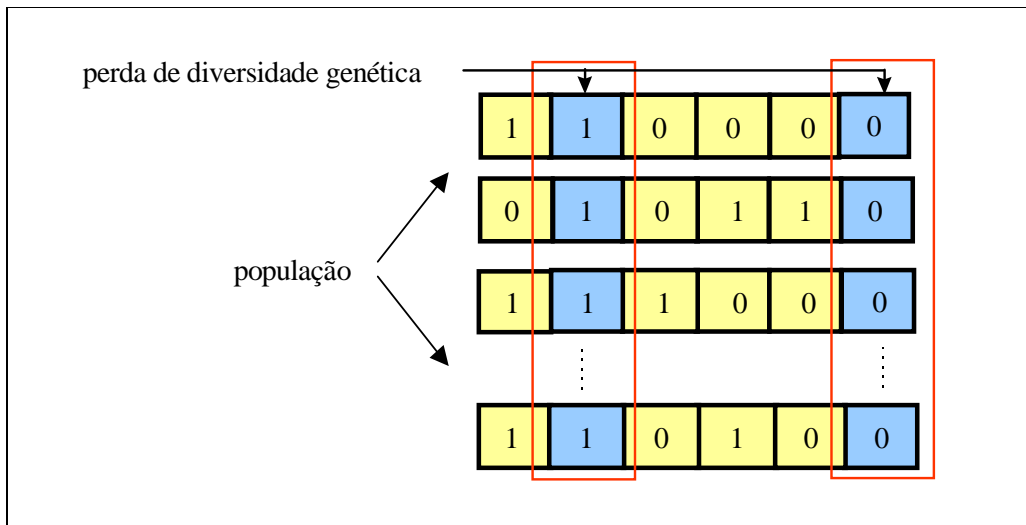


Figura 15 - Perda de diversidade em alguns genes dos indivíduos da população  
CLAUMANN (1999)

Conforme a Eq. (28) o valor de diversidade pode variar entre 0 e 1. Para  $D = 1$ , há diversidade em todos os detetores. Para  $D = 0$ , todas as *strings* são iguais.

### 2.5.7. Seleção

A seleção é o operador que determina quais *strings* serão escolhidas para posterior trocas de bits a partir do valor do *fitness*. A seleção pode ser implementada de várias formas, no entanto, os melhores indivíduos sempre devem possuir maior possibilidade de serem selecionados em relação aos piores, permitindo que os genes dos melhores passem preferencialmente às próximas gerações.

O operador de seleção utilizado no *SGA* (“Simple Genetic Algorithm – GOLDBERG/1989”) é a *rolleta*. Inicialmente, este operador atribui para cada indivíduo da população uma probabilidade de seleção proporcional ao valor do seu *fitness*. Para transformar o *fitness* em probabilidade de seleção simplesmente calcula-se o somatório do *fitness* de todos os elementos e efetua-se uma normalização. O *fitness*, no mais simples dos casos, é igual ao da função objetivo, porém, em muitas situações isto não é possível ou mesmo o mais adequado a ser feito (neste o operador de escalonamento manipula o valor do *fitness* para aumentar a eficiência do algoritmo genético). Em seguida, escolhe-se aleatoriamente um indivíduo da população, sendo que, a chance de cada um depende de sua probabilidade de seleção. Como a *rolleta* só seleciona um indivíduo por vez, este operador deve ser aplicado um número de vezes igual ao número de indivíduos da população. Devido à natureza estocástica da *rolleta*, os elementos mais adaptados da população nem sempre serão selecionados, no entanto, é mais provável que o sejam. Uma ilustração da *rolleta*, para uma população de quatro indivíduos, pode ser vista na Figura 16.

Outro operador de seleção bastante utilizado é o *tournament*. Este escolhe um grupo de  $N$  indivíduos aleatoriamente, sendo que, cada indivíduo da população possui igual probabilidade de ser sorteado. No entanto, é selecionado o indivíduo de *fitness* mais alto entre os indivíduos do grupo escolhido. Dessa forma indivíduos com *fitness* mais elevado terão maior possibilidade de serem selecionados. O *tournament* deve ser aplicado duas vezes para determinar dois indivíduos para a posterior utilização do *crossover*. O parâmetro  $N$  pode ser variado entre 1 e  $N$ . Se  $N$  é igual a 1, todos os indivíduos da população possuem a mesma probabilidade de seleção, não importando o valor do seu *fitness*, sendo que, neste caso tem-se uma seleção completamente aleatória. Se  $N$  é o tamanho da população, o melhor indivíduo dessa sempre será selecionado. Estes são os casos extremos, porém, normalmente se utiliza um valor de  $N = 2$ .

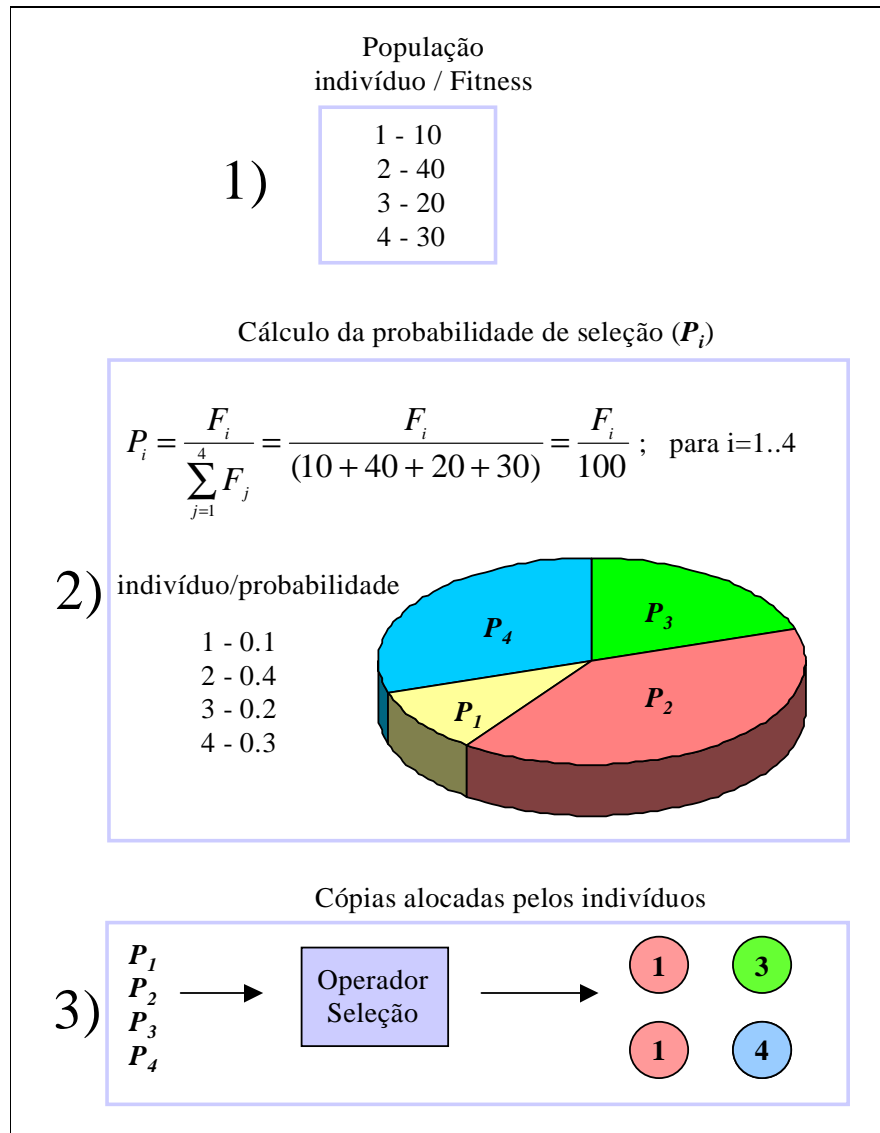


Figura 16 - Ilustração do funcionamento do *rolleta* em um população de 4 indivíduos

- (1) A cada indivíduo é atribuído um *fitness* baseado no valor da função objetivo.
- (2) Calcula-se a probabilidade de seleção de cada indivíduo.
- (3) Cópias dos indivíduos são alocadas baseadas na probabilidade de seleção.

### 2.5.8. Operadores Genéticos

#### Crossover

O operador de *crossover* é responsável pela geração de novos indivíduos a partir da combinação dos indivíduos selecionados. Para decidir se o *crossover* deve ou não ser aplicado, efetua-se um sorteio cujo resultado dependerá do valor de um parâmetro conhecido por **taxa de *crossover***. Quando se utilizam valores próximos a um para a taxa de *crossover* tem-se, na maior parte dos casos, a geração de filhos cujos genes são obtidos a partir da combinação dos pais. No outro extremo, para uma taxa de *crossover* próxima a zero, os filhos serão, em sua grande maioria, idênticos aos pais. Em geral, utilizam-se valores altos para este parâmetro, como por exemplo 0.95, pois valores pequenos tendem a diminuir a eficiência do algoritmo genético sem melhorar os resultados finais da otimização.

O *crossover* utilizado no SGA é conhecido por *crossover 1-ponto*. A partir da seleção de dois pais, o *crossover 1-ponto* sorteia uma posição ao longo do comprimento das *strings* e troca os *bits* entre estas após o ponto de *crossover*. Ver Figura 17.

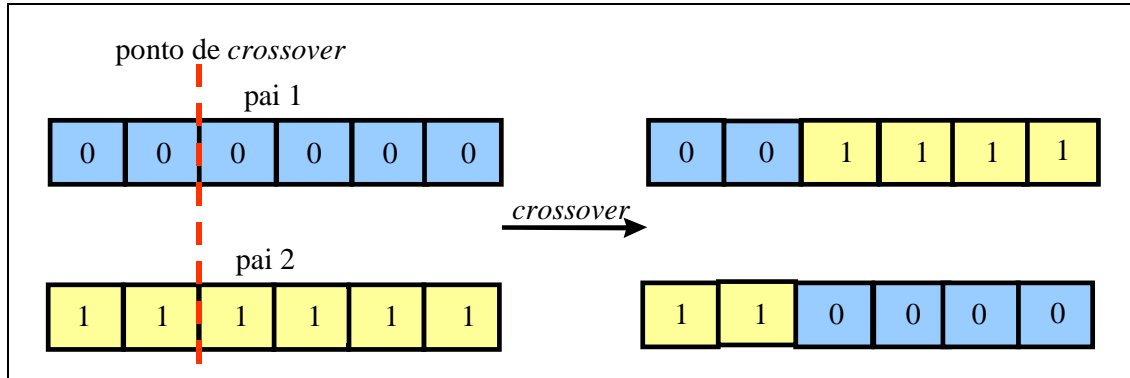


Figura 17 - Ilustração do funcionamento do *crossover 1-ponto*

Outro operador de *crossover* bastante conhecido é chamado de *crossover uniforme*. Este operador gera apenas um filho a partir de dois pais, sendo que, utiliza uma probabilidade de 50% de que qualquer *bit* no filho provenha do primeiro pai, caso contrário o *bit* provém do segundo pai. Uma variação deste operador, conhecida por *crossover parametrizado uniforme*, possui um parâmetro que determina a probabilidade de que qualquer *bit* no filho seja herdado do primeiro pai. A definição deste parâmetro em 1 resulta em um filho igual ao primeiro pai, e definindo-o em 0 resulta um filho

idêntico ao segundo pai. Este parâmetro pode aceitar qualquer valor no intervalo [0,1] para obter o *grau* de mistura desejado entre os pais.

## Mutação

Mutação é um operador que tem por objetivo evitar a perda de diversidade genética da população. A mutação é necessária ao algoritmo genético pois a utilização apenas do *crossover* e seleção pode resultar na igualdade de um ou mais detetores em todos os indivíduos da população. Nesta situação, caso o valor contido no detetor que perdeu diversidade não corresponda ao ótimo, não será possível mudá-lo por *crossover* pois todas as *strings* possuem o mesmo valor no detetor (Figura 15), e qualquer cruzamento efetuado resultará na transferência da mesma carga genética dos pais para os filhos.

O operador de mutação utilizado no *SGA* é a mutação *uniforme*, sendo que, cada gene recebe uma probabilidade (também chamada de **taxa de mutação**) para que esse operador seja aplicado. Dessa forma, deve-se percorrer a cada geração todos os *bits* de todas as *strings* da população, e para cada um deles decidir, por sorteio, se a mutação deve ou não ser aplicada. Em caso afirmativo, troca-se o valor do *bit* (Figura 18). Normalmente, valores de taxa de mutação de 1% são adequados para a maioria dos problemas. Deve-se tomar cuidado para não utilizar a taxa de mutação elevada demais, transformando o algoritmo genético em um método enumerativo.

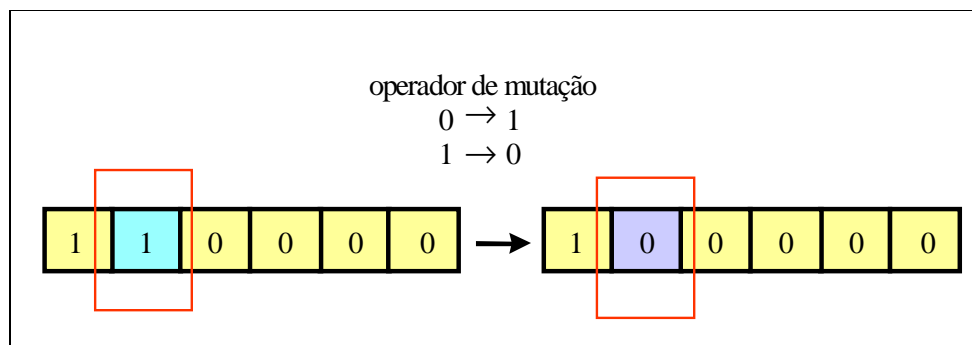


Figura 18 - Mutação ocorrida em um gene da *string*

A mutação é um operador secundário (tem pequena probabilidade de ocorrer), mas permite o melhoramento da população, sem perda de diversidade, quando utilizada de forma adequada conjuntamente com os operadores de *crossover* e seleção.

### **Efeito dos Operadores Genéticos Básicos**

- A utilização de seleção isoladamente tenderá a preencher a população com cópias do melhor indivíduo, que não é necessariamente a solução do problema.
- A utilização apenas dos operadores de seleção e *crossover* poderá levar a convergência prematura do algoritmo genético.
- O emprego de mutação exclusivamente induz um procedimento enumerativo na busca do ótimo.

### **Outros Operadores**

Nesta seção serão descritos os operadores de escalonamento e elitismo. Estes são alguns dos operadores opcionais que podem ser utilizados para aumentar a eficiência e robustez de um algoritmo genético.

#### **Escalonamento**

O escalonamento é um operador que tem por função manter a competição entre os indivíduos de uma população a níveis aceitáveis. O escalonamento pode ser utilizado conjuntamente com operadores que atribuem a probabilidade de seleção dos indivíduos de forma proporcional ao *fitness*, tal como a *rolleta*.

No início de uma otimização aparecem, freqüentemente, indivíduos muito superiores ao restante da população (super-indivíduos). Quando se utiliza a seleção por *rolleta*, os super-indivíduos assumem uma significativa proporção da população em poucas gerações. Isto é indesejável e pode levar a uma convergência prematura. No outro extremo, durante a otimização ou próximo ao seu final, os *fitness* do pior, médio e do melhor indivíduos podem estar muito próximos. Caso não seja aplicada nenhuma medida de controle dos níveis de competição, os elementos ruins, médios e melhores alocarão o mesmo número de cópias a partir deste momento. Dessa forma a busca pelo ótimo se transformará em processo enumerativo. Em ambos os casos, no começo e no final da otimização, o escalonamento do *fitness* pode melhorar o desempenho do algoritmo genético. Serão descritos quatro tipos de escalonamento:

- *Linear*;
- *Truncamento Sigma*;
- *Lei de potência*;
- Procedimentos de *ranking*.

*Escalonamento linear*: Definindo o *fitness* natural  $f$  e o *fitness* escalonado  $F$ . O escalonamento *linear* relaciona  $F$  a  $f$  de acordo com a Eq.(29):

$$F = a \cdot f + b \quad \text{Eq. (29)}$$

Os coeficientes  $a$  e  $b$  podem ser escolhidos de diversas formas, entretanto, o *fitness* escalonado médio  $F_{\text{medio}}$  é normalmente igualado ao *fitness* natural médio  $f_{\text{medio}}$  (Eq.(30)).

$$F_{\text{medio}} = f_{\text{medio}} \quad \text{Eq. (30)}$$

Para calcular os dois coeficientes da Eq.(28) deve-se definir outra relação. A mais utilizada é a seguinte:

$$F_{\text{max}} = C_{\text{mult}} \cdot f_{\text{medio}} \quad \text{Eq. (31)}$$

Onde:  $C_{\text{mult}}$  é um valor constante maior que *um*;

$F_{\text{max}}$  é o *fitness* escalonado do melhor indivíduo da população.

Para populações pequenas (50 a 100 indivíduos)  $C_{\text{mult}} = 1.2$  a  $2$  tem sido utilizado com sucesso. Uma ilustração do escalonamento *linear* pode ser vista na Figura 19.

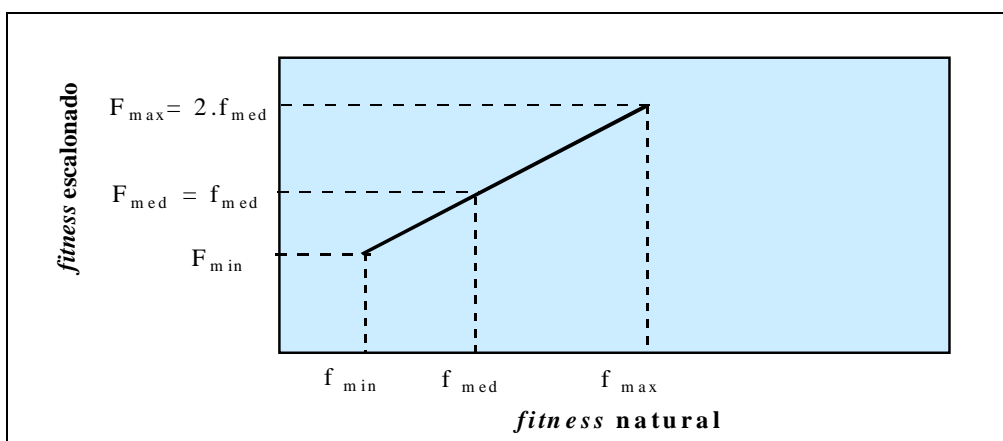


Figura 19 - Ilustração do funcionamento do escalonamento *linear*

Próximo ao final da otimização, os indivíduos da população podem ter *fitness* muito semelhantes. Nesta situação, o escalonamento aumenta a diferença entre o *fitness* do pior, médio e do melhor de modo a evitar que a otimização se torne pesquisa enumerativa. Isto, por sua vez, dificulta a aplicação do escalonamento *linear* quando algumas *strings* estão bem abaixo da média da população, porém, o *fitness* médio está próximo do máximo. Se o escalonamento é aplicado nesta situação o alargamento requerido para separar os médios dos bons fará que as *strings* ruins tenham valores negativos após o escalonamento. Quando ocorre esta situação, mantém-se o *fitness* natural médio igual ao *fitness* escalonado médio, porém mapeia-se o mínimo *fitness* natural para o *fitness* escalonado = 0. Dessa forma os parâmetros contidos no escalonamento *linear* deverão ser calculados a partir das relações Eq.(32) e Eq.(33).

$$F_{medio} = f_{medio} \quad \text{Eq. (32)}$$

$$F_{min} = 0 \quad \text{Eq. (33)}$$

Onde:  $F_{min}$  = é o *fitness* do pior indivíduo da população.

O escalonamento *linear* pode não funcionar adequadamente quando há diferenças significativas entre o *fitness* do pior, médio e melhor indivíduos da população, ou seja, nas seguintes situações: quando o *fitness* do menor está bem abaixo da médio mas o *fitness* médio está próximo do máximo ou, de forma contrária, quando o



*fitness* do menor está próximo da médio mas o *fitness* do médio está distante do melhor. O primeiro caso, já descrito acima, é resolvido modificando as relações utilizadas para o cálculo dos coeficientes da equação do escalonamento. Esta medida evita a atribuição de *fitness* negativos aos piores indivíduos. No entanto, o escalonamento *linear* atribuirá um valor de *fitness* para os indivíduos próximos a média praticamente igual ao do melhor. No segundo caso, o *fitness* escalonado do pior ficará ainda mais próximo do médio. Nos dois casos percebe-se que o escalonamento *linear* tem dificuldades de explicitar, de modo adequado, a diferença entre os indivíduos. Este comportamento tende a diminuir a eficiência do algoritmo genético pois um *fitness* praticamente igual é atribuído a indivíduos que possuem desempenhos diferentes.

*Truncamento sigma:* Neste operador utiliza-se a variância da população para transformar o *fitness*, Eq.(34).

$$F = \max [f - (f_{med} - c.\sigma), 0] \quad \text{Eq.(34).}$$

Onde: F = *fitness* escalonado;

f = *fitness* natural;

f<sub>med</sub> = *fitness* médio da população;

c = constante; normalmente entre 1 e 3;

σ = variância da população;

Max = o maior entre  $f - (f_{med} - c.\sigma)$  e 0.

*Escalonamento com lei de potência:* Neste escalonamento, o *fitness* escalonado é tomado como o *fitness* natural elevado a alguma especificada potência, Eq.(35).

$$F = f^k \quad \text{Eq. (35).}$$

Um valor normalmente utilizado para a constante k é de 1.005. Entretanto, o valor de k depende do problema a ser otimizado, e deveria ser modificado durante a corrida para alongar ou encurtar a diferença entre os *fitness* dos indivíduos da população.

*Procedimentos de Ranking:* Neste tipo de escalonamento a população é ordenada com base no valor da função objetivo no sentido do melhor para o pior (*ranking*). O valor do *fitness* é atribuído em função da posição no *ranking* e, portanto, indiretamente a partir da função objetivo. O melhor indivíduo recebe o valor mais alto e a medida em que se desce o *ranking* o *fitness* atribuído diminui.

As duas distribuições mais utilizadas na atribuição do *fitness* em função da posição do *ranking* são:

Distribuição linear: Neste caso fixa-se o valor do *fitness* para o melhor e pior elementos da população. Os elementos intermediários recebem valores entre o máximo e o mínimo respeitando a sua posição no *ranking*;

Distribuição Exponencial: Para esta distribuição fixa-se o valor do *fitness* do melhor elemento da população e atribui-se um fator de decaimento dado pela Eq.(36).

$$\frac{P_{i+1}}{P_i} = k \quad \text{Eq.(36).}$$

Onde:  $P_i$  é o  $i$ -ésimo melhor elemento do *ranking* e  $P_{i+1}$  é o  $i$ -ésimo + 1;

$k$  é uma constante entre 0 e 1, normalmente 0.9, que dá o formato exponencial a esta distribuição.

O valor absoluto de *fitness* atribuído ao melhor e pior elementos da população na lei de decaimento linear e para o melhor elemento na lei exponencial não é importante, já que o *fitness* de cada elemento da população será normalizado posteriormente para ser convertido em probabilidade de seleção.

### **Elitismo**

O elitismo é o operador que repassa os melhores indivíduos da população de uma geração para a próxima sem que estes sofram qualquer tipo de alteração.

No algoritmo genético, um excelente indivíduo pode aparecer na população em determinada geração e ser perdido devido ao caráter estocástico do método. Neste caso, muitas gerações podem ser necessárias para que seja encontrado outro semelhante. Para não perder os melhores indivíduos da população em determinada geração, costuma-se

repassá-los diretamente para a próxima geração sem que sejam alterados por *crossover* ou mutação.

O número de indivíduos repassados deve ser uma pequena fração da população pois um elitismo elevado pode levar a convergência prematura do algoritmo genético. O valor recomendado está em torno de 5 a 10 % da população.

### **Modificação da Faixa de Interesse de Otimização**

A decodificação das *strings* gera apenas valores inteiros não negativos. Porém, na maioria dos casos, esse não é o domínio de interesse da otimização e a precisão obtida é muito baixa. Para mapear o valor decodificado de cada *string* para o domínio de interesse efetua-se uma transformação linear conforme a Eq.(37).

$$X = \frac{U_{\max} - U_{\min}}{2^N - 1} \cdot I + U_{\min} \quad \text{Eq.(37)}$$

Onde:

[ $U_{\min}$ ,  $U_{\max}$ ] são os limites inferior e superior da variável de interesse;

I - é o inteiro sem sinal obtido da decodificação das *strings*;

X - variável real limitada a faixa de interesse;

N - número de *bits* na *string*.

Assim, admitindo que o tamanho das *strings* é N=4 a transformação linear resultará.

Decodificação		Transformação	
0 0 0 0	→ 0	→	$U_{\min}$
1 1 1 1	→ $2^4$	→	$U_{\max}$
outros valores são mapeados linearmente entre estes limites $U_{\min}$ e $U_{\max}$			

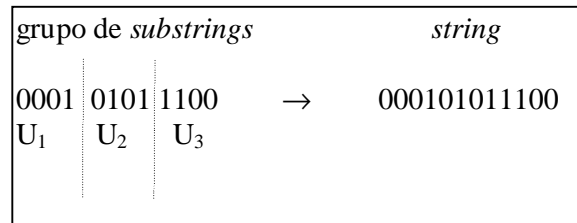
Deve-se notar que o termo:  $\pi = \frac{U_{\max} - U_{\min}}{2^N - 1}$  na Eq.(37); é a precisão e portanto,

pode-se controlar a faixa de operação e a precisão da variável pela manipulação de  $U_{\max}$ ,  $U_{\min}$  e N.

### 2.5.9.Otimização de Problemas Multivariáveis

Para resolver problemas com múltiplas variáveis pode-se simplesmente justapor todas as variáveis codificadas (*substrings*) em uma formando apenas uma *string*. Claro, cada variável terá seu próprio comprimento e valores de  $U_{min}$  e  $U_{max}$ .

Por exemplo, em um problema de otimização com 3 variáveis, onde cada uma delas é codificada como *substrings* de 4 bits, concatenam-se as *substrings* da seguinte forma:



Onde:  $U_1, U_2, U_3$  = variáveis 1,2 e 3 respectivamente.

#### Mapeamento da função objetivo para avaliação do *fitness*

Quanto maior o valor da função objetivo de um indivíduo maior também deve ser o *fitness* atribuído. No entanto, conforme comentado, o *fitness* não pode, ou mesmo não deve em muitas situações, ser igualado a função objetivo. Como por exemplo o *fitness* não poderá ser igualado a função objetivo quando essa última assumir valores negativos pois o *fitness* representa uma medida de adaptação, e posteriormente será convertido em probabilidade. Para resolver este problema faz-se o *fitness* de cada indivíduo igual ao valor da função objetivo acrescido de um valor positivo fixo para toda população em uma geração, porém, esse pode ser alterado de uma geração para outra. Os indivíduos que mesmo assim ficarem com *fitness* negativo são mapeados para zero. Matematicamente esta operação é dada pela Eq.(38):

$$\begin{aligned}
 \text{Fitness}(X) &= u(X) + C_{min} \quad \text{se } u(X) + C_{min} > 0 \\
 &= 0 \quad \text{se } u(X) + C_{min} \leq 0
 \end{aligned}
 \tag{38}$$

Onde:

$u(x)$  = valor da função objetivo;

$C_{\min}$  = Coeficiente positivo somado ao valor da função objetivo.

O coeficiente  $C_{\min}$  é normalmente escolhido como o valor absoluto do elemento com menor *fitness* da população na corrente geração ou como uma função da variância da população.

### 2.5.10. Resolução de Problemas de Minimização

Conforme a descrição do operador de seleção, os indivíduos com maior valor de função objetivo são aqueles que possuem probabilidade de seleção mais elevada. Dessa forma a pesquisa do ótimo é direcionada, com maior intensidade, em torno destes indivíduos. Este procedimento resulta numa maximização. **Para resolver problemas de minimização, utilizando algoritmos genéticos, deve-se multiplicar a função custo por menos um.** Esta operação transformará a minimização em uma maximização e não modificará a solução do problema. A transformação não garante que o *fitness* seja não negativo para todos os casos. Caso isso ocorra deve-se proceder de forma análoga ao exposto na seção que trata do **mapeamento da função objetivo**.

### Restrições

Muitos problemas práticos contém restrições que devem ser satisfeitas. Um caso típico é a otimização de um processo a partir de um modelo do mesmo. As restrições aparecem em função de limitações em equipamentos e condições de operações. Restrições são normalmente classificadas como relações de igualdade e desigualdade.

Restrições de igualdade aparecem em problemas que envolvam balanços globais de massa, energia etc.

Para tratar de restrições de igualdade deve-se colocar um procedimento de validação diretamente no *crossover*. Assim evita-se que o algoritmo genético opere com populações de indivíduos inválidos, que no caso do problema de equilíbrio de fases citado equivale ao não fechamento dos balanços globais.

Problemas com restrições de desigualdade podem ser divididos em pouco e muito restritos: Para resolver problemas do primeiro tipo, basta fazer que qualquer indivíduo que viole uma restrição não receba nenhum *fitness*. No segundo tipo, o

problema pode ter tantas restrições que encontrar uma solução possível pode ser tão difícil quanto encontrar o ótimo. A melhor maneira de lidar com este tipo de problema é penalizar as *strings* que violam as restrições, diminuindo seu *fitness*. Então um problema com restrições é transformado em um problema irrestrito pela associação de um custo ou penalidade por cada restrição violada. O método em questão é conhecido como o método das penalizações. O objetivo deste método é fazer que as *strings* que violam as restrições sejam penalizadas, porém, seu *fitness* não deve ser zerado já que pode haver boas combinações de *bits* nessas.

Matematicamente um problema com restrições:

$$\begin{aligned} &\text{minimizar } g(x) \\ &\text{sujeito a } b_i(x) \geq 0 \text{ onde } i = 1, 2, \dots, n \end{aligned}$$

Pode ser transformado em outro onde um termo de penalidade é incluído na função objetivo, GOLDBERG (1989):

$$\text{minimizar } g(x) + r \cdot \sum_{i=1}^n (\varphi |b_i(x)|)$$

onde:  $\varphi$  - função penalidade;

$r$  - coeficiente de penalidade.

A função penalidade é normalmente assumida como uma função quadrática. O coeficiente  $r$  pode ser diferente para cada tipo de restrição, de forma que a violação de qualquer restrição resulte em um aumento do valor nominal da função custo.

### 2.5.11. Codificação Real

Até então tratamos de AG em codificação binária. Entretanto, este tipo de codificação pode gerar *strings* muito grandes na representação de problemas multivariáveis, sendo que a aplicação dos operadores genéticos *bit a bit* demandará um elevado tempo computacional. Para minimizar este problema pode-se trabalhar diretamente com operadores para números reais, como já fora citado anteriormente neste texto. Além do menor comprimento de *strings*, a codificação real apresenta outras vantagens em relação à codificação binária:

- Na codificação real não há necessidade de conversões de *strings* para avaliação da função objetivo pois cada gene corresponde a uma variável. Em codificação binária, vários genes (*bits*) são utilizados para representar uma única variável;

- O limite de precisão da solução obtida em codificação real é o da precisão da máquina. Em codificação binária este limite é baseado no número de *bits* utilizados na representação das variáveis;

- A utilização de codificação real permite um maior controle em relação à ação dos operadores genéticos nas *strings*, pois cada gene representa uma variável. No caso da codificação binária, a aplicação dos operadores genéticos nas *strings* produz modificações nos fenótipos que são difíceis de serem previstas.

É importante observar que os operadores a serem alterados, em relação à codificação binária, são aqueles que trabalham ao nível das *strings*, ou seja, a mutação e o *crossover*. O escalonamento e a seleção operam diretamente no *fitness* dos indivíduos e, portanto, serão os mesmos não importando o tipo de codificação.

## 2.6.Redes Neurais Artificiais

No presente trabalho procurou-se estudar a aplicabilidade de uma metodologia empírica de representação de um processo de controle de temperatura segundo uma rede neural artificial *feedforward* conforme descrito no trabalho de CANCELIER(2004).

Procura-se estudar a aplicação de um controlador preditivo baseado em modelo (MPC) representado por um modelo sustentado por uma rede neural artificial. Esta aplicação deve-se ao fato de que o sistema em questão pode apresentar não-linearidades difíceis de serem representadas por metodologias de modelagem fenomenológica ou outras metodologias tradicionais; tempo morto, e dinâmica variável, sendo que as redes neurais artificiais se apresentam como uma boa opção a fim de representarem este tipo de processos.

Este tipo de aplicação se deve ao fato de que as redes neurais artificiais podem aproximar satisfatoriamente funções complexas, sem necessidade de um conhecimento detalhado do processo em questão, necessitando apenas de um conjunto de dados significativo a fim de que a mesma possa ser “treinada”.

A seguir apresenta-se uma revisão a respeito da teoria de redes neurais artificiais a fim de situar o assunto abordado no trabalho.

### 2.6.1.Inspiração da Metodologia

A metodologia de redes neurais artificiais foi inspirada no funcionamento biológico do neurônios humano e no que se conhece a respeito dele.

O cérebro humano é considerado um processador baseado em carbono, sendo composto por aproximadamente 10 bilhões neurônios. Todas as funções e movimentos do organismo estão relacionados ao funcionamento destas pequenas células. Os neurônios estão conectados uns aos outros através de sinapses, e juntos formam uma grande rede, chamada *Rede Neural*. As sinapses transmitem estímulos através de diferentes concentrações de  $\text{Na}^+$  (Sódio) e  $\text{K}^+$  (Potássio), e o resultado disto pode ser estendido por todo o corpo humano. Esta grande rede proporciona uma fabulosa capacidade de processamento e armazenamento de informação.

O sistema nervoso é formado por um conjunto extremamente complexo de neurônios. Nos neurônios a comunicação é realizada através de impulsos, quando um



impulso é recebido, o neurônio o processa, e passado um limite de ação, dispara um segundo impulso que produz uma substância neurotransmissora o qual flui do corpo celular para o axônio (que por sua vez pode ou não estar conectado a um dendrito de outra célula). O neurônio que transmite o pulso pode controlar a frequência de pulsos aumentando ou diminuindo a polaridade na membrana pós sináptica. Os neurônios têm um papel essencial na determinação do funcionamento, comportamento e do raciocínio do ser humano. Ao contrário das **redes neurais artificiais**, **redes neurais naturais** não transmitem sinais negativos, sua ativação é medida pela frequência com que emite pulsos, frequência esta de pulsos contínuos e positivos. As redes naturais não são uniformes como as redes artificiais, e apresentam uniformidade apenas em alguns pontos do organismo. Seus pulsos não são síncronos ou assíncronos, devido ao fato de não serem contínuos, o que a difere de redes artificiais.

Os principais componentes dos neurônios são:

- Os dentritos, que tem por função, receber os estímulos transmitidos pelos outros neurônios;
- O corpo de neurônio, também chamado de *somma*, que é responsável por coletar e combinar informações vindas de outros neurônios;
- O axônio, que é constituído de uma fibra tubular que pode alcançar até alguns metros, e é responsável por transmitir os estímulos para outras células.

### **2.6.2. Neurocomputação**

Os modelos neurais, procuram aproximar o processamento dos computadores ao cérebro. As redes neurais possuem um grau de interconexão similar a estrutura dos cérebro e um computador convencional moderno a informação é transferida em tempos específicos dentro de um relacionamento com um sinal para sincronização.

A Tabela 4 abaixo, traça um comparativo entre o cérebro humano e o computador:

Tabela 4 - Comparação entre cérebro e computador

Parâmetro	Cérebro	Computador
Material	Orgânico	Metal e plástico
Velocidade	Milisegundos	Nanosegundos
Tipo de Processamento	Paralelo	Sequencial
Armazenamento	Adaptativo	Estático
Controle de Processos	Distribuído	Centralizado
Número de elementos processados	10 e 11 à 10 e 14	10 e 5 à 10 e 6
Ligações entre elementos processados	10.000	<10

O mesmo paralelo pode ser traçado comparando o computador com as redes neurais. Para tanto, a comparação não se dará com um computador específico encontrado no mercado, mas sim com o paradigma predominante nos computadores atuais. A Tabela 5 apresenta esta comparação.

Tabela 5– Comparativo entre computadores e Neurocomputadores

Computadores	Neurocomputadores
Executa programas	Aprende
Executa operações lógicas	Executa operações não lógicas, transformações, comparações
Depende do modelo ou do programador	Descobre as relações ou regras dos dados e exemplos
Testa uma hipótese por vez	Testa todas as possibilidades em paralelo

### 2.6.3.Motivação do uso da Redes Neurais Artificiais

As redes neurais artificiais podem fornecer soluções para problemas complexos; mas não trabalhando individualmente. Estas, precisam ser integradas em um contexto de aplicabilidade. O uso das redes neurais deve observar as suas capacidades de corresponder as necessidades. Para tanto devemos observar algumas características apresentadas.

1. Não-linearidade: Um neurônio artificial pode ser linear ou não-linear. Uma rede neural, constituída por conexões de neurônios não-lineares é ela mesma não-linear. Além disso, a não linearidade é de um tipo especial, no sentido de ela ser distribuída por toda a rede.

2. Mapeamento de entrada-saída: Apresenta-se para a rede um exemplo escolhido ao acaso do conjunto, e os pesos sinápticos (parâmetros livres que relacionam os neurônios entre si) da rede são modificados para minimizar a diferença entre a resposta desejada e a resposta real da rede, produzida pelo sinal de entrada, de acordo com um critério estatístico apropriado. O treinamento da rede é repetido para muitos exemplos do conjunto até que a rede alcance um estado estável onde não haja mais modificações significativas nos pesos sinápticos.

3. Adaptabilidade: As redes neurais têm capacidade de adaptar seus pesos sinápticos a modificação do meio ambiente. Uma rede neural treinada para operar em um ambiente específico pode ser “retreinada” para lidar com modificações nas condições de operação. Além disso, quando está operando em um ambiente não-estacionário (ambientes com dinâmica variável no tempo), uma rede neural pode ser projetada para modificar seus pesos sinápticos em tempo real. A arquitetura natural de uma rede neural para classificação de padrões, processamento de sinais e aplicações de controle, aliada à capacidade de adaptação da rede, a torna uma ferramenta muito útil para classificação adaptativa de padrões, processamento adaptativo de sinais e controle adaptativo. É importante registrar que adaptabilidade não garante robustez a um sistema de controle por exemplo; na verdade pode resultar no contrário. Para aproveitar todos os benefícios da adaptabilidade, as constantes de tempo principais do sistema devem ser grandes o suficiente para que o sistema ignore perturbações espúrias, mas ainda assim serem suficientemente pequenas para responder a mudanças significativas no sistema.

4. Resposta a Evidências: No contexto de classificação de padrões, uma rede neural pode ser projetada para fornecer informação não somente sobre qual padrão particular selecionado, mas também sobre a confiança ou crença na decisão tomada. Esta última informação pode ser utilizada para rejeitar padrões ambíguos, caso eles estejam presentes, e com isso melhorar o desempenho de classificação da rede.

5. Informação contextual: O conhecimento é representado pela própria estrutura e estado de ativação de uma rede neural. Cada neurônio de rede é potencialmente afetado pela atividade de todos os outros neurônios da rede. Conseqüentemente, a informação contextual é tratada naturalmente pela rede neural.

6.Tolerância a falhas: Uma rede neural, implementada na forma física (em hardware), tem o potencial de ser inerentemente tolerante a falhas, ou capaz de realizar computação robusta, no sentido de que seu desempenho se degrada suavemente sob condições de operação adversas. Se um neurônio ou suas conexões é danificado, por exemplo, a recuperação de um padrão armazenado é prejudicada em qualidade. Contudo, devido à natureza distribuída da informação armazenada na rede, o dano deve ser extenso para que a resposta global da rede seja degradada seriamente. Assim, a princípio, uma rede neural artificial exibe uma degradação suave do desempenho em vez de apresentar uma falha catastrófica.

7.Implementação em VLS: A natureza paralela de uma rede neural artificial a faz a mesma ser potencialmente rápida na computação de certas tarefas. Esta mesma característica torna uma rede neural adequada para implementação utilizando tecnologia de integração em escala muito ampla. Uma virtude benéfica particular da tecnologia VLSI (very-large-scale-integration) é que ela fornece um meio de captar comportamentos realmente complexos de uma forma altamente hierárquica.

8. Uniformidade de análise e projeto: Basicamente, as redes neurais desfrutam de universalidade como processadores de informação. Dizemos isso no sentido de que a mesma notação é utilizada em todos os domínios envolvendo a aplicação de redes neurais. Esta característica se manifesta de diferentes modos:

1) Os neurônios de uma forma ou de outra representam um ingrediente comum a todas as redes neurais.

2) Esta uniformidade torna possível compartilhar teorias e algoritmos de aprendizagem em aplicações diferentes de redes neurais.

3) Redes modulares podem ser construídas através de uma integração homogênea de módulos.

9. Analogia Neurobiológica: O projeto de uma rede neural é motivado pela analogia com o cérebro, que é uma prova viva de que o processamento paralelo tolerante a falhas é não somente possível fisicamente, mas também rápido e poderoso. Os neurobiológicos olham para as redes neurais (artificiais) como uma ferramenta de pesquisa para a interpretação de fenômenos neurobiológicos. Por outro lado, os engenheiros olham para a neurobiologia procurando novas idéias para resolver problemas mais complexos do que aqueles baseados em técnicas convencionais de projeto por conexões fixas.

## 2.6.4. Entendendo o funcionamento das redes neurais Artificiais

As redes neurais artificiais consistem em um método de solucionar problemas de inteligência artificial, construindo um sistema que tenha circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. São mais que isso, são técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento, enquanto que o cérebro de um mamífero pode ter muitos bilhões de neurônios.

Sinais de entrada para uma rede neural a partir de fora da rede chegam através de conexões que se originam do mundo externo, saídas da rede para o mundo externo são conexões que deixam a rede.

De forma geral, a operação de uma célula da rede se resume em:

- Sinais são apresentados à entrada;
- Cada sinal é multiplicado por um peso que indica sua influência na saída da unidade;
- É feita a soma ponderada dos sinais que produz um nível de atividade;
- Se este nível excede um limite (threshold) a unidade produz uma saída;

### O Neurônio Artificial e a Rede Neural

Assim como o sistema nervoso é composto por bilhões de células nervosas, a rede neural artificial também seria formada por unidades que nada mais são que pequenos módulos que simulam o funcionamento de um neurônio. Estes módulos devem funcionar de acordo com os elementos em que foram inspirados, recebendo e retransmitindo informações.

O fisiologista Warrem MacCulloch interpretou o funcionamento do neurônio biológico como sendo um circuito de entradas binárias combinadas por uma soma ponderada (com pesos) produzindo uma entrada efetiva, a Figura 20 ilustra esta topologia.

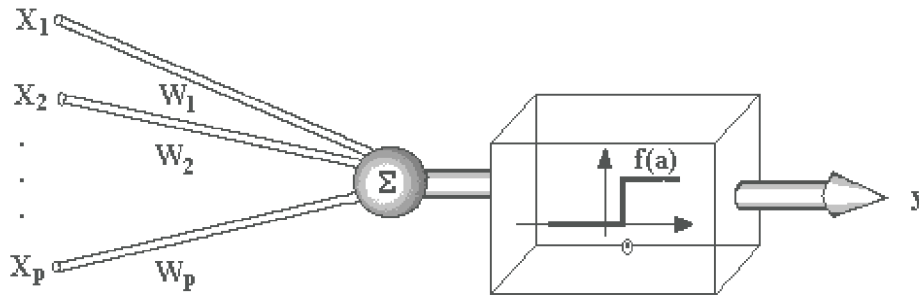


Figura 20 – Topologia de um neurônio artificial de Warren McCulloch

No modelo geral de neurônio (Figura 20) as entradas  $W_i X_i$  são combinadas usando uma função  $f$ , para produzir um estado de ativação do neurônio (correspondente à frequência de descarga do neurônio biológico). As entradas chegam através dos dendritos e tem um **peso** atribuído pela sinapse.

A função básica de um neurônio é somar as entradas e retornar uma saída, caso esse valor seja maior que o valor de soma (threshold).

O diagrama de blocos da Figura 21 mostra o modelo de um neurônio, que forma a base para o projeto de redes neurais (artificiais).

A função de ativação normalmente restringe (ou limita) o intervalo de amplitude do sinal de saída a um valor finito. Normalmente o intervalo normalizado utilizado é  $[0,1]$  ou  $[-1,1]$ .

Neste trabalho o intervalo utilizado foi de  $[-0.9,0.9]$ .

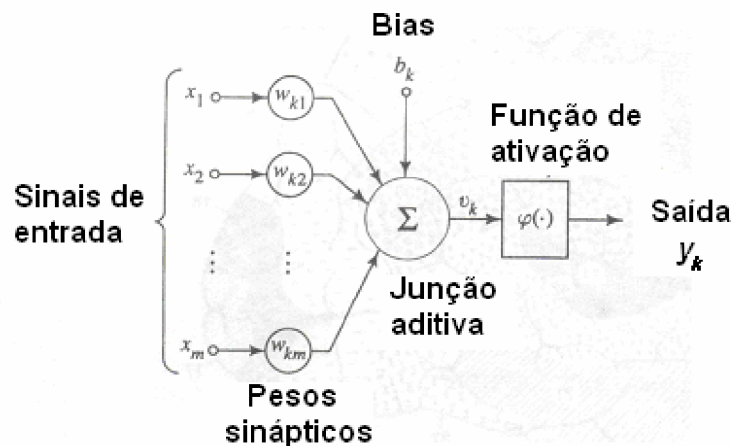


Figura 21 – Modelo de um neurônio artificial

O modelo neuronal da Figura 21 inclui também um bias aplicado externamente, representado por  $b_k$ . O bias  $b_k$  tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ele é positivo ou negativo, respectivamente.

Em termos matemáticos, podemos descrever um neurônio  $k$  escrevendo o seguinte par de equações:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad \text{Eq (39).}$$

$$y_k = \varphi(u_k + b_k) \quad \text{Eq.(40).}$$

Onde:  $x_1, x_2, \dots, x_m$  são os sinais de entrada;  $w_{k1}, w_{k2}, \dots, w_{km}$  são os pesos sinápticos do neurônio  $k$ ;  $u_k$  é a saída do combinador linear devido aos sinais de entrada;  $b_k$  é o bias;  $\varphi$  é a função de ativação; e  $y_k$  é o sinal de saída do neurônio. O uso de bias  $b_k$  tem o efeito de aplicar uma transformação afim à saída  $u_k$  do computador linear no modelo da Figura 21 como mostrado pela Equação 41.

$$v_k = u_k + b_k \quad \text{Eq.(41).}$$

Em particular, dependendo se o bias  $b_k$  é positivo ou negativo, a relação entre o campo local induzido ou potencial de ativação,  $v_k$ , do neurônio  $k$  e a saída do combinador linear  $u_k$  é modificada na forma ilustrada da Figura 22. Note que como resultado desta transformação afim, o gráfico de  $v_k$  em função de  $u_k$  não passa mais pela origem.

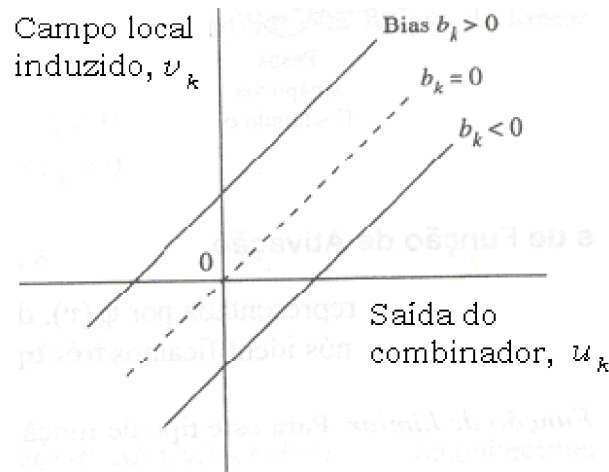


Figura 22 –  $v_k$  versus  $u_k$ , em relação ao sinal das bias

O bias  $b_k$  são um parâmetro externo do neurônio artificial  $k$ . Podemos considerar a sua presença como na Equação (40). Equivalentemente, podemos formular a combinação das Equações (39) até (41) como segue:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad \text{Eq.(42).}$$

$$y_k = \Phi(v_k) \quad \text{Eq.(43).}$$

Na equação (42) adiciona-se uma nova sinapse. A sua entrada é escrita pela equação (44), enquanto seu peso, pela equação (45).

$$x_0 = +1 \quad \text{Eq.(44).}$$

$$w_{k0} = b_k \quad \text{Eq.(45).}$$

Podemos, portanto, reformular o modelo do neurônio  $k$  como na **Figura 21**. Nesta figura, o efeito do bias é levado em conta de duas maneiras: **(1)** adicionando-se um novo sinal de entrada fixo em  $+1$  e **(2)** adicionando-se um novo peso sináptico igual ao bias  $b_k$ . Embora os modelos da Figuras 21 e 23 sejam diferentes na aparência, eles são matematicamente equivalentes.



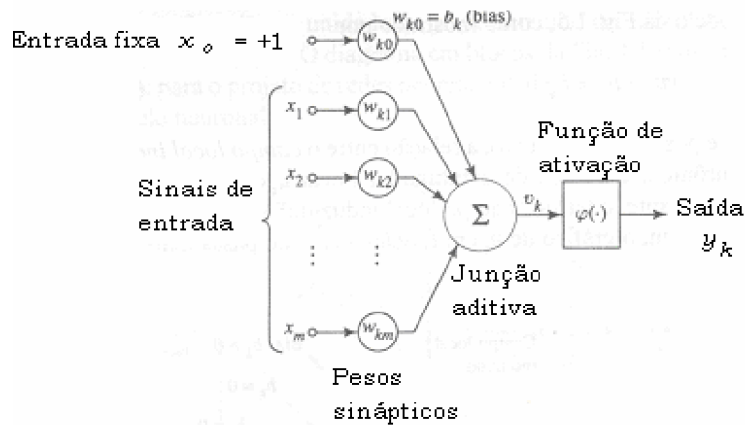


Figura 23– Modelo não-linear de um neurônio

### A arquitetura da rede neural artificial

Aqui pretende-se abordar conceitualmente e apresentar uma topologia a qual se assemelhe com a utilizada neste trabalho.

A arquitetura de uma rede neural artificial compreende basicamente, número de camadas de neurônios, a função de ativação utilizada e a disposição entre conexões dos neurônios em cada camada e entre as camadas.

A rede neural artificial é um sistema de neurônios ligados por conexões sinápticas e dividido em neurônios de entrada, que recebem estímulos do meio externo, neurônios internos ou hidden (ocultos) e neurônios de saída, que se comunicam com o exterior. A forma de arranjar perceptrons em camadas é denominado Multilayer Perceptron. O multilayer perceptron foi concebido para resolver problemas mais complexos, os quais não poderiam ser resolvidos pelo modelo de neurônio básico. Um único perceptron ou uma combinação das saídas de alguns perceptrons poderia realizar uma operação XOR, porém, seria incapaz de aprendê-la. Para isto são necessárias mais conexões, os quais só existem em uma rede de perceptrons dispostos em camadas. Os neurônios internos são de suma importância na rede neural pois provou-se que sem estes torna-se impossível a resolução de problemas linearmente não separáveis. Em outras palavras pode-se dizer que uma rede é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas

conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede.

A maioria dos modelos de redes neurais possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. Em outras palavras, elas aprendem através de exemplos. Arquiteturas neurais são tipicamente organizadas em camadas, com unidades que podem estar conectadas às unidades da camada posterior.

A rede neural passa por um processo de treinamento a partir dos casos reais conhecidos, adquirindo, a partir daí, a sistemática necessária para executar adequadamente o processo desejado dos dados fornecidos. Sendo assim, a rede neural é capaz de extrair regras básicas a partir de dados reais, diferindo da computação programada, onde é necessário um conjunto de regras rígidas pré-fixadas e algoritmos.

Usualmente as camadas são classificadas em três grupos:

- 1) Camada de Entrada: onde os padrões são apresentados à rede;
- 2) Camadas Intermediárias ou Ocultas: onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;
- 3) Camada de Saída: onde o resultado final é concluído e apresentado.

Redes neurais são também classificadas de acordo com a arquitetura em que foram implementadas, topologia, características de seus nós, regras de treinamento, e tipos de modelos.

No tocante a topologia, podemos observar alguns tipos gerais como segue:

#### *Redes Alimentadas Adiante com Camada Única*

Em uma rede neural em camadas, os neurônios estão organizados na forma de camadas. Na forma mais simples de uma rede em camadas, temos uma camada de entrada de nós de fonte (fornecem sinais de entrada) que se projeta sobre uma camada de saída de neurônios (nós computacional), mas não vice-versa. Em outras palavras, esta rede é estritamente do tipo alimentada adiante ou acíclica (em inglês, feedforward). Ela é ilustrada na **Figura 24** para o caso de quatro nós tanto na camada de entrada como na de saída. Esta rede é chamada de rede de camada única, sendo que a designação “camada única” se refere à camada de saída de nós computacionais (neurônios). Não contamos a camada de entrada de nós de fonte, porque lá não é realizada qualquer computação.

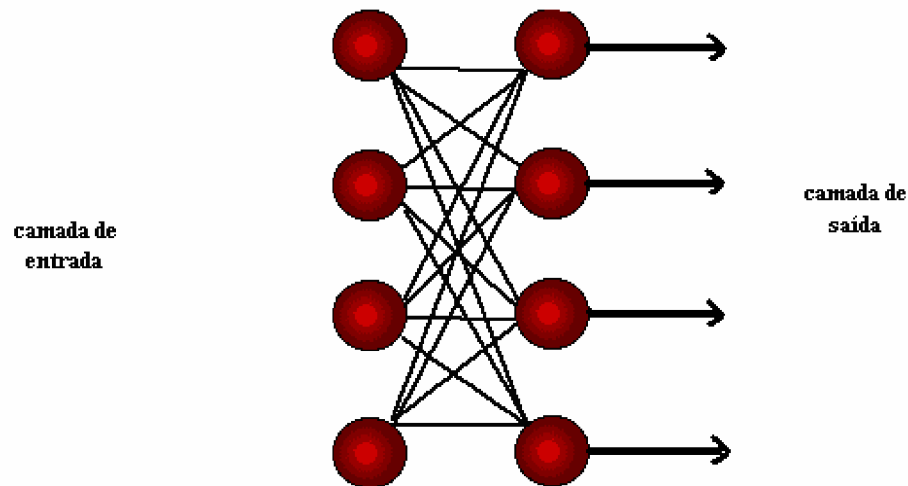


Figura 24 - Rede alimentada adiante ou acíclica (*feedforward*)

#### Redes Alimentadas Diretamente com Múltiplas Camadas

A segunda classe de uma rede neural alimentada adiante se distingue pela presença de uma ou mais camadas ocultas, cujos nós computacionais são chamados correspondentemente de neurônios ocultos ou unidades ocultas. A função dos neurônios ocultos é intervir entre a entrada externa e a saída da rede de maneira útil. Adicionando-se uma ou mais camadas ocultas, torna-se a rede capaz de extrair estatísticas de ordem elevada. Em um sentido bastante livre, a rede adquire uma perspectiva global apesar da sua conectividade local, devido ao conjunto extra de conexões sinápticas e da dimensão extra de interações neurais. A habilidade dos neurônios ocultos extraírem estatísticas de ordem elevada é particularmente valiosa quando o tamanho da camada de entrada

Os nós de fonte da camada de entrada da rede fornecem os respectivos elementos do padrão de ativação (vetor de entrada), que constituem os sinais de entrada aplicados aos neurônios (nós computacionais) na segunda camada (i.e., a primeira camada oculta). Os sinais de saída da segunda camada são utilizados como entradas para a terceira camada, e assim por diante para o resto da rede. Tipicamente, os neurônios em cada camada da rede têm como suas entradas apenas os sinais de saída da camada precedente. O conjunto de sinais de saída dos neurônios da camada de saída (final) da rede constitui a resposta global da rede para o padrão de ativação fornecido pelos nós de fonte da camada de entrada (primeira). A **Figura 25** mostra uma arquitetura de rede neural de múltiplas camadas.

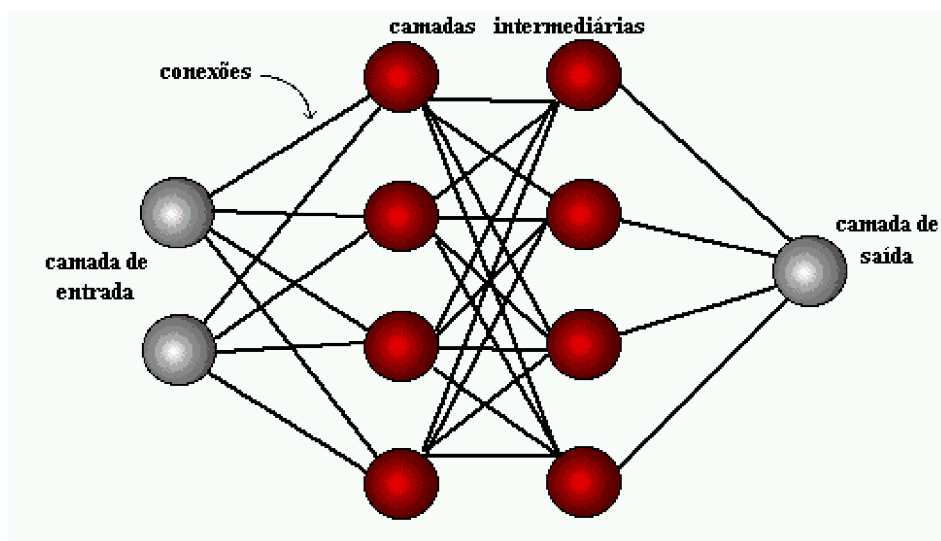


Figura 25 - Representação de uma Rede Neural Artificial com múltiplas camadas.

### Redes Recorrentes

Uma rede neural recorrente se distingue de uma rede neural alimentada adiante por ter pelo menos um laço de realimentação. Uma rede recorrente pode consistir, por exemplo, de uma única camada de neurônios com cada neurônio alimentando seu sinal de saída de volta para as entradas de todos os outros neurônios.

### **Topologia utilizada no trabalho**

A topologia de RNA utilizada neste trabalho é a mesma da Figura 26 uma rede neural *feedforward* (FANN - *Feedforward Artificial Neural Network*); só que com uma camada escondida somente.

O número de neurônios da camada escondida (intermediária) não é determinado segundo nenhuma técnica formal. HIECHT E NIELSEN, EM 1989 (FAUSETT, 1994), mostraram que uma FANN com três camadas pode aproximar qualquer função contínua com qualquer grau de precisão, desde que existam neurônios suficientes na camada intermediária. O número destes depende da qualidade que se deseja para a aproximação da rede sendo, na grande maioria dos casos, determinado de forma empírica. Este número não pode ser pequeno demais para haver uma aproximação pobre e nem grande demais, afim de gerar perda da generalidade da rede, fazendo com que até os ruídos sejam modelados. HECHT E NIELSEN (FAUSETT, 1994) chegaram a propor que uma

FANN representa bem uma função contínua, quando possui neurônios na camada intermediária, sendo  $n_e$  o número de neurônios da camada de entrada, contudo, esta metodologia não é referenciada para aplicações em identificação e/ou controle de processos.

Considerando especificamente a aplicação da FANN para a modelagem/identificação empírica de sistemas, o número de neurônios da camada de entrada da rede depende do número de variáveis relevantes para representar o processo. O número de neurônios da camada de saída depende do número de variáveis que se deseja avaliar, que correspondem, na maioria das vezes, nas saídas processo que devem ser determinadas. Para o processo que se deseja modelar, a função que representa a rede pode ser descrita como segue:  $y(k+1) = f[y(k), y(k-1); U_1(k-d_1), U_2(k-d_2)]$ , onde  $f$  é a função que considera todos os processamentos da rede.  $U_1(k-d_1)$  e  $U_2(k-d_2)$  são, respectivamente, as entradas 1 e 2 do processo, com  $d_1$  e  $d_2$  sendo o tempo morto em relação a cada variável de entrada.  $y(k)$  e  $y(k-1)$  são, respectivamente, as saídas do processo nos instantes  $k$  e  $k-1$ , de um sistema MISO (Multiple Input / Single Output). No sistema de polimerização estudado, a variável de saída de interesse é a temperatura de reação,  $y$ , sendo esta controlada pela manipulação das aberturas das válvulas de controle de vazão de vapor,  $U_1$ , e de controle de descarte de água quente da camisa e alimentação de água fria,  $U_2$ . Ainda, o sistema está sujeito às restrições descritas pelas Equações (46) e (47).

$$\begin{cases} 1\text{volt} \leq U_1 \leq 5\text{volts} \\ 1\text{volt} \leq U_2 \leq 5\text{volts} \end{cases} \quad \text{Eq.(46).}$$

$$\text{Temperatura M\u00ednima} \leq y \leq \text{Temperatura M\u00e1xima} \quad \text{Eq.(47).}$$

Quando a abertura das v\u00e1lvulas for selecionada em 1 volt, significa que esta encontra-se completamente fechada e 5 volts indica 100% de abertura para as v\u00e1lvulas de controle de vaz\u00e3o. As restri\u00e7\u00f5es na temperatura obedecem os limites f\u00edsicos do sistema. Considerando que este n\u00e3o \u00e9 dotado de sistema de refrigera\u00e7\u00e3o, a menor temperatura que se consegue \u00e9 o equil\u00edbrio com a temperatura ambiente, sendo a maior o ponto de ebuli\u00e7\u00e3o da \u00e1gua, meio cont\u00ednuo utilizado na polimeriza\u00e7\u00e3o em suspens\u00e3o do estireno.

Neste trabalho, uma rede neural feedforward completamente interconectada com três camadas foi empregada como modelo empírico do processo. A arquitetura da FANN utilizada é mostrada na Figura 26.

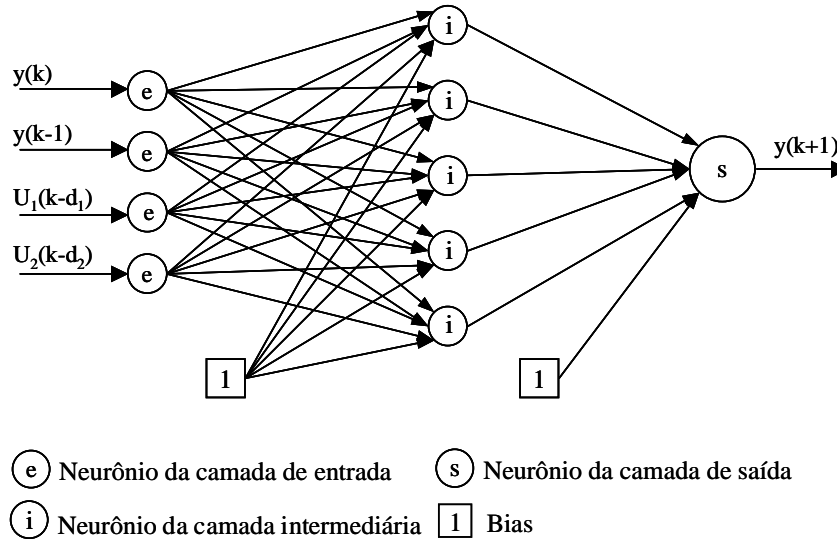


Figura 26 – Rede Neural utilizada no trabalho

O número de neurônios na camada intermediária pode variar, dependendo de para quantos neurônios se treinou a rede.

A ativação calculada para cada neurônio da camada intermediária fica definida conforme a Equação (48).

$$At_{2,i} = f(x_i) \text{ com } i = 1, \dots, \text{NNI} \quad \text{Eq. (48).}$$

Onde NNI é o número de neurônios da camada intermediária. A função de ativação utilizada para os neurônios da camada intermediária é a função tangente hiperbólica, dada pela Equação (49). Esta necessita que os dados de entrada sejam normalizados entre  $[-1,1]$ ; no caso deste trabalho; seguindo as orientações de CANCELIER(2004) os dados de entrada são normalizados entre  $[-0,9,0,9]$ .

$$f(x_i) = \tanh x_i = \frac{1 - e^{-2x_i}}{1 + e^{-2x_i}} \quad \text{Eq. (49).}$$

Temos que  $x_i$  é determinado pela Equação (50):

$$x_i = b_i + \sum_{e=1}^{NNE} W_{e,i} \times At_{1,e} \quad \text{Eq.(50).}$$

Onde NNE é o número de neurônios da camada de entrada.  $W_{e,i}$  corresponde ao peso do neurônio “i” da camada intermediária conectado ao neurônio “e” da primeira camada, assim como  $At_{1,e}$  a ativação de cada neurônio da primeira camada.  $b_i$  é o bias de cada neurônio “i” da camada intermediária.

Para a última camada da rede utilizou-se uma função linear. A Equação (51) apresenta a função utilizada para o neurônio da terceira camada.

$$At_{3,s} = b_s + \sum_{i=1}^{NNI} W_{i,s} \times At_{2,i} \quad \text{Eq.(51).}$$

Onde  $W_{i,s}$  corresponde ao peso do neurônio “s” da última camada conectado com cada neurônio “i” da camada intermediária,  $At_{2,i}$  a ativação de cada neurônio da segunda camada e  $b_s$  o bias de cada neurônio “s” da camada de saída.

### **2.6.5. Conjunto de dados e Treinamento de uma Rede Neural Artificial**

Na construção de um modelo neural, deve-se inicialmente analisar o problema a ser modelado, quais suas características mais importante, os fatos relevantes que podem influenciá-lo como também, as respostas que se querem obter como saídas da rede neural.

Os dados a serem fornecidos à rede, tanto na fase de aprendizado quanto durante a utilização do modelo, podem estar compreendidos entre limitados (quando normalizados, de preferência entre 0 e 1), dependendo da função de ativação escolhida, que deve ser compatível com estes intervalos. Isso acaba por tornar mais eficiente a etapa de treinamento da rede. Para adaptar o conjunto de dados a estes intervalos limitados, deve-se normalizá-los.). A normalização pode ser Linear ou não-linear; dependendo do tipo de problema a ser analisado e do efeito que a normalização pode ter sobre os dados (a normalização não linear pode introduzir uma estrutura para os dados completamente diferente da original, por exemplo).

Depois da normalização dos dados, estes são introduzidos à rede a serem obtidas as saídas, sendo que as mesmas estarão também em sua forma normalizada. Para poderem ser analisadas e interpretadas, as saídas deverão ser transformadas novamente para seu domínio original.

### ***Aprendizado***

As Redes Neurais Artificiais (RNAs) possuem a capacidade de aprender através exemplos e fazer interpolações e extrapolações do que aprenderam. No aprendizado se procura determinar a intensidade das conexões entre os neurônios. Um conjunto de procedimentos bem definidos para adaptar os parâmetros de uma RNA, afim de que a mesma possa aprender uma determinada função é chamado de algoritmo de aprendizado. Como era de se esperar, não há um único algoritmo de aprendizado. O que temos é um conjunto de ferramentas representadas por diversos algoritmos, cada qual com suas vantagens e desvantagens. Estes algoritmos, basicamente, diferem pela maneira através da qual o ajuste dos pesos é feito.

A utilização de uma RNA na solução de uma tarefa passa, inicialmente, por uma fase de aprendizagem, onde a rede extrai características relevantes de padrões de informações apresentados para a mesma, criando assim uma representação própria para o problema. A etapa de aprendizagem consiste em um processo iterativo de ajuste de parâmetros da rede, os pesos das conexões entre as unidades de processamento, que guardam o conhecimento que a rede adquiriu do ambiente em que está operando. Uma definição geral do que vem a ser aprendizagem pode ser expressa da seguinte forma.

*"Aprendizagem é o processo pelo qual os parâmetros de uma rede neural são ajustados através de uma forma continuada de estímulos fornecidos pelo ambiente no qual a rede está operando, sendo que o tipo específico de aprendizagem realizado, é definido pela maneira particular como ocorrem os ajustes realizados nos parâmetros".*

Diversos métodos para treinamento de redes foram desenvolvidos, podendo estes serem agrupados em dois métodos principais: Aprendizado Supervisionado e Aprendizado Não Supervisionado.



### Aprendizado Supervisionado

Este método de aprendizado é o mais comum no treinamento das RNAs, tanto de neurônios com pesos, quanto de neurônios sem pesos, sendo chamado Aprendizado Supervisionado porque a saída desejada que a rede deve resultar é fornecida por um supervisor (professor) externo. O objetivo é ajustar os parâmetros da rede, de forma a encontrar uma ligação entre os pares de entrada e saída fornecidos. A Figura 27 ilustra o mecanismo de Aprendizado Supervisionado.

O professor indica, explicitamente, um comportamento bom ou ruim para a rede, visando direcionar o processo de treinamento. A rede tem sua saída corrente (calculada) comparada com a saída desejada, recebendo informações do supervisor sobre o erro da resposta atual. A cada padrão de entrada submetido à rede compara-se a resposta desejada (que representa uma ação ótima a ser realizada pela rede) com a resposta calculada, e os pesos das conexões são ajustados para minimizar o erro. A minimização da diferença é incremental, já que pequenos ajustes são feitos nos pesos a cada etapa de treinamento, de tal forma que estes caminhem, se possível, para uma solução.

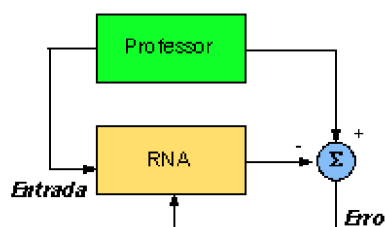


Figura 27 - Aprendizado Supervisionado

O exemplo mais conhecidos de algoritmos de Aprendizado Supervisionado para redes de múltiplas camadas, o algoritmo “*backpropagation*”.

### Aprendizado Não-Supervisionado

No Aprendizado Não-Supervisionado, como o próprio nome sugere, não há um professor ou supervisor para acompanhar o processo de aprendizado. Este método é ilustrado na Figura 28. Para este tipo de aprendizado, somente os padrões de entrada estão disponíveis para rede, ao contrário do Aprendizado Supervisionado, cujo conjunto de treinamento possui pares de entrada e saída. A partir do momento em que a rede

estabelece uma harmonia com as regularidades estatísticas da entrada de dados, desenvolve-se nela uma habilidade de formar representações internas para codificar características da entrada e criar novas classes ou grupos automaticamente.

Este tipo de aprendizado, só se torna possível, quando existe redundância nos dados de entrada. Sem redundância seria impossível encontrar quaisquer padrões ou características dos dados de entrada.

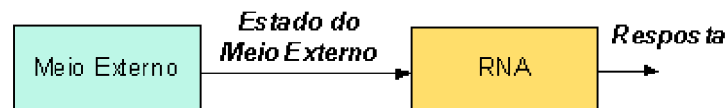


Figura 28 -Aprendizado não – supervisionado

A desvantagem está, intimamente, ligada ao conjunto de dados para treinamento, onde este deve ser redundante para que a rede consiga abstrair características em seu treinamento.

### Treinamento

Após a escolha do tipo e arquitetura da rede a ser utilizada na construção do modelo, definidos os parâmetros que servirão de entradas à rede e as saídas que se quer obter e coletados os dados referentes ao fenômeno a ser modelado, pode-se iniciar o processo de treinamento da rede neural.

O conjunto de treinamento (também chamado de In Sample) será composto de exemplos que serão utilizados na fase de treinamento da rede neural

O treinamento consiste na apresentação dos exemplos do conjunto de treinamento ao sistema. A rede processará os parâmetros de entrada relativos a estes exemplos através da multiplicação dos mesmos pelos pesos sinápticos e da posterior aplicação destes valores as funções de ativação dos neurônios, fornecendo então as respostas (saídas da rede) a este estímulo). Estas saídas da rede deverão ser comparadas como os valores reais dos parâmetros de saída correspondentes aos exemplos do conjunto de treinamento, e desta comparação será obtido um valor de erro da fase de treinamento. Procura-se então ajustar os valores dos pesos sinápticos, através de um algoritmo matemático, visando a diminuição do erro de treinamento. Contudo, deve-se fazer um controle deste ajuste dos pesos, pois além da rede estar sendo treinada, esta

não deverá perder a capacidade de generalização. Deste modo, paralelamente à fase de treinamento, deve-se apresentar à rede os exemplos do conjunto de teste, obtendo-se assim o chamado erro de teste, através da comparação das saídas da rede com as respostas desejadas correspondentes aos exemplos de teste. Monitorando-se os dois valores de erro (de treinamento e de teste), deve-se procurar chegar a um ponto ótimo, equivalente à minimização de ambos os erros, em que a rede estará bem treinada e com sua capacidade de generalização mantida.

Deste modo, o objetivo da fase de treinamento é ajustar da melhor maneira possível os valores dos pesos sinápticos da rede, procurando fazer com que as saídas fornecidas pela rede estejam bastante próximas dos valores reais de saída correspondentes a cada um dos exemplos apresentados, sem que se perca sua capacidade de generalização.

Existem diversos algoritmos para se treinar as redes neurais (Haykin, 1999), sendo os mais comumente utilizados o de error backpropagation (retro-propagação do erro) e o de Levenberg-Marquardt (também abreviado por LM), apresentaremos também a metodologia **utilizada neste trabalho** que se baseia em uma combinação de algoritmos genéticos com a metodologia de mínimos quadrados recursivos utilizada no trabalho de CLAUMANN(1999) e de CANCELIER (2004).

### **Algoritmo BackPropagation**

O treinamento por este algoritmo consiste basicamente no seguinte processo:

- a. as entradas dos exemplos do conjunto de treinamento são apresentadas à rede;
- b. estes dados são processados pela rede: em cada camada, através da multiplicação pelos pesos sinápticos, posterior somatório e aplicação da função de ativação, são fornecidas as saídas relacionadas à entradas apresentadas;
- c. as saídas fornecidas pela rede são comparadas com os valores reais obtidos dos experimentos e, desta variação, é calculado o erro;
- d. com o valor do erro, o ajuste para os pesos da última camada é calculado. O erro é então "retro-propagado" na rede, corrigindo os pesos sinápticos das camadas escondidas, visando um melhor ajuste da rede neural ao fenômeno a ser modelado.

Este processo iterativo deverá acontecer até que o erro seja aceitável, ou seja, que a rede tenha aprendido a tarefa a ser realizada.

### Algoritmo LM (LEVENBERG – MARQUARDT)

Após a popularização do algoritmo de *backpropagation* para o treinamento de redes, diversas pesquisas foram realizadas e métodos foram desenvolvidos com a finalidade de tornar sua convergência mais rápida. Assim, técnicas de otimização foram utilizadas para tanto. Um dos desenvolvimentos realizados neste sentido foi a incorporação do algoritmo de Marquardt para mínimos quadrados não lineares no algoritmo de *backpropagation*.

O método LM é um algoritmo tipo "quase Newton" anterior às redes neurais, que utiliza a informação das segundas derivadas em contraste com o gradiente descendente, que usa apenas a primeira. Por este motivo o algoritmo LM tende a ser mais eficiente que o gradiente descendente. Contudo, há a necessidade de se armazenar uma matriz **Hessiana**, que contém as segundas derivadas, o que resulta num uso significativamente maior da memória do computador, podendo comprometer sua eficiência no treinamento das redes muito grandes.

Este método foi incorporado à área de redes neurais para resolver um problema de otimização que aparece no treinamento das redes multi-camadas, podendo-se inclusive utilizar outros métodos de otimização para tanto.

### **3. Aplicações e ferramentas utilizadas**

#### **3.1. Algoritmo de CLAUMANN(1999) / CANCELIER(2004) para treinamento off-line da RNA - “Aplicação de um Algoritmo Genético em Codificação Real para o Treinamento da Rede Neural Feedforward”**

CALAUMANN(1999) desenvolveu um algoritmo de treinamento de redes neurais artificiais baseado em Inteligência artificial (GA) a fim de se determinar os pesos da camada escondida de uma RNA mutli-camadas. Posteriormente, CANCELIER(2004) utilizou o mesmo algoritmo em uma aplicação de controle de processo de polimerização. Este trabalho esta baseado nos trabalhos desenvolvidos pelos mesmos a fim de se aplicar a metodologia utilizada para controle de temperatura de um reator. Para tanto, vamos abordar a metodologia de treinamento off-line utilizada nos trabalhos citados. O algoritmo utilizado para treinamento da rede neural utilizada neste trabalho é um Algoritmo Genético com codificação real, baseado originalmente na ferramenta disponível no *Tolbox* do aplicativo Matlab® (*the Matworks, inc.*)(*Houck et al. 2003*).

O trabalho apresentado difere do de CANCELIER(2004) apenas quanto aos dados utilizados para treinamento da rede neural artificial, onde as variáveis manipuladas (válvula de aquecimento) e (válvula de resfriamento) , no trabalho de CANCELIER os dados padrões são gerados com a ação de uma sendo o “espelho” da outra (*Split-range*); para o caso deste trabalho, este ponto foi desconsiderado a fim de se avaliar mais amplamente a aplicabilidade da rede neural e do algoritmo de treinamento, já que na lei de controle utilizada neste trabalho que e a mesma de CANCELIER(2004) as ações de controle de resfriamento e aquecimento são desacopladas, ou melhor, independentes uma da outra.

Como os operadores e funcionamento dos algoritmos genéticos já foram introduzidos anteriormente, somente será exposto a metodologia global utilizada para o treinamento da rede com base nesta estratégia de inteligência artificial (GA).

Pelas vantagens descritas da codificação real em relação à binária, e motivado pela aplicação de GA's em codificação real na resolução de problemas com grande

dimensionalidade, como por exemplo o treinamento de redes neurais *feedforward*, optou-se por este tipo de codificação para o treinamento da FANN utilizada neste trabalho.

Um esquema do algoritmo genético em codificação real utilizado pode ser visto na Figura 29. Neste nota-se, em relação aos algoritmos em codificação binária, a ausência do procedimento de decodificação de *strings* e do procedimento para controle da extrapolação (se for utilizado o *crossover* intermediário) (CLAUMANN, 1999). Em relação aos operadores, devem ser alterados, em relação à codificação binária, aqueles que trabalham no nível das *strings*, ou seja, a mutação e o *crossover*. O escalonamento e a seleção operam diretamente no *fitness* dos indivíduos e, portanto, serão os mesmos não importando o tipo de codificação.

O *fitness* é uma medida relativa de desempenho entre os elementos da população e pode ser determinado diretamente a partir do valor da função objetivo ou de algum critério baseado nela.

No entanto, o *fitness* não pode, ou mesmo não deve em muitas situações, ser igualado a função objetivo.

Como exemplo, o *fitness* não poderá ser igualado a função objetivo quando essa última assumir valores negativos pois o *fitness* representa uma medida de adaptação, e posteriormente será convertido em probabilidade.

Os operadores que envolvem a metodologia já forma apresentados anteriormente.

CLAUMANN (1999) realizou um estudo sobre o desempenho de operadores genéticos para a otimização dos pesos de uma rede *feedforward* de modo que esta fosse capaz de representar um conjunto de *benchmarks*, normalmente encontrados na literatura especializada para este tipo de teste de eficiência. Os *benchmarks* de otimização em algoritmos genéticos são problemas com as seguintes características: unimodalidade, multimodalidade, descontinuidades, alta interação entre variáveis etc. A partir dos testes realizados, obteve uma configuração para os operadores do GA que apresentaram melhor desempenho. Esta configuração corresponde ao *crossover* intermediário e mutação exponencial. Considerando estas análises, estes operadores serão utilizados neste trabalho tanto para o **treinamento off-line da FANN, quanto em sua adaptação on-line**. Para completar, os demais operadores adotados foram o escalonamento tipo truncamento sigma, seleção pelo método da *rolleta* e o elitismo.

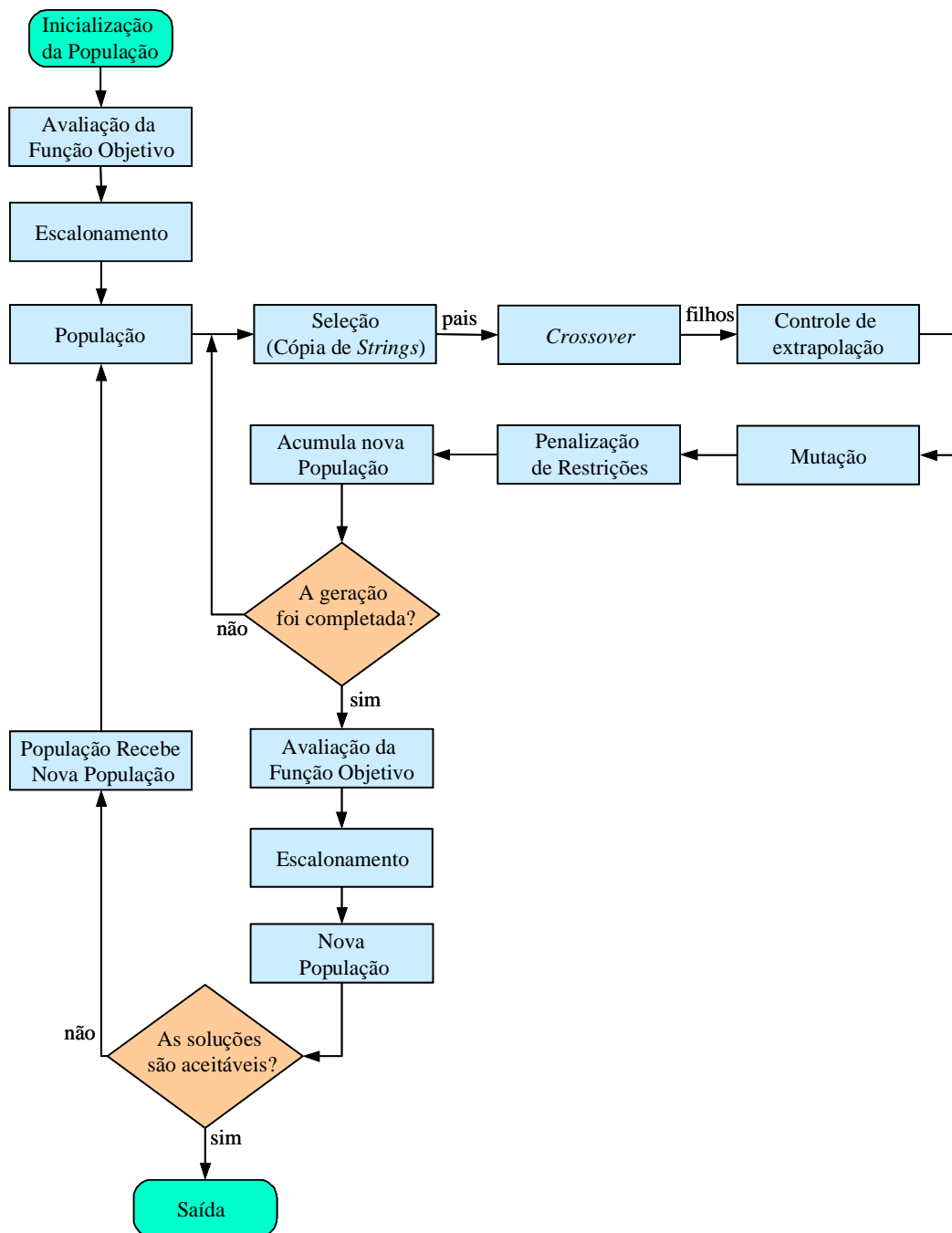


Figura 29 - Fluxograma do algoritmo genético em codificação real. (Claumann, 1999)

O procedimento a seguir é descrito com base nos trabalhos de CLAUANN(1999) e CANCELIER(2004) como já citado anteriormente, neste trabalho o algoritmo de treinamento e o controlador preditivo foram utilizados como **ferramentas** a fim de se explorar a potencialidade das metodologias em um controle de temperatura.

No operador *crossover* intermediário cada gene de um filho é gerado de acordo com a Equação (52).

$$\text{Filho}[i] = \text{Pai}_1[i] + \alpha (\text{Pai}_2[i] - \text{Pai}_1[i]) \quad \text{Eq. (52).}$$

Onde:

$i$  = posição na *string*, valor pode variar entre [1, comprimento da *string*];

Filho[ $i$ ] = Valor do  $i$ -ésimo gene para o filho gerado;

Pai\_1[ $i$ ], Pai\_2[ $i$ ] = Valor do  $i$ -ésimo gene no primeiro e segundo pais.

O valor  $\alpha$  pode ser escolhido aleatoriamente no intervalo  $[-d; 1+d]$ , sendo que, normalmente utiliza-se  $d = 0,25$ . Neste caso  $\alpha$  poderia variar entre  $[-0,25; 1,25]$ . Quando  $\alpha$  assume um valor entre 0 e 1 o *crossover* intermediário gera o filho através de uma interpolação. Para  $\alpha$  menor que zero ou maior que um atribui-se a cada gene do filho um valor que está fora do intervalo delimitado pelo gene dos pais que o geraram (extrapolação).

A mutação exponencial é uma variante da mutação por deslocamento. A diferença básica é a forma de como se calcula o valor a ser adicionado em cada gene. No caso da mutação exponencial, sorteia-se para cada gene um valor a partir de uma distribuição de probabilidade exponencial com média zero. Dessa forma, tem-se uma maior possibilidade de somar valores pequenos e uma menor possibilidade de somar valores elevados a cada gene que sofre a mutação. Deve-se utilizar uma taxa de mutação pequena para evitar pesquisa enumerativa.

O operador de escalonamento truncamento sigma utiliza-se da variância da população para transformar o *fitness*, de acordo com a Equação (53).

$$F = \max [f - (f_{med} - c\sigma), 0] \quad \text{Eq.(53).}$$

Onde:

$F$  = *fitness* escalonado;

$f$  = *fitness* natural;

$f_{med}$  = *fitness* médio da população;

$c$  = constante; normalmente entre 1 e 3;

$\sigma$  = variância da população.



O operador de seleção por *rolleta*, inicialmente, atribui para cada indivíduo da população uma probabilidade de seleção proporcional ao valor do seu *fitness*. Para transformar o *fitness* em probabilidade de seleção simplesmente calcula-se o somatório do *fitness* de todos os elementos e efetua-se uma normalização, conforme a Equação (54). Em seguida, escolhe-se aleatoriamente um indivíduo da população, sendo que, a chance de cada um depende de sua probabilidade de seleção. Como a *rolleta* só seleciona um indivíduo por vez, este operador deve ser aplicado um número de vezes igual ao número de indivíduos da população. Devido à natureza estocástica da *rolleta*, os elementos mais adaptados da população nem sempre serão selecionados, no entanto, é mais provável que o sejam.

$$P_i = \frac{F_i}{\sum_{j=1}^{Pop} F_j} \quad \text{Eq.(54).}$$

Onde:

$P_i$  = probabilidade de seleção do indivíduo  $i$ ;

$F_i$  e  $F_j$  = *fitness* escalonado do indivíduo  $i$  e de cada elemento da população, respectivamente;

$Pop$  = tamanho da população.

Definidos os operadores utilizados pelo GA, a próxima etapa compreende na codificação dos pesos da FANN no formato de um cromossomo e na definição de seus parâmetros. A codificação dos pesos da rede *feedforward* utilizada na forma de um cromossomo pode ser observada na Figura 30. Portanto, o número de parâmetros a serem determinados, que corresponderão ao número de genes no cromossomo, é idêntico ao número total de pesos mais o número de *bias* da FANN. Desta forma, o número de genes pode ser determinado pela Equação (55).

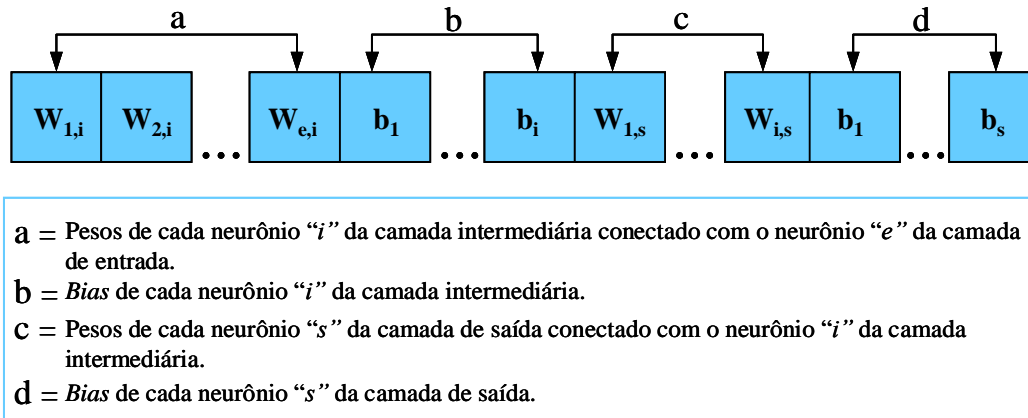


Figura 30 - Codificação da FANN na forma de um cromossomo. CANCELIER (2004)

$$N_{par} = NNI \left( NNE + NNS + \frac{NNS}{NNI} + 1 \right) \quad \text{Eq.(55).}$$

Onde  $N_{par}$  é o número de parâmetros para otimização (número de genes do cromossomo) e NNE, NNI e NNS o número de neurônios da camada de entrada, intermediária e saída da rede, respectivamente.

Neste trabalho, a abordagem para o treinamento da FANN utilizada por CANCELIER(2004) foi adotada. Nesta, o treinamento é realizado pela união do GA e do método de Mínimos quadrados, tendo-se, portanto, um algoritmo híbrido entre um método estocástico e um determinístico, respectivamente. **Neste caso, os pesos dos neurônios da camada intermediária, incluindo os *bias* destes, foram determinados utilizando-se o GA, enquanto que os pesos dos neurônios da camada de saída, incluindo o *bias*, foram, em seqüência, determinados pelo método de mínimos quadrados.** É importante lembrar que, neste trabalho, os neurônios da camada intermediária são não lineares, utilizam a função de ativação tangente hiperbólica, e os neurônios da camada de saída são lineares, permitindo assim o emprego do **método de mínimos quadrados lineares**. Desta forma, o cromossomo do GA é construído conforme mostra a Figura 31 e o número de parâmetros,  $N_{Par,1}$ , a ser otimizado por este método (número de genes do cromossomo) é dado pela Equação (56). O número de parâmetros a ser otimizado pelo método de mínimos quadrados,  $N_{Par,2}$ , é dado pela Equação (57). Contudo, o número total de parâmetros a serem estimados é dado pela soma de  $N_{Par,1}$  e  $N_{Par,2}$ .

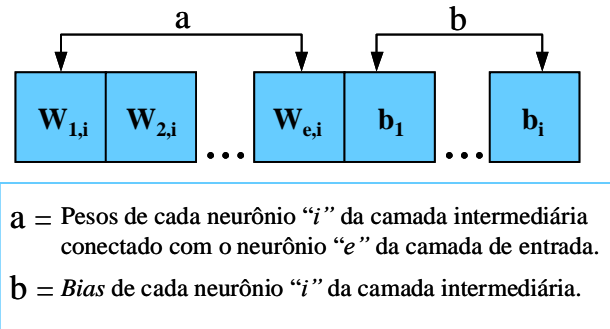


Figura 31 - Codificação dos pesos da camada intermediária da FANN na forma de um cromossomo – CANCELIER(2004)

$$N_{Par,1} = NNI(NNE + 1) \quad \text{Eq. (56).}$$

$$N_{Par,2} = NNS(NNI + 1) \quad \text{Eq. (57).}$$

Utiliza-se uma **configuração em série entre o GA e o método de mínimos quadrados para a composição do sistema híbrido**, para o treinamento da FANN, conforme mostra o fluxograma do algoritmo apresentado pela Figura 32. Considerando-se que a dimensão do problema para a estimação dos pesos pelo GA diminuiu e que, no passo seguido pela determinação dos pesos dos neurônios lineares pelo método de mínimos quadrados a solução encontra-se mais próxima do ótimo, “*espera-se que esta configuração apresente melhores resultados que a aplicação isolada do algoritmo genético*” CANCELIER(2004) descrita anteriormente. Há de se considerar também que o método de mínimos quadrados lineares leva a uma melhor solução possível para a otimização.

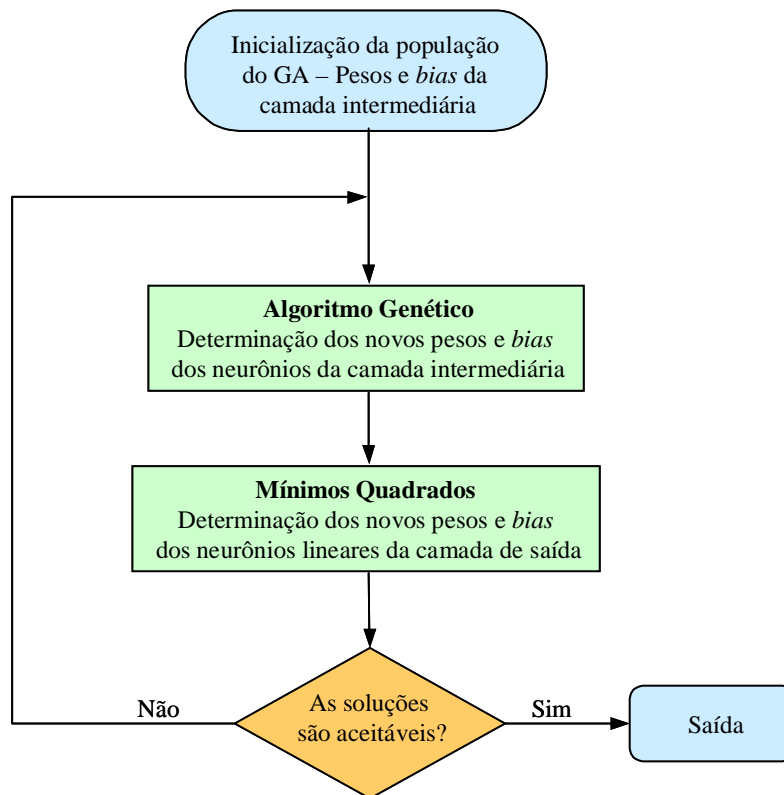


Figura 32 - Fluxograma do algoritmo para o sistema híbrido utilizado no treinamento da FANN – CANCELIER(2004)

### 3.2. Adaptação on-line dos pesos de uma Rede Neural Artificial

Como já citado anteriormente, a metodologia de algoritmos genéticos foi utilizada também em um sistema de controle de temperatura a fim de adaptar os pesos da rede neural em um sistema on-line, aplicando a ferramenta utilizada no trabalho de CANCELIER(2004); “...modificações na dinâmica do processo irão ocorrer...”.

Em razão de que é esperado modificação na dinâmica do processo; o qual já teria sido; após treinamento off-line; representado por um modelo empírico de uma RNA, uma ferramenta a qual possa perceber e atualizar os pesos da RNA segundo mudanças na dinâmica é muito interessante. Esta aplicação foi investigada neste trabalho e uma breve descrição do mecanismo utilizado na ferramenta é apresentado a seguir.

“Considerando que a rede é utilizada como modelo de previsão para o controlador preditivo, espera-se que ela tenha melhor desempenho se treinada de forma recorrente, já que é desta forma sua utilização no algoritmo de controle”(CANCELIER(2004)).

A Figura 33 apresenta a forma de representação da RNA dos padrões recorrentes de treinamento, utilizando adaptação on-line dos pesos.

No caso, a saída da rede é utilizada como padrão de entrada, ou seja, não são utilizados os dados reais de saída do processo como entrada da rede, o que caracteriza uma representação recorrente da mesma. **J** na figura representa o tamanho da janela de pontos (número de padrões) utilizada para o treinamento, obtidos por um histórico de valores do processo em andamento. Sendo que no caso, dado um valor de **J**, em um instante  $k$ , os  $(J + \max(d_1, d_2))$  pontos passados foram armazenados para o treinamento.

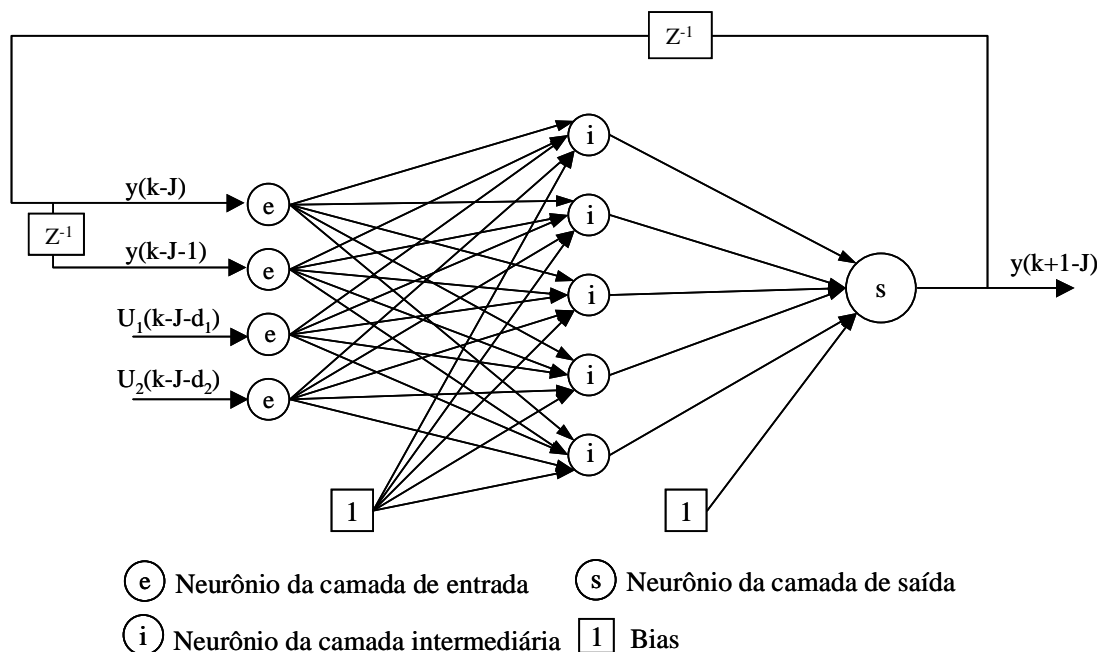


Figura 33 - FANN com a representação utilizada no treinamento recorrente  
CANCELIER(2004)

A adaptação dos pesos se faz necessária em função de diferenças dos pesos da rede treinada em relação ao processo real que ocorrem devido a mudanças na dinâmica do mesmo. Podemos concluir então que não há a necessidade de se atualizar todos os pesos da RNA, então os pesos do treinamento off-line podem ser utilizados como ponto de partida para adaptação.

No caso da ferramenta utilizada, a adaptação é realizada nos pesos e bias dos neurônios da camada de saída da RNA. Com isto o **cromossomo do GA** é construído conforme mostra a Figura 34. O número de parâmetros a ser utilizado (número de genes) é dado pela Equação 58. O número de parâmetros a serem otimizados (número de genes do cromossomo) é dado pela Equação (31). Onde  $NNI$  é o número de neurônios na camada escondida (intermediária) e  $NNS$  é o número de neurônios na camada de saída da RNA. Dependendo da topologia utilizada na RNA o processamento computacional pode ser maior ou menor.

$$N_{par} = NNS(NNI + 1) \quad \text{Eq.(58).}$$

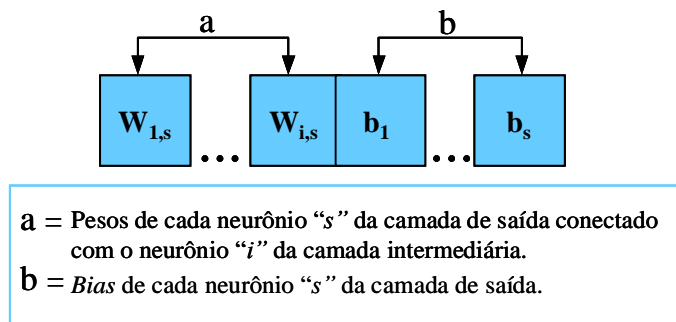


Figura 34 - Codificação dos pesos da camada de saída da RNA em um cromossomo

A fim de que a adaptação seja satisfatória, o tempo de amostragem do sistema de controle deve ser maior que o tempo levado para se realizar o cálculo do GA. O algoritmo inicia com os parâmetros do treinamento off-line e logo após vai sendo alimentado pelos pesos e *bias* utilizados no instante passado.

Um algoritmo de supervisão é utilizado a fim de garantir que os parâmetros atualizados pela metodologia de GA sejam realmente melhores que os utilizados no intervalo anterior ou que os utilizados pelo treinamento off-line.

O critério de otimização (ou minimização) do algoritmo genético para determinar quais os parâmetros utilizar é o do erro médio quadrático, representado pela Equação 59. Neste ponto o algoritmo de supervisão possui três elementos a serem considerados, ditos Rede1 (os parâmetros da rede treinada de modo off-line), Rede2 (RNA que recebe a cada intervalo de amostragem novos pesos e bias do neurônio da camada de saída, conforme adaptação por GA), Rede3 (a cada intervalo de amostragem recebe os melhores parâmetros entre as RNA concorrentes).

Em resumo, Rede1 não sofre adaptação, Rede2 é constantemente adaptada e Rede3 é a melhor rede utilizada em um momento passado.

$$SEMQ = \sum_{i=1}^J (y_{real,i} - y_{pred,i})^2 \quad \text{Eq. (59).}$$

Desta forma “o algoritmo de otimização,.....tem como principal objetivo a adaptação dos parâmetros da rede de forma a garantir robustez do modelo em conjunto com o controlador preditivo”-CANCELIER(2004).

### **3.3.O controlador utilizado**

O controlador utilizado é do tipo baseado em modelo, preditivo, utilizando como modelo do processo o modelo empírico de um Rede Neural Artificial-RNA.

O controlador têm sua lei de controle formulada de maneira que as ações de controle, válvula de aquecimento e resfriamento sejam independentes uma da outra, ou seja, temos duas variáveis manipuladas neste processo.

Também foi utilizados controlador linear PID ajustado segundo metodologias de sintonia clássicas, a partir de metodologias de identificação já citadas anteriormente.

## 4. Material e Métodos

Neste ponto é apresentada uma descrição das instalações utilizadas nos experimentos de ajuste, e teste dos controladores aplicados. Também é apresentada uma descrição da reação química conduzida nos experimentos.

Todos os testes foram conduzidos em um reator de polimerização de aço conforme descrito a seguir. O sistema se assemelha muito ao utilizado por *CANCELIER* (2004) e *MAZZUCCO* (2003), com algumas diferenças já citadas anteriormente.

### 4.1. Reator de Aço

Esta unidade de reação é constituída, basicamente, de um reator encamisado em aço inox 316L, marca Suprilab Ltda, com capacidade de 5 litros e pressão máxima de operação de  $15 \text{ kg}_f/\text{cm}^2$ , provido de um sistema de agitação com acionamento por *software*, com um motor trifásico marca Eberle com saída reduzida em 1:5, um controlador de frequência da marca Siemens e um agitador do tipo turbina dupla, centrada, que se estende até a base do reator. As trocas térmicas necessárias ao reator são realizadas por um trocador de calor a placas com correntes cruzadas, marca Alfa Laval, uma bomba centrífuga de  $\frac{1}{4}$  cv, marca Schneider, modelo 02.01, além de duas válvulas pneumáticas do tipo igual porcentagem ar-abre/falha-fecha, da marca Badge Meter Inc, que são responsáveis pela dosagem das correntes quente e fria. Também fazem parte do sistema um termopar do tipo J da marca Ecil em conjunto com um amplificador/transmissor, além de um reservatório de nitrogênio gasoso o qual não foi utilizado.

A Figura 35 mostra o esquema da planta piloto utilizada para as reações de polimerização em suspensão do estireno e estudo/aplicação da estratégia de controle desenvolvida. Maiores detalhes sobre o reator podem ser observados na Figura 36.



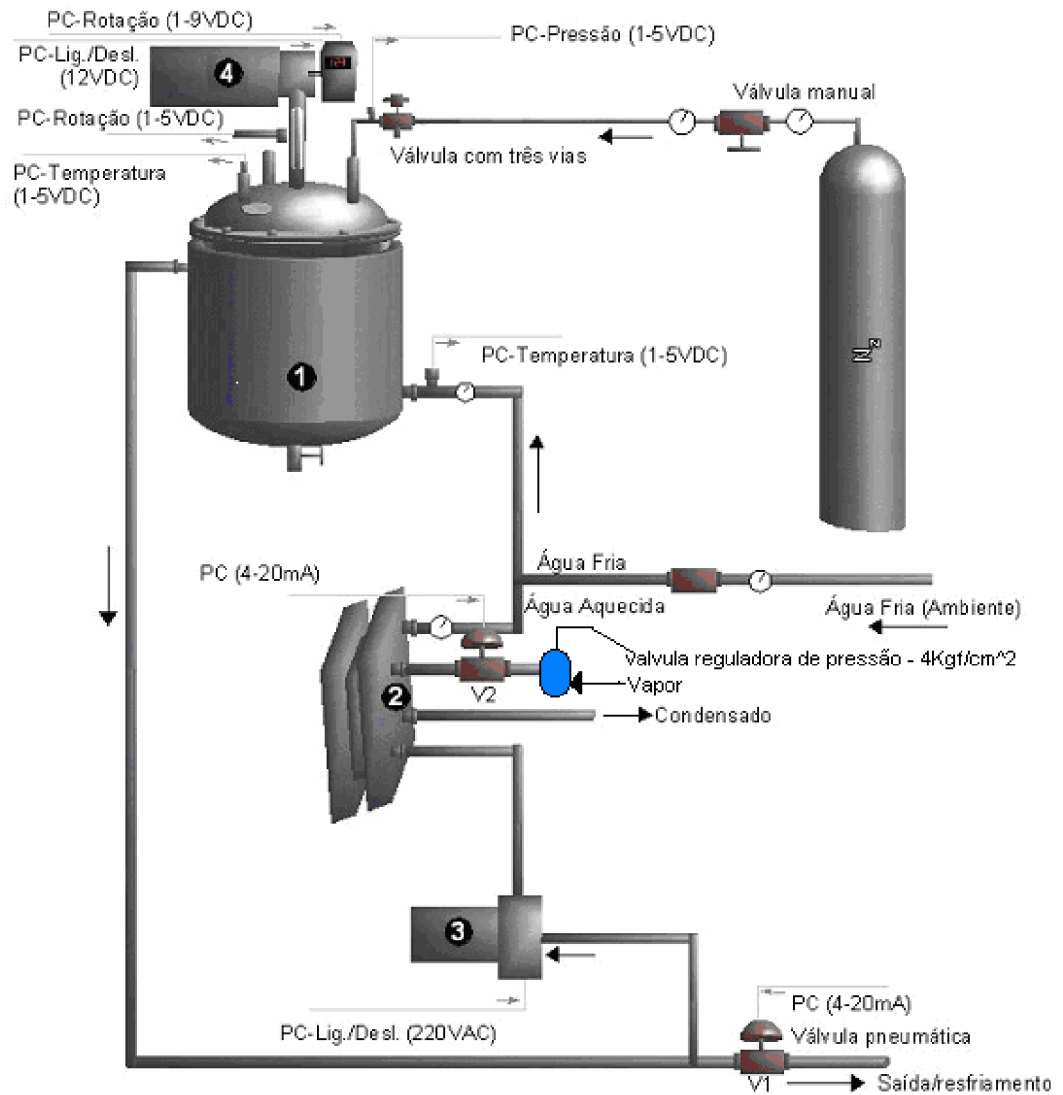


Figura 35 - Esquema da unidade piloto de polimerização (Mazzucco, 2003) “com pequena alteração a respeito do trabalho de Mazzucco quanto a válvula reguladora de pressão de vapor”.

- 1 – Reator de aço encamisado.
- 2 – Trocador de calor a placas.
- 3 – Bomba centrífuga.
- 4 – Motor para agitação e controlador de frequência.

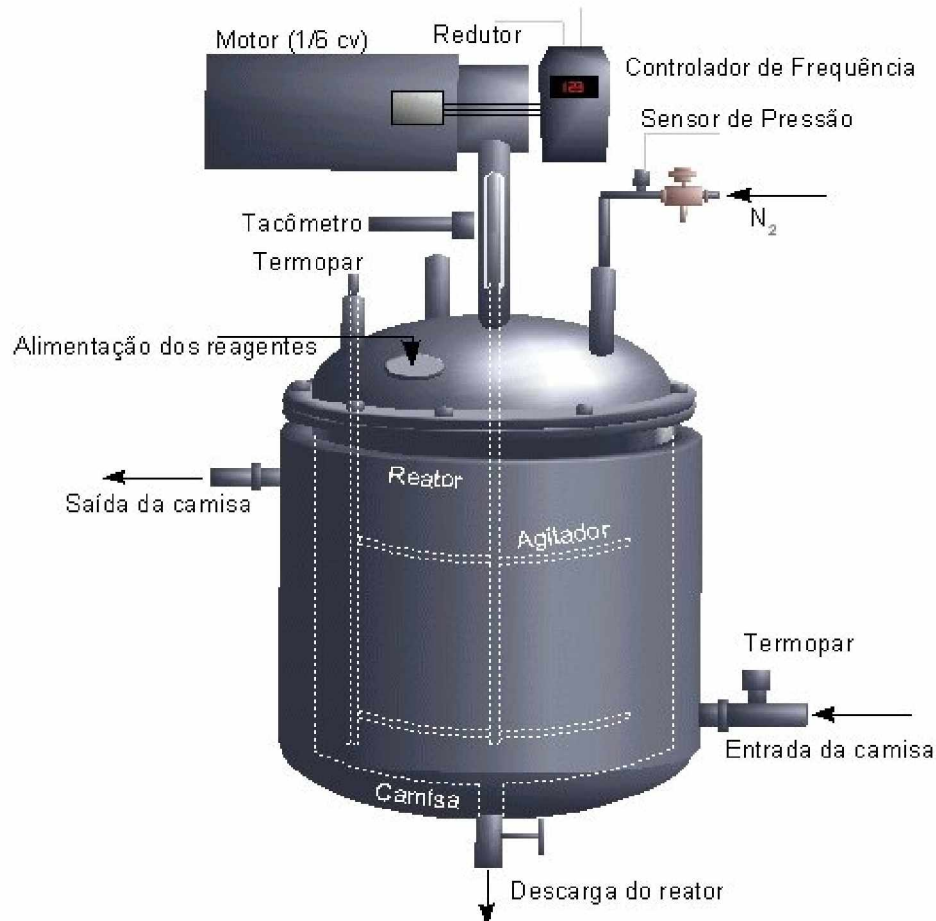


Figura 36 - Detalhes do reator de polimerização (Mazzucco, 2003).

A frequência de agitação, que pode variar de 0-700 rpm, é manipulada remotamente por um sinal de 1-10 VDC. O controlador de frequência requer um sinal de 12VDC para que possa ser iniciada a manipulação remota.

O sistema de troca térmica é iniciado pelo acionamento da bomba centrífuga, que faz com que a pressão na camisa do reator seja de aproximadamente  $2 \text{ kg/cm}^2$ . A válvula V2 é responsável pela circulação de vapor no trocador de calor a placas, através do qual é gerada a corrente quente para o aquecimento do reator. Com a válvula V1 mantida fechada, apenas a corrente quente circula no reator, recebendo a cada passagem pelo trocador de calor mais energia, podendo atingir, desta forma, elevadas temperaturas em um intervalo de tempo relativamente curto. A abertura da válvula V1 diminui a pressão na camisa do reator, bem como em toda linha de circulação, fazendo com que a corrente fria circule nesta. Como V1 e V2 são válvulas proporcionais, é possível obter a temperatura ótima para a reação.

O vapor é fornecido por uma caldeira com aquecimento elétrica, marca Engeterm, com capacidade de produção de 300 Kg/h de vapor e pressão de trabalho máxima de 15 Kgf/cm<sup>2</sup>. Devido ao seu sistema de controle, fornece vapor a uma pressão que pode variar de 6 a 8 Kgf/cm<sup>2</sup>, caracterizando uma fonte de perturbação ao sistema de polimerização, a mesma é amenizada por uma válvula reguladora de pressão na entrada de vapor para o sistema conforme mostra a Figura 35. A água, para refrigeração, é proveniente da própria rede hidráulica do laboratório, estando sempre a temperatura ambiente.

O acionamento das válvulas pneumáticas é feito através de conversores eletropneumáticos, marca Helix, com sinal de entrada de 4-20 mA e saída de 3-15 psi, a partir de uma alimentação contínua de 20 psi, obtida por um filtro regulador de pressão, marca Hélix - tipo F11-11, que suporta uma pressão de alimentação de até 250 psi.

Esta unidade possui um sistema remoto de controle de temperatura e frequência de agitação realizado através de microcomputador PC-Intel Pentium-S 200MHz, uma placa de aquisição de dados AD/DA marca Data Translation modelo DT2812 e *software* para processamento em tempo real desenvolvido no próprio laboratório. A Figura 37 mostra a tela principal do *software* desenvolvido para a aplicação das estratégias de controle e aquisição de dados.

O software foi desenvolvido em linguagem *Object-Pascal*, utilizando-se o ambiente integrado para desenvolvimento *Borland Delphi 6.0*<sup>®</sup>. Este ambiente dispõe de ferramentas de grande utilidade para agilizar a escrita/manutenção do código e rastreamento de erros de sintaxe e lógica, facilitando o projeto da interface com o usuário. A codificação dos módulos de aquisição de dados, algoritmos de controle e algoritmos de otimização foi feita utilizando-se a técnica de programação orientada a objetos, desta forma proporcionando maior compreensão do código e maior agilidade.

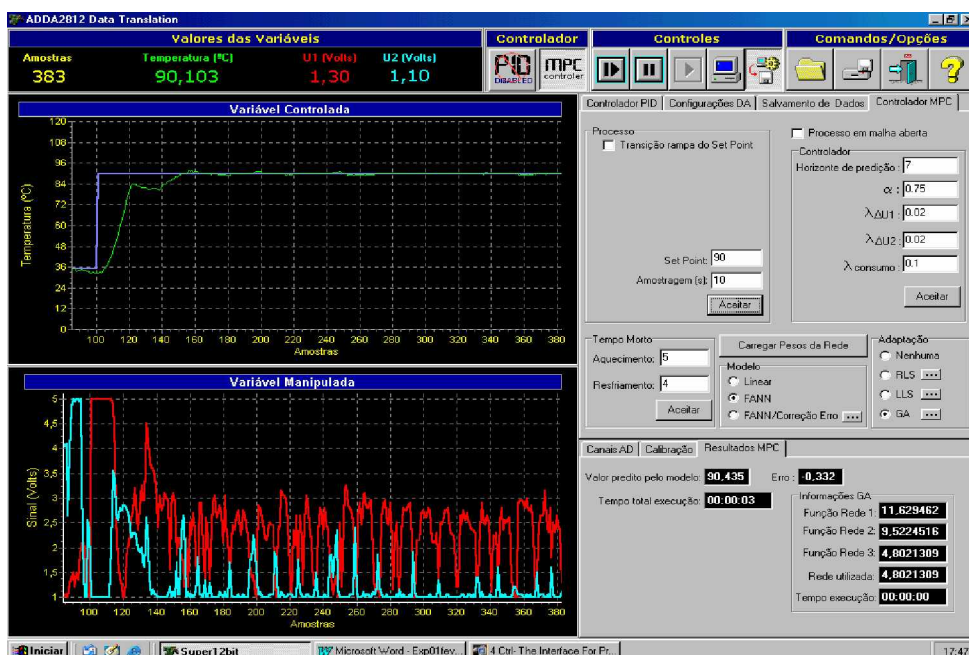


Figura 37 - Tela principal do software de aquisição de dados e controle.

## 4.2. Reação Química de Polimerização

A reação química realizada contém:

- 1,5 litros de estireno em 3,5 litros de água.
- *Holdup* de 0,3.

São adicionados:

- 118,125 ml de PVP.
- 4,624 g de BPPO com 2,524 g de TBP.

Inicia-se a reação química adicionando-se estireno e água, leva-se a temperatura à 90<sup>0</sup>C e após 1 hora e trinta minutos decorridos do momento em que se obteve 90<sup>0</sup>C se adiciona o PVP (anti-coagulante). Os sais são adicionados no início da reação, e suas quantidades podem variar dependendo do produto que se pretende obter. A velocidade de agitação é mantida em 500 rpm durante toda a reação de polimerização.

Esta formulação foi utilizada em todos os experimentos com reação química neste trabalho.

## 5. Resultados e Discussões

### 5.1. Identificação do Sistema

Neste ponto se apresenta os resultados obtidos para o estudo do sistema de controle de temperatura (sistema de polimerização) no tocante a identificação do sistema.

No presente estudo realizaram-se aplicações de procedimentos de identificação gráficos (síntese direta), procurando-se a partir da curva de resposta do sistema em malha aberta aproximar a função transferência do sistema para um sistema de primeira ordem com tempo morto, aproximar a função transferência do sistema a sistemas de segunda ordem sobreamortecidos (Método de Sundaresan), identificação do sistema por metodologia de mínimos quadrados.

Os resultados encontrados a partir da tentativa de aproximação da função transferência de segunda ordem sobreamortecida conforme o Método de Sundaresan não foram satisfatórios e não são apresentados aqui. Pode-se observar estes resultados no **Apêndice E**.

#### 5.1.1. Identificação através dos Mínimos Quadrados

**Metodologia:** Utilizando-se a ferramenta Tollbox do Matlab<sup>®</sup> a fim de se obter uma identificação mais confiável do sistema em função de ser um método mais robusto para este tipo de conjunto de dados, já que outros métodos para identificação de sistemas de segunda ordem não retornaram resultado satisfatório.

Tomou-se um outro conjunto de dados com pontos experimentais sendo obtidos através do sistema de aquecimento e resfriamento do reator, com as válvulas sendo variadas a fim de se obter pontos em torno de uma região de operação do processo a fim de que possamos identificar o sistema através do método dos mínimos quadrados.

A Figura 38 ilustra a curva de resposta obtida no experimento.

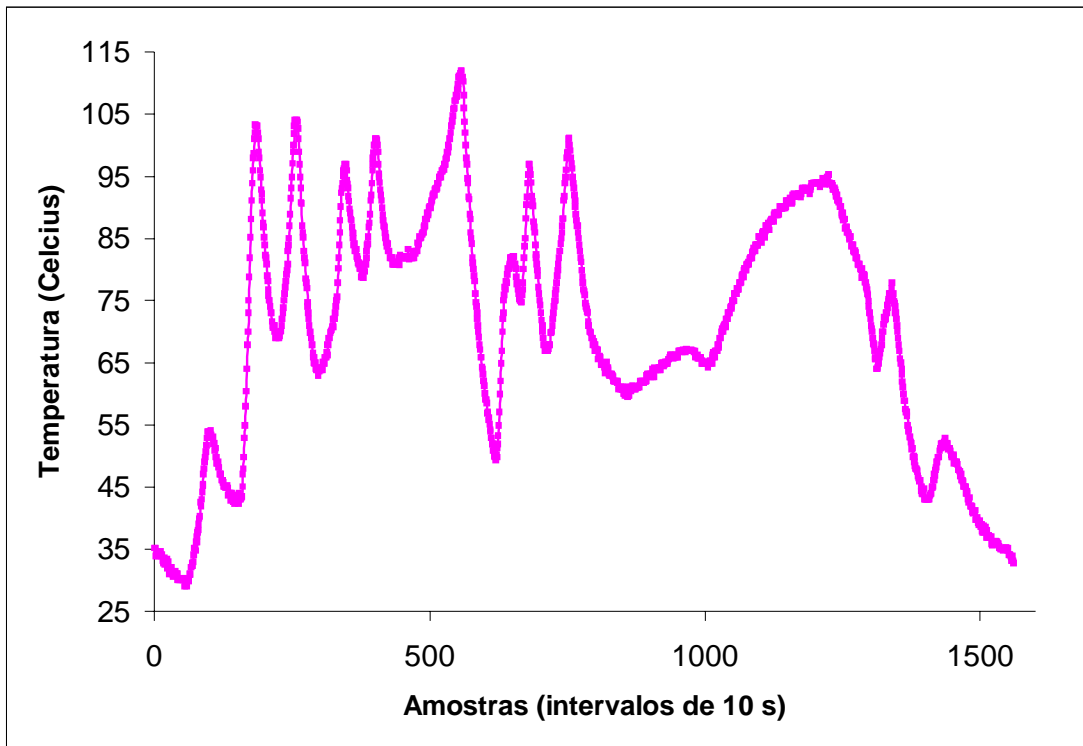


Figura 38– Respostas obtidas para identificação do processo (Mínimos Quadrados)

As variações das válvulas, tomadas nesta identificação podem ser observadas na Figura 39.

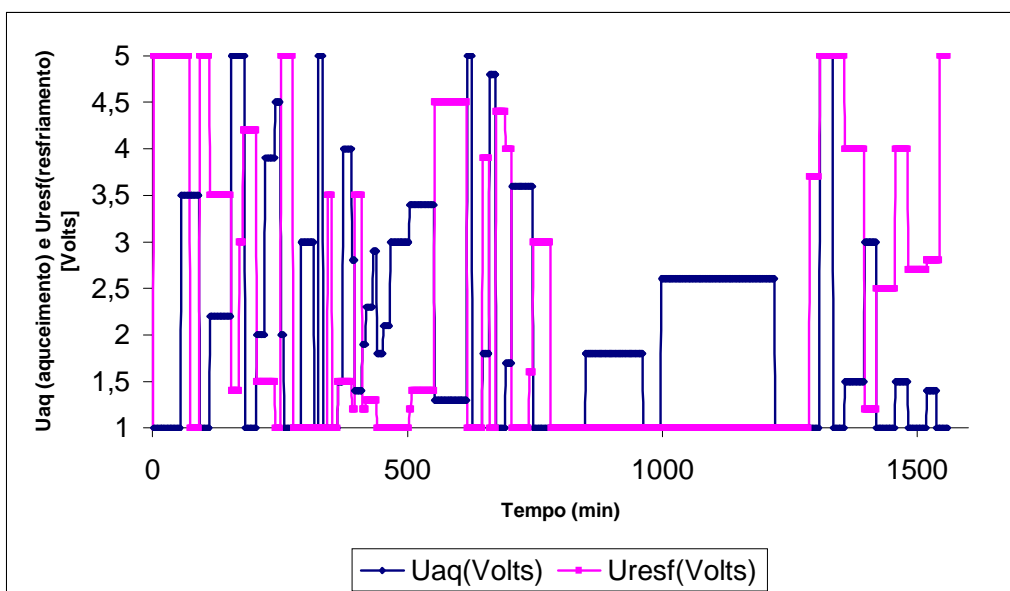


Figura 39 – Aberturas das válvulas para geração dos padrões de identificação

Estes mesmos pontos foram utilizados mais tarde para treinamento off-line da rede neural *Feedforward*.

Com estes pontos e tendo também os pontos referentes aos valores de abertura das válvulas de aquecimento e resfriamento; poderemos proceder o cálculo dos coeficientes pelo método mínimos quadrados como já citado. Os resultados encontrados são apresentados a seguir.

Para aplicação da Metodologia de mínimos quadrados, a matriz de regressores pode aproximar a função de transferência do processo frente a série temporal obtida segundo a válvula de aquecimento e de resfriamento. Para tanto vamos ter como esperado dinâmicas diferentes para os dois processos, então teremos duas funções de transferência, uma referente ao aquecimento e outra referente o resfriamento. Com isto podemos estimar os tempos mortos envolvidos nestes processos e compará-los com outras metodologias de síntese direta via método gráfico.

O resultado obtido vem a seguir para funções de segunda ordem com tempo morto. O método não foi aplicado para sistema de primeira ordem com tempo morto pois se utilizou outro método (gráfico) para fins de comparação.

Identificação dos dados quanto ao aquecimento, onde o arquivo de dados é ajustado frente as ações tomadas para a válvula de aquecimento.

A Figura 40 apresenta o resultado encontrado comparando o modelo obtido versus os pontos reais.

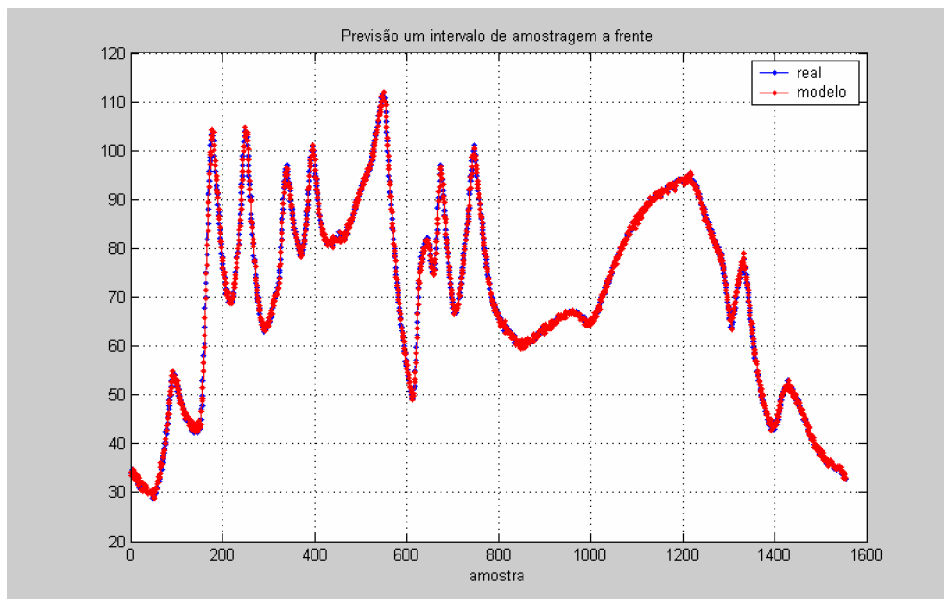


Figura 40 – Comparação entre o modelo obtido e os dados reais

A seguir os comandos utilizados para obtenção deste resultado em Matlab®.

```
[Y,U]=dadosR6aq;
[teta,TempoMorto,R]=idtctr2(Y,U);
*****
DETERMINACAO DA FUNÇÃO DE TRANSFERENCIA
*****
*****
MÉTODO DE ESTIMAÇÃO DOS PARÂMETROS DA FUNÇÃO DE
TRANSFERENCIA
digite 0: algoritmo do Erro no modelo (Mínimos quadrados)
digite 1: algoritmo do Erro na saída (Levenberg-Marquardt)
método = 0
*****
*****
DEFINIR PONTO DE OPERACAO (Y0, U0)
digite 0: nao
digite 1: sim
= 0
*****
*****
VALOR DO TEMPO MORTO
definir: digite o valor (em amostras)
determinar: digite [] (pesquisa ate 30 amostras)
Tempo morto = []
*****
*****
DEFINIR ORDEM DO MODELO
digite 1: primeira ordem
digite 2: segunda ordem
ordem = 2
*****
*****
RESPOSTA INVERSA
digite 0: sem resposta inversa
digite 1: com resposta inversa
resposta inversa = 0
*****
*****
A função transferência encontrada:
Transfer function:
0.3554 z^-6
-----
1 - 1.215 z^-1 + 0.2249 z^-2
Sampling time: unspecified
*****
Coeficiente de correlacao (Previsão um intervalo de amostragem a frente)
99.89
```

---



A seguir é mostrado o resultado obtido do sistema quando a válvula de resfriamento é manipulada.

A Figura 41 apresenta o resultado encontrado comparando o modelo obtido versus os pontos reais.

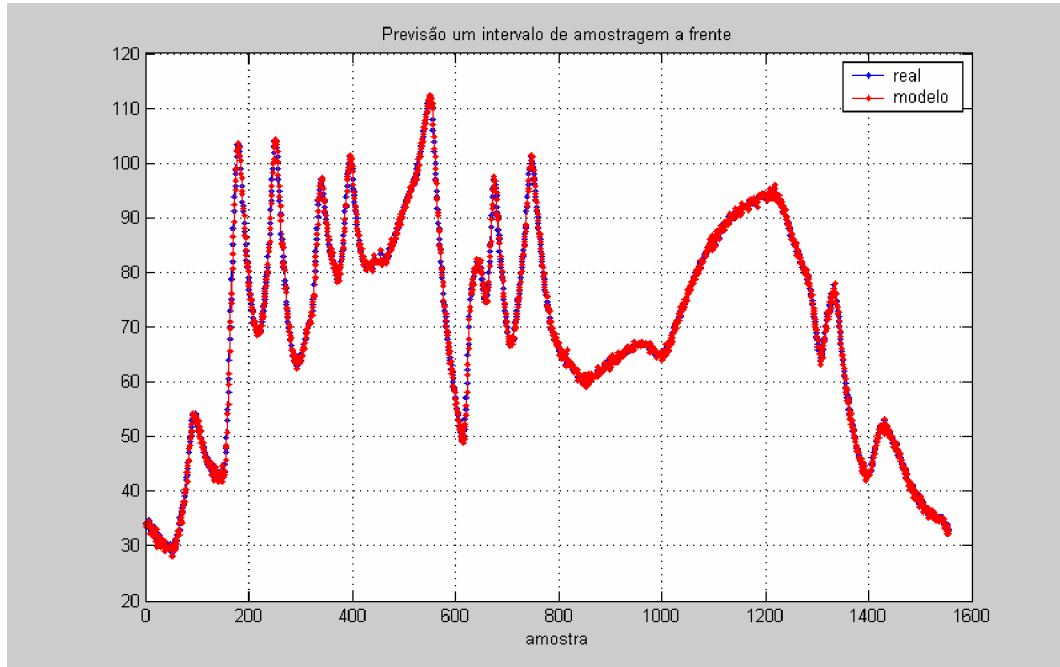


Figura 41 – Comparação entre o modelo obtido e os dados reais

A seguir o código utilizado para obtenção dos resultados em *Matlab*®.

```
[Y,U]=dadosR6re;
[teta,TempoMorto,R]=idtctr2(Y,U);
*****
DETERMINACAO DA FUNÇÃO DE TRANSFERENCIA
*****
*****
MÉTODO DE ESTIMAÇÃO DOS PARÂMETROS DA FUNÇÃO DE
TRANSFERENCIA
digite 0: algoritmo do Erro no modelo (Mínimos quadrados)
digite 1: algoritmo do Erro na saída (Levenberg-Marquardt)
método = 0
*****
*****
DEFINIR PONTO DE OPERACAO (Y0, U0)
digite 0: nao
digite 1: sim
= 0
*****
*****
```

## VALOR DO TEMPO MORTO

definir: digite o valor (em amostras)

determinar: digite [] (pesquisa ate 30 amostras)

Tempo morto = []

\*\*\*\*\*

\*\*\*\*\*

## DEFINIR ORDEM DO MODELO

digite 1: primeira ordem

digite 2: segunda ordem

ordem = 2

\*\*\*\*\*

\*\*\*\*\*

## RESPOSTA INVERSA

digite 0: sem resposta inversa

digite 1: com resposta inversa

resposta inversa = 0

\*\*\*\*\*

\*\*\*\*\*

**A função transferência obtida:**

**Transfer function:**

$$-0.0778 z^{-5}$$

-----

$$1 - 1.46 z^{-1} + 0.4583 z^{-2}$$

Sampling time: unspecified

\*\*\*\*\*

\*\*\*\*\*

Coefficiente de correlacao (Previsão um intervalo de amostragem a frente)

99.86

---

**Vemos o tempo morto encontrado, para o caso do intervalo de amostragem de 10 segundos é.**

**U aq = 5 intervalos (50 s)**

**U re = 4 intervalos (40 s).**

Estes valores vão ser confrontados com valores segundo outras metodologias a fim de se utilizar para fins de treinamento da rede neural Feedforward a qual será utilizada como modelo do controlador MPC.

### 5.1.2. Identificação através do método gráfico para sistemas de primeira ordem com tempo morto.

**Metodologia:** Considerando a equação temporal do sistema de primeira ordem  $y(t) = K (1 - e^{-t/\tau})$ , podemos notar que se considerarmos que quando o tempo  $t = \tau$ , ou seja, estamos considerando que o tempo decorrido é igual a uma constante de tempo, o valor da resposta do sistema sai do valor inicial (antes da aplicação degrau) e atinge 63,2 % da variação total relativa ao degrau, temos  $y(t) = K (1 - e^{-1}) = K (1 - 0,367879) \approx K (0.6321)$ , vemos que  $y(t)$  para este caso é igual a 63,2 % do valor de  $y(t)$  no estado estacionário frente à perturbação degrau ou seja; para obtermos a constante de tempo  $\tau$  basta obtermos o valor do intervalo de tempo necessário para o sistema atingir este valor frente a uma perturbação degrau.

Se sistema possuir atraso puro no tempo ou também conhecido como tempo morto; o desenvolvimento e análogo mais consideramos  $t = 0$  o tempo em que sistema começa a responder a perturbação.

Sendo assim, a partir das curvas de resposta obtidas para o sistema, podemos obter aproximações para sistemas de primeira ordem com tempo morto.

**Caso 1:** Aquecimento geral com degrau de 25 unidades na válvula de aquecimento, deixando a válvula de resfriamento fechada. A Figura 42 ilustra a curva de resposta.

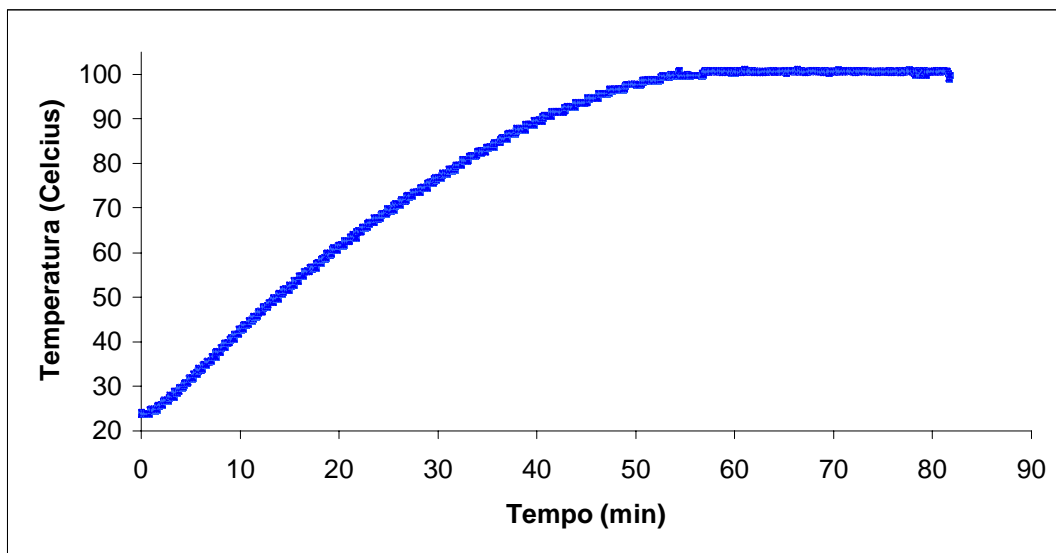


Figura 42 – Curva de resposta para o sistema

Os resultados obtidos por esta metodologia foram os que seguem:

**CALCULO DAS CONSTANTES DO PROCESSO**

**K = Diferença da variação / amplitude do degrau (em volts): 3,044184 °C/% ab.**

**Tempo morto: 0.835833 Minutos = 50.15 segundos**

**TAU = 25.929 minutos = 1555.74 segundos.**

Observando-se a Figura 42 simplesmente não se pode visualizar o tempo morto, isto foi feito se tomando apenas os pontos iniciais do conjunto de dados e observando o início da variação da temperatura em comparação com o momento da perturbação. O tipo de sistema (1ª ordem) pode ser sugerido pela forma da curva de resposta obtida.

**Caso 2:** Resfriamento com degrau de 25 unidades na válvula de resfriamento, deixando a válvula de aquecimento fechada. A Figura 43 ilustra a curva de resposta.

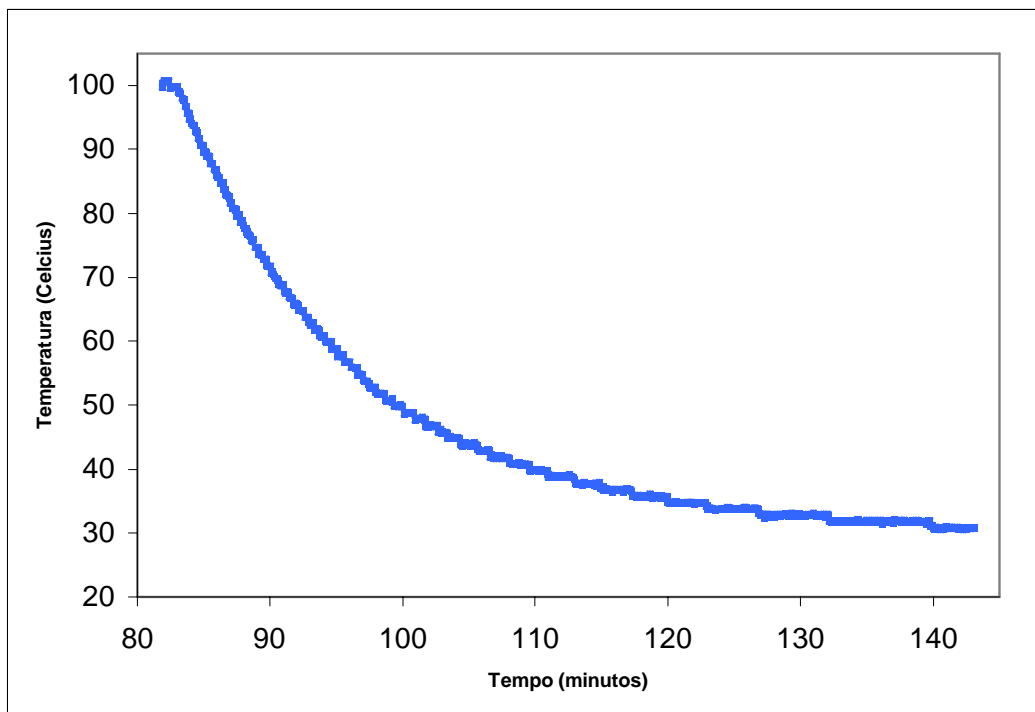


Figura 43 – Curva de resposta obtida

Os resultados obtidos por esta metodologia foram os que seguem:

**CALCULO DAS CONSTANTES DO PROCESSO**

**K = Diferença da variação / amplitude do degrau (em volts): -2,78167 °C/% aber.**

**Tempo morto: 0.497 Minutos**

**29.82 segundos**

**TAU = 14.60787 minutos**

**876.4725 segundos**

Novamente não se pode observar na Figura 43 o tempo morto do sistema, o mesmo foi obtido utilizando-se a mesma metodologia do Caso 1.

Podemos notar que por esta metodologia obtivemos os valores de tempo morto para aquecimento de cerca de 5 intervalos de amostragem e para o resfriamento de cerca de 3 intervalos de amostragem.

Para tanto vamos utilizar neste trabalho o valor de tempo morto de aquecimento de **5 intervalos de amostragem, ou seja, 50 segundos e para resfriamento de 4 intervalos de amostragem, ou seja de 40 segundos.**

Pois, para o Caso 1 (válvula de aquecimento), houve coincidência quanto a comparação entre os métodos gráfico e de Mínimos Quadrados), e quanto ao Caso 2 (válvula de resfriamento); houve divergência, então vamos tomar o maior valor encontrado. Notar que foram considerados sistemas de ordem diferentes para os métodos de identificação utilizados, e no Caso 1 o tempo morto encontrado foi o mesmo, para o Caso 2 a diferença foi pequena (1 intervalo de amostragem).

### 5.1.3. Parâmetros preliminares para o controlador PID

A partir dos resultados encontrados para a função transferência do processo em questão podemos calcular os parâmetros para um controlador PID (Proporcional, Integrativo, Derivativo) por meio de muitas metodologias. Foi desenvolvido um Software o qual elimina a tarefa enfadonha de calcular termo a termo os parâmetros para os Métodos de Cohen-Coon e ITAE. Esta é uma ferramenta útil a fim de se minimizar o tempo devotado a estes tipos de cálculo. O código utilizado pode ser encontrado no **Apêndice F**. Uma ilustração da tela do programa é apresentada a seguir na Figura 44.

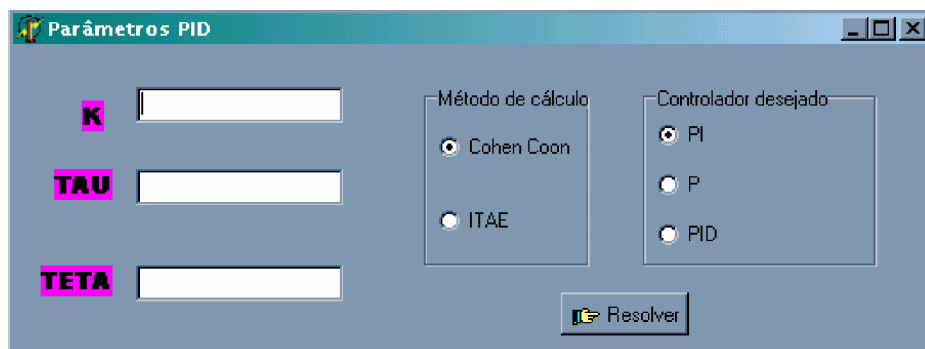


Figura 44 – Tela do programa para determinação dos parâmetros preliminares PID

Com o auxílio desta ferramenta foram calculados parâmetros iniciais para um controlador PID ser aplicado na planta de controle em questão, podendo proceder o ajuste fino deste(s) controlador(es) para os casos em que melhor se apresentar.

As tabelas a seguir apresentam os resultados encontrados considerando modelo de primeira ordem com tempo morto.

Tabela 6 - Parâmetros ajustados para controlador PID – Caso 1

Parâmetros do controlador PID							
Item	Caso 1 – (Controlador PI)			Caso 1 – (Controlador PID)			Caso 1 – (Contr. P)
	Kc	Kd	KI	Kc	Kd	KI	Kc
Método							
Cohen-Coon	9,1988180	0	156,49510	13,669447	18,130103	121,77657	10,299991107287
ITAE	4,4749980	0	1518,2674	5,8745226	19,711966	1966,1117	0

Tabela 7 - Parâmetros ajustados para controlador PID – Caso 2

Parâmetros do controlador PID							
Item	Caso 2 – (Controlador PI)			Caso 2 – (Controlador PID)			Caso 2 – (Contr. P)
	Kc	Kd	KI	Kc	Kd	KI	Kc
Método							
Cohen-Coon	9,5396776	0	92,727427	14,178347	10,776970	72,356403	10,686187303028
ITAE	4,6611324	0	855,60744	6,1407005	11,676232	1108,0343	0

Estes resultados apresentam parâmetros de um controlador PID para o caso do sistema ser considerado de primeira ordem com tempo morto. A partir destes parâmetros pode-se obter um controlador PID procedendo-se um “ajuste fino” ou manual dos parâmetros. A partir dos dados da Tabela 6 em azul, Caso 1, método de Cohen-Coon, controlador PID;  $K_c = 13,67$ ;  $T_i = 18,13$  e  $T_d = 121,77$ , após alguns ajustes por tentativa e erro chegou-se nos seguintes parâmetros:  $K_c = 1,8$ ,  $T_i = 144,1141$  e  $T_d = 7$ , realizando-se o experimento no reator de aço descrito sob agitação tendo como seu conteúdo água. Este controlador PID apresentou um bom desempenho para controlar a temperatura em faixas que variam de 30 graus celcius a 90 graus celcius, apresentando-se bom na mudança de *set-point* e para manutenção da referência, para o caso de um controle somente com água no reator. Podemos observar este comportamento na Figura 45.

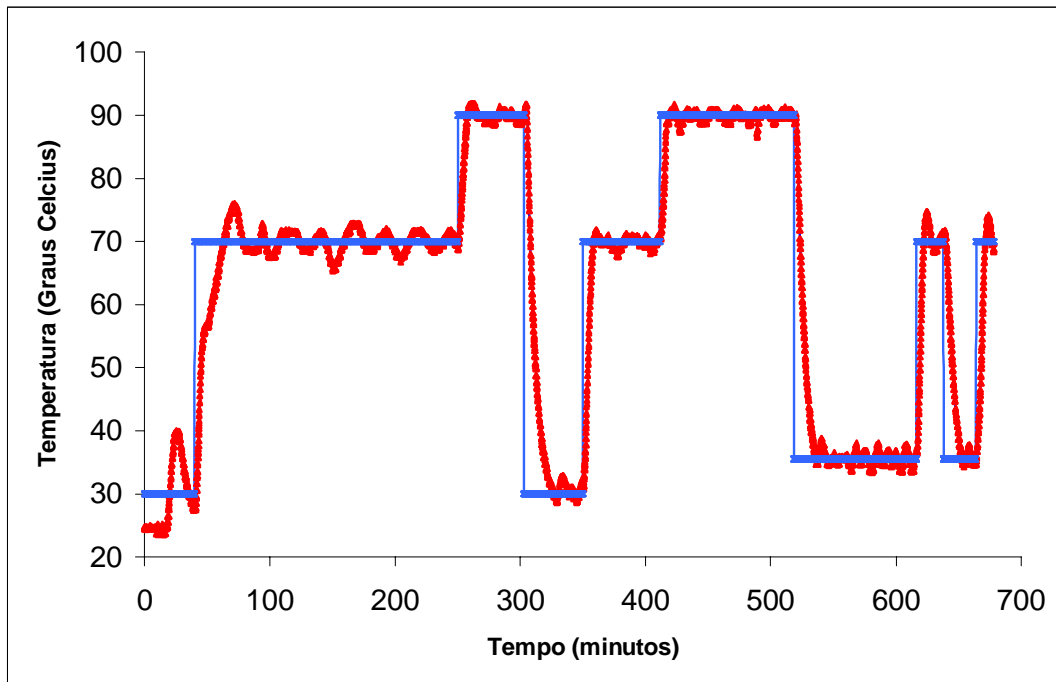


Figura 45 – Resposta do Controlador PID a variações de referência a partir de ajustes clássicos de parâmetros

A Figura 46 apresenta as ações de controle tomadas neste experimentos pelo controlador.

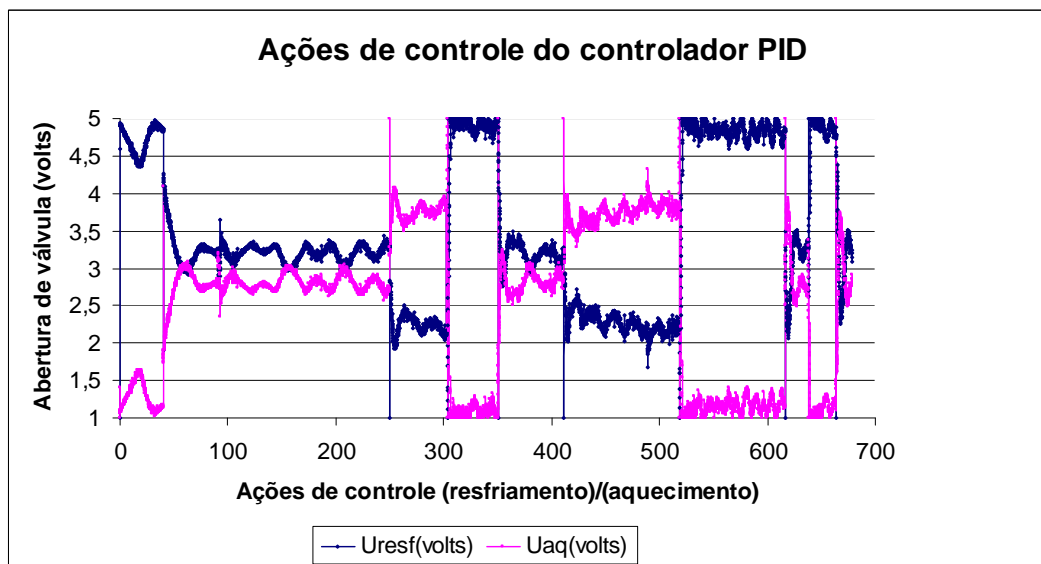


Figura 46– Ações de controle do controlador PID

Cabe registrar que o controlador PID utilizado funciona com sistema em *Split-Range* para as válvulas de controle. Posteriormente vamos aplicar um controlador Preditivo baseado em Rede Neural com estratégia de controle *MISO* (ações de controle

das válvulas de aquecimento e resfriamento independentes). E poderemos ter uma idéia se haverá melhora significativa com esta mudança.

Posteriormente, para se verificar a possibilidade da aplicação do controlador PID ajustado anteriormente, procedeu-se uma reação química de polimerização, para a obtenção de poliestireno, a partir de uma suspensão com cerca de 30 % de em fase orgânica; em função da quantidade de água ser significativa, se espera que o controlador tenha condições de oferecer uma boa atuação para manutenção da referência e também para transição de referência. A reação foi conduzida a uma temperatura de 85 celcius. A Figura 47 apresenta a resposta obtida para esta reação com o uso do controlador PID.

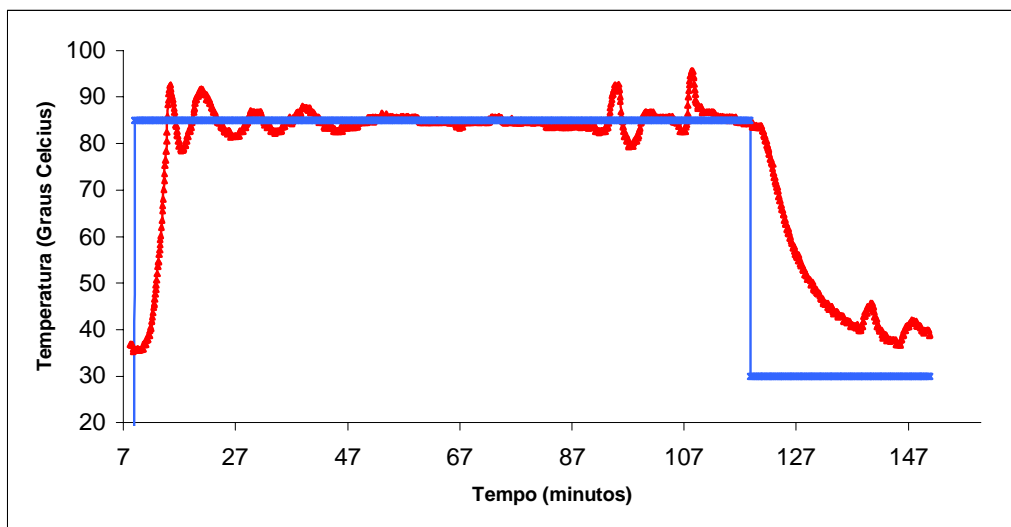


Figura 47 – Experimento aplicando o controlador PID ajustado por métodos clássicos a uma reação de polimerização do estireno

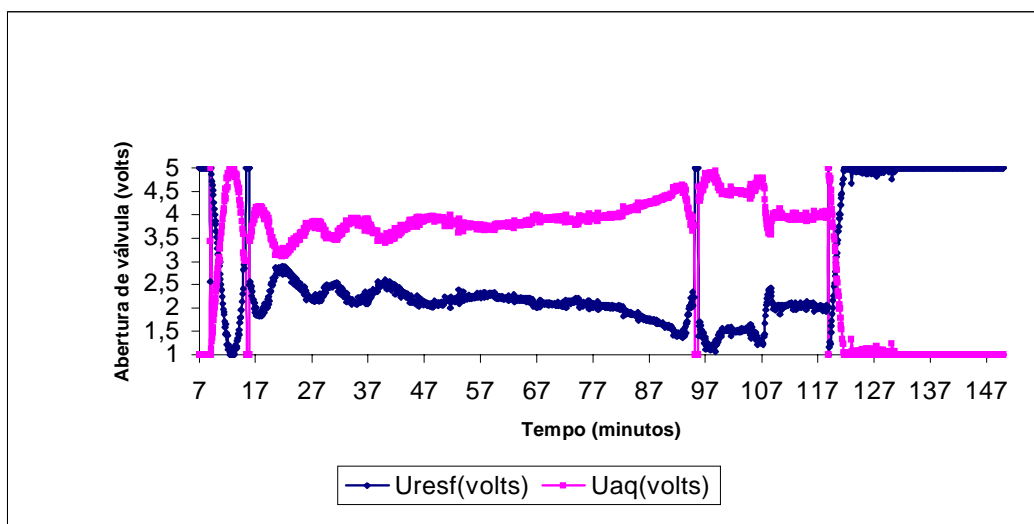


Figura 48- Ações de controle do controlador PID para reação química (com uma breve intervenção manual na transição)



Neste caso, houve necessidade de intervenção manual junto ao controlador na transição da referência, pois a reação química é exotérmica e apresenta-se como uma mudança na dinâmica do sistema. Como o controlador havia sido ajustado via metodologia tradicional e ajustado mais acuradamente manualmente após a determinação dos parâmetros preliminares (Tabela 6), com o reator com água, apesar da reação química utilizada ser realizada em uma suspensão com água (cerca de 30 % de fase orgânica), esta mudança de dinâmica ocorrida não permitiu ao controlador PID obter bons resultados, sendo que mesmo na manutenção da referência, quando inicia-se provavelmente um processo de transição vítrea (por volta de 97 minutos de experimento); onde ocorre mais liberação de calor ao meio, ocorrem oscilações de temperatura muito grandes que não permitem a obtenção de um produto dentro das características desejadas, ou pode ter sido causado por flutuação na pressão da rede de água de refrigeração ou mudança na dinâmica do processo.

A Figura 48 apresenta as ações de controle tomadas na condução desta reação, sendo que a inversão rápida das válvulas de aquecimento e resfriamento no início da reação (por volta de 16 minutos) foi conduzida manualmente para evitar elevação sobre elevação, logo em seguida o controlador foi colocado no modo automático para observarmos sua atuação no controle servo como já apresentado na Figura 47.

Vemos que em função do tempo morto do processo, bem como das variações de dinâmica que ocorrem durante a reação química, a utilização de um controlador PID a parâmetros fixos pode ser dificultada.

Contudo, temos necessidade de aplicar outras metodologias as quais possam representar melhor o sistema em questão.

## **5.2. Aplicação de um controlador Preditivo Baseado em um modelo não-linear de uma Rede Neural Feedforward treinada *off-line***

Neste trabalho é proposta a aplicação de um controlador preditivo baseado em modelo (MPC), utilizando uma rede neural feedforward como modelo para representar o processo de aquecimento de temperatura em um reator de polimerização (Indústria Petroquímica de III Geração); sendo a RNA treinada *off-line* a partir de uma série temporal que possa representar o processo.

Neste trabalho optou-se por não executar o levantamento da série temporal a qual representará o processo frente o treinamento da rede neural RNA utilizando-se comportamento *Split-Range* nas válvulas de controle, esperando-se obter um controlador bom para o processo já que a estratégia aplicada é de um sistema MISO (múltiplas saídas e uma entrada). CANCELIER (2004) utilizou os padrões de treinamento da RNA com as perturbações adotando comportamento *Split-Range* para formação dos padrões de treinamento, e em um sistema onde a dinâmica era diferente; também trabalhou com outras faixas de temperatura (60 graus celcius a 90 graus celcius). Neste trabalho se trabalhou uma faixa de temperatura de 35 graus celcius a 90 graus celcius).

As diferenças físicas aferidas ao sistema de controle de temperatura, evidenciam que a dinâmica para o processo sofreu alterações, e uma investigação sobre a eficiência do controlador preditivo para este caso torna-se interessante.

Para tanto, procedeu-se uma nova identificação do sistema, utilizando-se metodologias descritas anteriormente, sendo que os resultados utilizados são apresentados a seguir.

Uma aplicação de um controlador PID ajustado segundo métodos tradicionais é realizada, os resultados são apresentados a posteriormente.

Antes porém, torna-se necessária uma breve descrição da problemática envolvida no sistema de controle de temperatura bem como uma justificativa para aplicação do controlador MPC.

### 5.2.1. Justificativas: Tempo Morto e não-linearidades

Uma das problemáticas encontradas neste sistema é o tempo morto envolvido tanto para o aquecimento quanto para o resfriamento. Tempo morto pode ser entendido simplificarmente como o tempo o qual o sistema leva para responder a uma variação em uma das variáveis manipuladas (no caso: válvula de vapor – aquecimento e válvula de água fria – resfriamento)

No caso em questão, e na maioria dos casos, vamos encontrar dinâmicas diferentes para o aquecimento e para o resfriamento, portanto teremos tempos mortos diferentes para estes processos. *INGIMUNDARSON & HÄGGLUND (2001)* descrevem procedimentos para obtenção de controladores com compensação de tempo morto, segundo eles: “... quando se têm grandes tempos mortos no processo; a performance de controle obtida com controladores PID é, contudo limitada. Para estes casos o controlador por compensação de tempo morto pode melhorar a performance consideravelmente”.

Portanto, para casos em que o tempo morto possa ser representativo, a aplicação do controlador PID simplesmente pode não ser satisfatória, sendo interessante a investigação de controladores que possam tratar melhor o problema.

Ainda nesta linha, *SHINSKEY (2001)* um comparativo entre o uso de controladores MPC e PID com compensação de tempo morto; “Enquanto controladores baseados em modelo têm sido utilizados com sucesso para controlar máquinas de papel e em processos dominados pelo atraso no tempo .... Um controlador PID com compensação por atraso no tempo pode ter o funcionamento similar ao MPC, e ambos são duplamente efetivos frente a um controlador PID convencional”.

Neste caso, o processo de temperatura pode apresentar resultado satisfatório caso se consiga incluir na lei de controle o tempo morto envolvido no processo.

Segundo *SHINSKEY (2001)*: “Uma classe importante de processos com atraso dominante no tempo incluem aquecedores e colunas de destilação, consistindo em atrasos distribuídos”.

No sistema em estudo temos além da presença de tempo morto, a dinâmica variável de resfriamento e aquecimento, não-linearidades do processo devido a própria natureza do sistema também são implícitas. Para tanto, o uso de um controlador linear PID convencional, bem como de um PID por compensação de tempo morto, pode não ser satisfatório devido a estas não-linearidades e variações de dinâmica.

*KAYA (2002) descreve: “... plantas com grandes atrasos no tempo podem não ser efetivamente controladas usando-se simplesmente um controlador PID convencional. A principal razão para isto é que a fase de atraso que contribuí para o tempo morto tende a desestabilizar o sistema em malha fechada”.*

Contudo, torna-se necessário propor a utilização de um controlador que possa atuar satisfatoriamente para o sistema com as problemáticas citadas de não-linearidades, com tempo morto e dinâmica variável.

*ASTRÖM & HÄGGLUND (2001), descrevem a utilização de Controle de Modelo Preditivo (MPC) em plantas industriais atuando como controle supervisorio em conjunto com controladores PID: “Muitos são os melhoramentos creditados ao MPC em processos industriais atualmente têm melhorado o ajuste de malhas clássicas. O MPC, entretanto, traz drásticos melhoramentos em respostas na transição de referência para sistemas multivariáveis porque a coordenação das variáveis é satisfeita.”*

A partir desta afirmação pode-se prever que talvez uma metodologia mais acurada de controle possa ser interessante para tratar o processo de controle de temperatura.

*JEYAB, et.al.(2001) descrevem: “Colunas de destilação são importantes processos unitários em refino e necessitam ser mantidos próximos de condições de operação ótimas por razões econômicas. .... Deste modo, o sistema de controle necessita ser restringido em um problema de otimização multivariável on-line e o modelo de controle preditivo (MPC) é o melhor processo para esta proposta.”*

A utilização de modelos lineares como representação do processo em controladores MPC foi estudada por *BLOEMEN, et.al. (2000)* em um controle de composição de uma coluna de destilação, apresentando resultados satisfatórios. Convém lembrar que este tipo de processo apresenta muitas não-linearidades e é uma operação unitária que envolve muitas trocas de calor e massa.

Pode-se então sugerir que a aplicação de um controlador Preditivo Baseado em Modelo possa ser capaz de representar satisfatoriamente o sistema e controle temperatura em estudo, e o controlador baseado em um modelo de uma Rede Neural Artificial (RNA) utilizado por *CANCELIER (2004)* apresenta-se como uma aplicação satisfatória para o problema.por apresentar-se como um controlador que utiliza um modelo não-linear o qual pode representar bem o sistema em estudo.

Esta aplicação é reforçada ainda mais pela observação de *ALI – (2002): “Devido a fatores simpáticos do MPC de modo a tratar as restrições para processos com grande*

*número de variáveis manipuladas e controladas, se torna muito ampla a utilização deste sistema de controle em indústrias químicas.”*

Contudo, a utilização de um modelo não-linear estático pode não ser satisfatória caso ocorram mudanças na dinâmica do processo, tornando-se necessário um ajuste on-line dos pesos da camada de saída da rede neural conforme utilizado por *CANCELIER (2004)*. Estes pesos serão atualizados por uma adaptação por algoritmos genéticos.

A seguir apresenta-se os resultados obtidos no trabalho para esta aplicação.

### **5.2.2. Treinamento off-line da rede neural Feedforward**

A fim de se formar padrões para treinamento da Rede Neural, é necessário a determinação do tempo morto envolvido no processo de aquecimento e de resfriamento, já que a aplicação do controlador se dá por meio de um sistema **MISO**. O mesmo já foi demonstrado anteriormente e os valores utilizados neste trabalho são de **5 intervalos de amostragem** (50 segundos) para o aquecimento e de **4 intervalos de amostragem** (40 segundos) para o resfriamento (no caso do resfriamento as duas metodologias aplicadas foram divergentes, a partir do método de mínimos quadrados se encontrou o valor de 40 segundos e pelo método gráfico de cerca de 30 segundos, optou-se por adotar o valor maior neste caso).

A partir dos valores de tempo morto calculados para o sistema, pode-se obter os padrões que devem ser alimentados a Rede Neural para seu treinamento, os dados utilizados para obtenção dos padrões foram retirados da série temporal mostrada na Figura 38 a partir de variações aleatórias nas válvulas de aquecimento e de resfriamento; a Figura 49 apresenta estas variações nas válvulas que foram aplicadas manualmente e aleatoriamente.. *MAZZUCCO (1996)* e *CANCELIER (1997)* demonstraram ser importante a consideração do tempo morto no processo de formação dos padrões para treinamento da Rede Neural. Contudo, para a formação dos padrões; as entradas do processo devem estar defasadas  $d$  intervalos de amostragem, onde  $d$  corresponde ao valor do tempo morto em intervalos. A Equação 60 apresenta a função geral que representa a Rede Neural para a modelagem do processo, considerando que a função do modelo é atuar como preditor para o processo, ou seja, deve ser treinado para que possa prever um passo futuro em relação ao atual,  $y(k+1)$ , dessa forma, as perturbações aplicadas devem estar defasadas pelo menos  $d+1$  intervalos de amostragem em relação a saída.

$$y(k+1) = f[y(k), y(k-1)U_1(k-5)U_2(k-4)] \quad \text{Eq.(60).}$$

A Tabela 8 ilustra a formação de um padrão para entradas da rede considerando o deslocamento temporal dos componentes, admitindo um tempo  $d_1$  de 5 intervalos de amostragem e  $d_2$  de 4 intervalos de amostragem para a previsão da variável de saída (neste caso a temperatura do sistema) em  $(k+1)$ .

**Tabela 8 - Formação dos padrões para treinamento da RNA**

Instante (k)	y(k)	y(k-1)	U <sub>1</sub> (k-5)	U <sub>2</sub> (k-4)
1	y(1)	y(1)	U <sub>1</sub> (1)	U <sub>2</sub> (1)
2	y(2)	y(2)	U <sub>1</sub> (2)	U <sub>2</sub> (2)
3	y(3)	y(3)	U <sub>1</sub> (3)	U <sub>2</sub> (3)
4	y(4)	y(4)	U <sub>1</sub> (4)	U <sub>2</sub> (4)
5	y(5)	y(5)	U <sub>1</sub> (5)	U <sub>2</sub> (5)
6	y(6)	y(6)	U <sub>1</sub> (6)	U <sub>2</sub> (6)
7	y(7)	y(7)	U <sub>1</sub> (7)	U <sub>2</sub> (7)

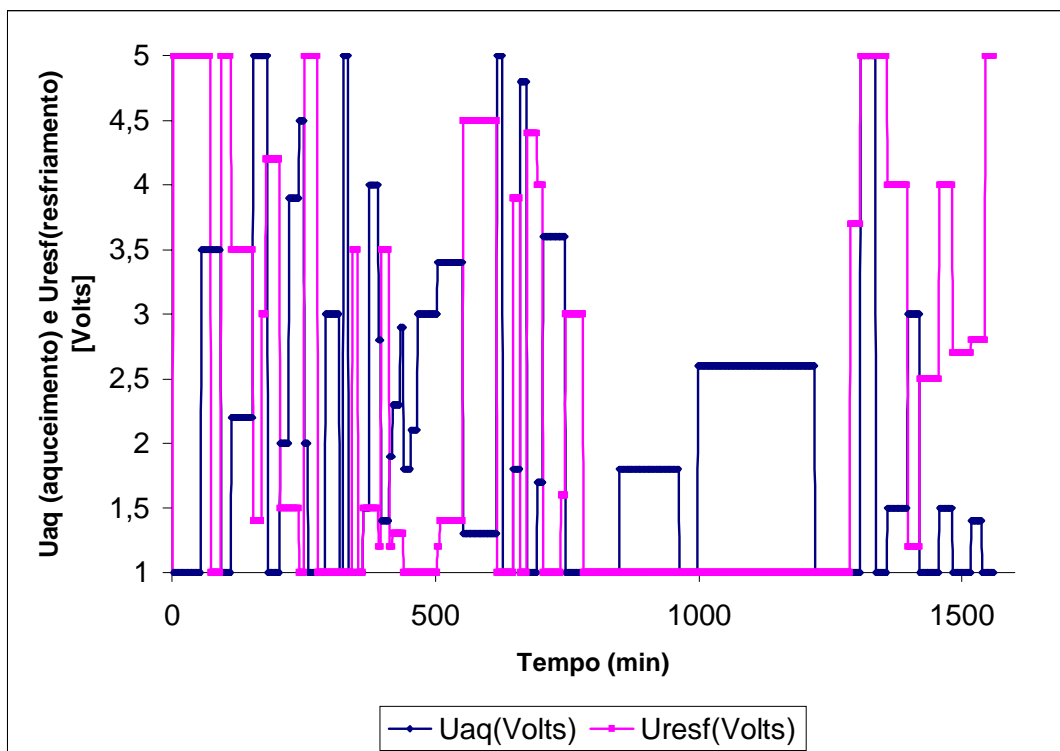


Figura 49 – Ações das válvulas tomadas na identificação do sistema

Sendo que 1 volt corresponde a 0% de abertura da válvula e 5 volts correspondem a 100% de abertura.

A obtenção do grupo de dados já foi descrita anteriormente, cabe ressaltar que foram coletados em intervalos de amostragem de 10 segundos, que representa o mesmo intervalo de amostragem que deve ser utilizado pelo controlador.

Para garantir que a rede identifique bem o sistema é importante que se tenham perturbações aleatórias, aplicadas em toda a faixa de trabalho. Cabe ressaltar que. Como as válvulas estão desacopladas, a princípio qualquer combinação para abertura das válvulas pode ser aplicada. Neste trabalho optou por **deixar as perturbações aplicadas nas válvulas aleatórias** para a formação dos padrões.

Após o grupo de dados formado, procedeu-se o treinamento da Rede Neural. Como *CANCELIER (2004)* demonstrou que o treinamento via algoritmo híbrido utilizando algoritmos genéticos e mínimos quadrados demonstrou ser superior a outras metodologias de treinamento, optou-se por utilizar somente o algoritmo híbrido para o treinamento da Rede Neural, mas aplica-lo utilizando um número de gerações elevado (de ordem de 40000), para observar o tempo de aplicação e a resposta que a rede pode trazer para tal número de gerações; que é bem mais elevado que o utilizado pelo autor.

O algoritmo híbrido já foi introduzido nas seções anteriores e seus operadores também, cabe lembrar que o Algoritmo Genético treina os pesos e bias da camada intermediária da Rede e o Mínimos Quadrados treina os pesos e bias da camada de saída da Rede. Todos os métodos utilizados foram originalmente baseados nas ferramentas do aplicativo Matlab®(*The MathWorks™, inc*).

Foram utilizados os seguintes parâmetros de treinamento da rede:

#### **Método Híbrido de treinamento:**

Sistema híbrido, GA conjugado com mínimos quadrados:

Para o GA:

- Método de seleção por *rolleta*, com escalonamento por truncamento sigma;
- *Crossover* intermediário, taxa de *Crossover* = 1,0;
- Mutação exponencial, taxa de mutação = 0,5% ou 0,005;
- População = 50 indivíduos;
- Faixa de variação dos pesos = [-5,+5]. Aqui define-se o espaço de busca, ou seja, a faixa de variação de cada gene do cromossomo;
- Número de gerações: 40000.

### Mínimos quadrados:

- Método tradicional, sem qualquer modificação em relação ao padrão da ferramenta do *Toolbox* do aplicativo MatLab.

A definição dos parâmetros seguiu os termos do trabalho de *CANCELIER (2004)*, salvo as alterações no número de gerações e no tamanho da população.

A Figura 50 apresenta o resultado para o treinamento da rede. Pode-se notar que a aproximação foi muito boa, sendo que o coeficiente de correlação obtido foi de mais de 99 %.

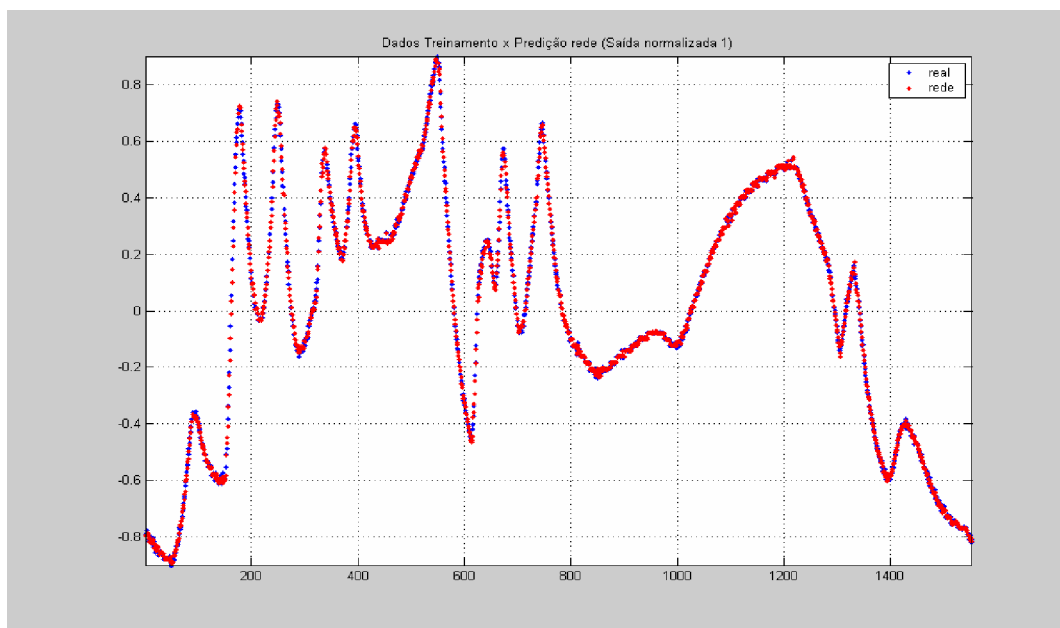


Figura 50– Sobreposição dos pontos experimentais e os pontos gerados pela rede treinada

Este resultado foi considerado o melhor resultado obtido de uma série de cerca de 30 treinamentos realizados, com diferentes faixas de gerações e de tamanho de população. A consideração foi feita tomando-se os valores dos pesos e *bias* e aplicando-se ao controlador preditivo, no Reator experimental de Aço, sendo que a rede que apresentou melhor resultado no tocante a controlar a temperatura na faixa desejada de 90 graus celcius é a apresentada na Tabela 9.

Esta faixa de temperatura é a faixa na qual é realizada a reação de polimerização de estireno que mais tarde foi realizada com o uso deste controlador preditivo baseado no modelo de rede Neural que se apresentou anteriormente. Contudo, se testou o controlador em outras faixas de temperatura também, obtendo-se resultados satisfatórios; sendo que estes testes se realizaram com o reator contendo água somente.



Os pesos e bias obtidos estão apresentados na Tabela 9.

Tabela 9 - Pesos e bias obtidos no treinamento da Rede Neural *(o arquivo esta no mesmo formato que é utilizado no software de controle)*

Número de neurônios das camadas de entrada/intermediária/saída.
4 5 1
Limites para a normalização/desnormalização para os dados de entrada (superior inferior).
5 1
5 1
111.953786 28.9889824
111.953786 28.9889824
Limites para a normalização/desnormalização para os dados de saída (superior inferior).
111.953786 28.9889824
Pesos dos neurônios da camada escondida. Cada coluna define um neurônio da camada de entrada. Cada linha define um neurônio da segunda camada.
-0.54429316221250 0.00812563803749 -4.09768339133109 5.00000000000000
-0.61001302315772 -0.48769972721168 -5.00000000000000 0.85288911197399
-0.53641492143468 0.09901355532485 4.28100378029950 -5.00000000000000
0.00003747469983 0.00050757618859 -0.13742109525872 -0.14898234665155
-1.03754904360086 5.00000000000000 4.78440101941269 -5.00000000000000
Bias dos neurônios da camada escondida.
-2.49533907593702
-2.97581211021921
0.77201433151343
-0.03745531432227
-0.52583284476815
Pesos dos neurônios da camada de saída. Cada coluna define um neurônio da camada de saída. Cada linha define um neurônio da camada escondida.
0.52827195661005
-0.01570200957908
-0.10341808455778
-3.24167406181156
-0.00664419776106
Bias dos neurônios da camada de saída.
0.44668626344331

### 5.2.3. Avaliação do desempenho do Controlador Preditivo Baseado em Modelo aplicado ao sistema de controle de Temperatura

Neste ponto serão apresentados os resultados obtidos para aplicação da Rede Neural Feedforward treinada off-line conforme mostrado na Figura 32 no controlador preditivo baseado em modelo utilizando a rede neural como tal. O controlador utilizado têm a rede atuando como preditor, rede neural estática; ou com adaptação dos pesos e bias dos neurônios da camada de saída da rede neural via metodologia de Algoritmos Genéticos, rede adaptada. Para utilização da rede adaptada, o Algoritmo Genético resolve o problema de minimização do erro médio quadrático em relação ao valor predito e o valor real obtido pelo sistema e procura obter um conjunto de pesos que reduzam a função objetivo do controlador. Esta função foi apresentada na Equação 59 e maiores detalhes sobre o mecanismo de adaptação on-line dos pesos podem ser obtidos na seção 3.2.

A lei de controle utilizada pelo controlador preditivo é da forma de Equação 61, foi implementada em um sistema MISO, ou seja, mais de uma entrada para o processo deverá ser determinada. Tem-se duas variáveis de entrada (variáveis manipuladas): a abertura da válvula de controle de vazão de vapor ( $U_1$ ) e a abertura da válvula que permite a entrada de água de refrigeração ( $U_2$ ).

Com base na minimização do consumo de vapor a Equação 61 foi utilizada como segue, para a função utilizada como critério de otimização.

$$J(N_1, N_2, N_u) = E \left\{ \begin{array}{l} \sum_{j=N_1}^{N_2} (y(k+j) - w(k+j))^2 + \lambda_1 \sum_{j=1}^{N_u} (\Delta U_1(k+j-1))^2 + \\ + \lambda_2 \sum_{j=1}^{N_u} (\Delta U_2(k+j-1))^2 + \lambda_c \sum_{j=1}^{N_u} ((U_1 - 1)(k+j-1))^2 \end{array} \right\} \quad \text{Eq. (61).}$$

O objetivo, ao minimizar a Equação 61, é fazer com que a saída futura  $y(k+j|k)$  siga a referência  $w(k+j)$  e, ao mesmo tempo, os esforços de controle  $\Delta U_i(k+j-1)$  sejam minimizados. Observando a Equação 61, percebe-se que essa possui alguns graus de liberdade ( $N_1$ ,  $N_2$ ,  $N_u$ ,  $\lambda_1$ ,  $\lambda_2$ , e  $\lambda_c$ ) que podem ser modificados para obter o comportamento desejado do sistema controlado. O significado de  $N_1$  e  $N_2$  é bastante intuitivo. Eles indicam o intervalo de tempo que é desejável que a saída siga a

referência. Se  $N_I$  é definido com um valor alto significa que os erros nos primeiros instantes não são importantes. O coeficiente  $\lambda_C$  determina a importância que a abertura da válvula monitorada pela ação  $U_I$ , neste caso a válvula de controle da vazão de vapor, possui para a minimização da função custo. Quanto maior o valor de  $\lambda_C$  mais importância terá o consumo de vapor para a função objetivo, em outras palavras, valores grandes de  $\lambda_C$  tentem a fazer com que se tenha um menor consumo de vapor. Deve-se, no entanto, ter o cuidado para que o termo que regula o consumo não tenha importância muito superior em relação aos outros termos da equação, o que tornaria o controle ineficiente. As ações de controle determinadas estão sujeitas às restrições de estarem entre 1 (0% de abertura – fechada) e 5 (100% de abertura – aberta) volts.

O diagrama de blocos que apresenta, de forma simplificada, o sistema de controle preditivo, utilizado para o processo de polimerização em suspensão do estireno, é ilustrado na Figura 51.

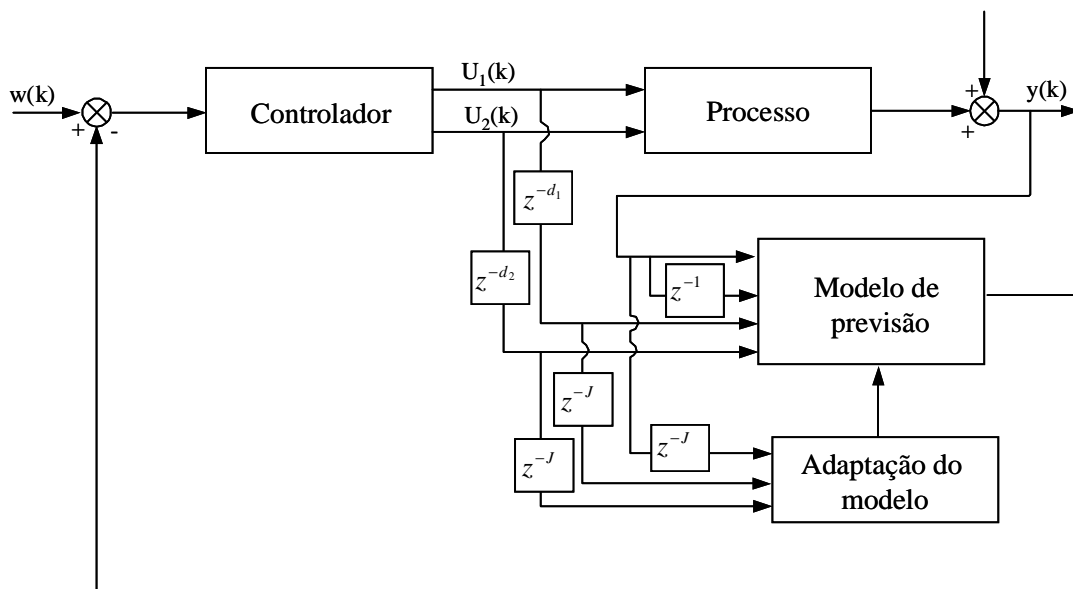


Figura 51 - Diagrama de blocos simplificado do sistema de controle preditivo

Os parâmetros utilizados no controlador Preditivo foram determinados por tentativa e erro, partindo-se inicialmente dos parâmetros utilizados por *CANCELIER (2004)*, avaliando-se o comportamento do sistema para diferentes conjuntos de pesos para rede neural no sistema contendo água (sem reação química).

Neste trabalho, chegou-se aos seguintes valores para os parâmetros:  $\alpha = 0,96, \lambda_1$  e  $\lambda_2 = 0,02, \lambda_c = 0,1$  (**controlador 1**), conforme mostrado na Figura 52; e também se

obteve bons resultados para  $\alpha = 0,75, \lambda_1$  e  $\lambda_2 = 0,02, \lambda_c = 0,1$  (**controlador 2**), conforme mostrado na **Figura 54**. O horizonte de predição utilizado foi de  $N = 7$  intervalos de amostragem ou 70 segundos, sendo um horizonte de predição representativo para o sistema em questão. Para o caso do controlador 1, somente testes com água foram realizados no reator, foi verificado o comportamento do controlador para o caso de transição de *set-point* e manutenção da referência para a faixa de cerca de 35 graus celcius a 90 graus celcius. Posteriormente foi aplicada a adaptação on-line dos pesos da camada de saída da rede neural para este mesmo caso, e verificou-se que o sistema elimina o pequeno off-set (positivo) que apresenta no controlador não adaptado. Os experimentos com o **controlador 1** foram realizados com o reator contendo somente água. A **Figuras 52 e 53**, apresenta o desempenho deste controlador para o caso sem adaptação e as ações de controle tomadas, respectivamente. As **Figuras 54 e 55** apresentam o desempenho do **controlador 1** para o caso com adaptação com os seguintes parâmetros de adaptação: **Janela de Pontos = 20; Número de iterações = 110, Variação superior e Inferior dos pesos = 0,001, Tamanho da População = 60**. A rede utilizada nos testes têm 5 neurônios na camada intermediária.

Podemos observar que o controlador apresenta uma leve tendência a *off-set*, isto pode ser explicado observando-se a Figura 39 que apresenta os padrões utilizados para treinamento da RNA, onde nota-se que com o reator contendo somente água em seu interior, a temperatura em alguns casos ultrapassa os  $100^{\circ}\text{C}$  (temperatura de ebulição da água), com isto; pelo fato de se estar trabalhando em pressão atmosférica, temos mudança de fase da água no interior do reator, e com isto podem ocorrer desvios na dinâmica que esta sendo transmitida a RNA para treinamento em razão destes pontos em que se têm mudança de fase. Ou seja, a rede pode ter sido treinada com informação desnecessária para operação mais eficiente do controlador.

Notamos que a resposta foi satisfatória, ocorreu um pequeno off-set (positivo). Este experimento foi conduzido por 415 amostras, ou seja cerca de 69 minutos. O valor elevado da penalização pela trajetória de referencia (0,96) se dá em função da dinâmica se apresentar muito rápida na transição, o que pode levar a *over-shoot* caso o valor seja muito pequeno.

Conforme já citado, ainda com o **controlador 1** foram realizados experimentos com o reator contendo somente água procurando se aplicar o método adaptativo por algoritmos genéticos, onde se espera elimina o off-set ocorrido com a metodologia sem

adaptação (apenas a Rede Neural Feedforward retro-alimentada). A seguir seguem os resultados obtidos.

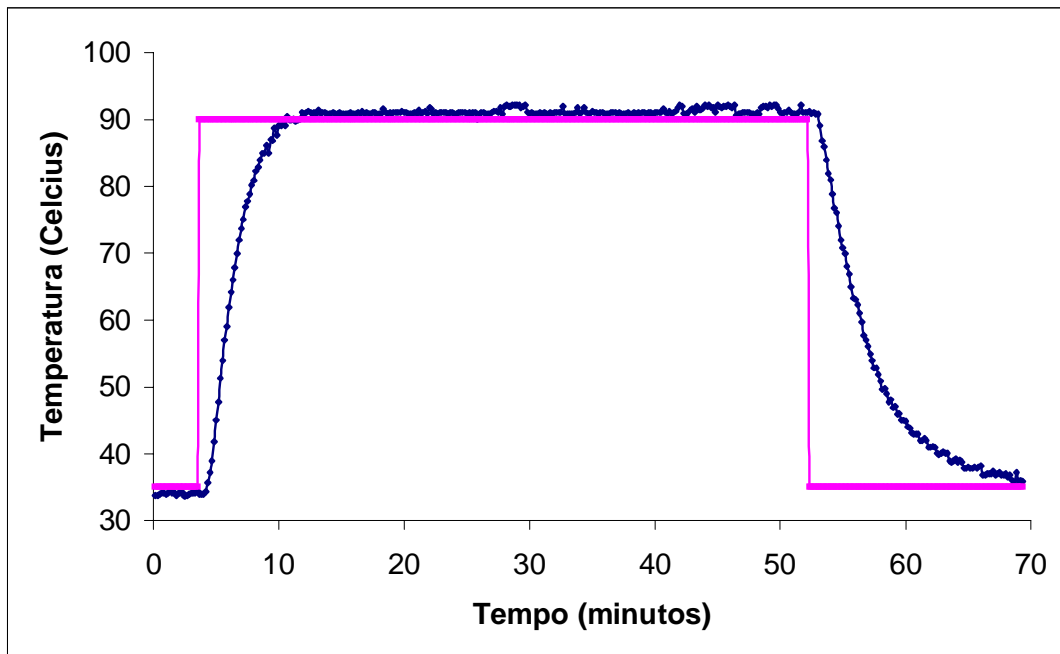


Figura 52 - Desempenho deste controlador 1 para o caso sem adaptação com  $\alpha = 0.96$ .

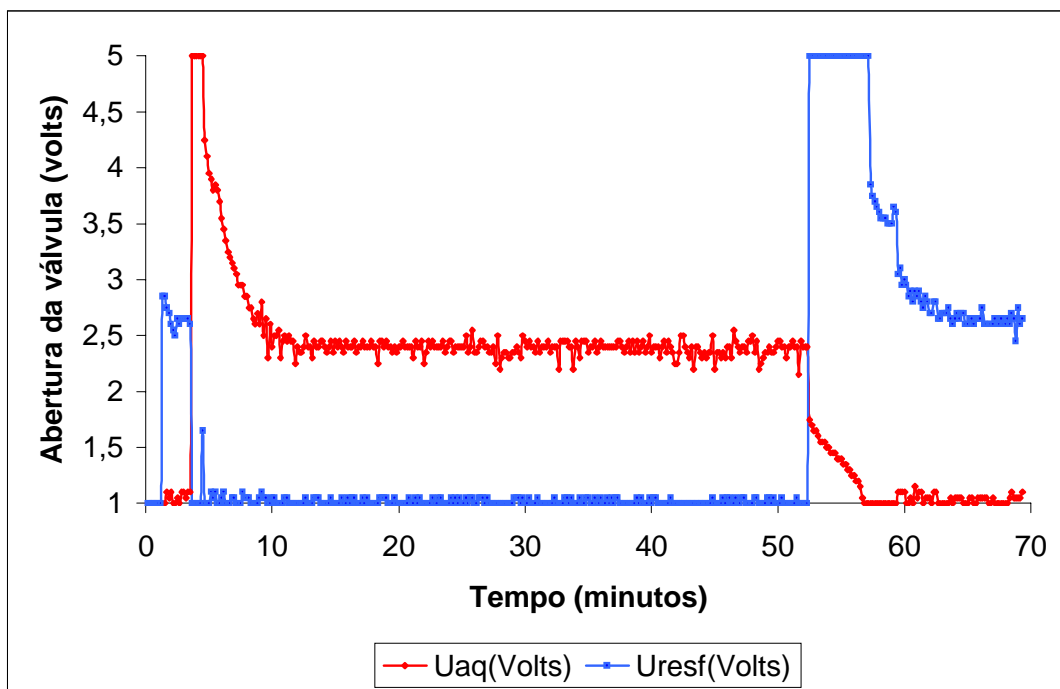


Figura 53 - Ações de controle tomadas pelo controlador 1 para o caso sem adaptação com  $\alpha=0.96$

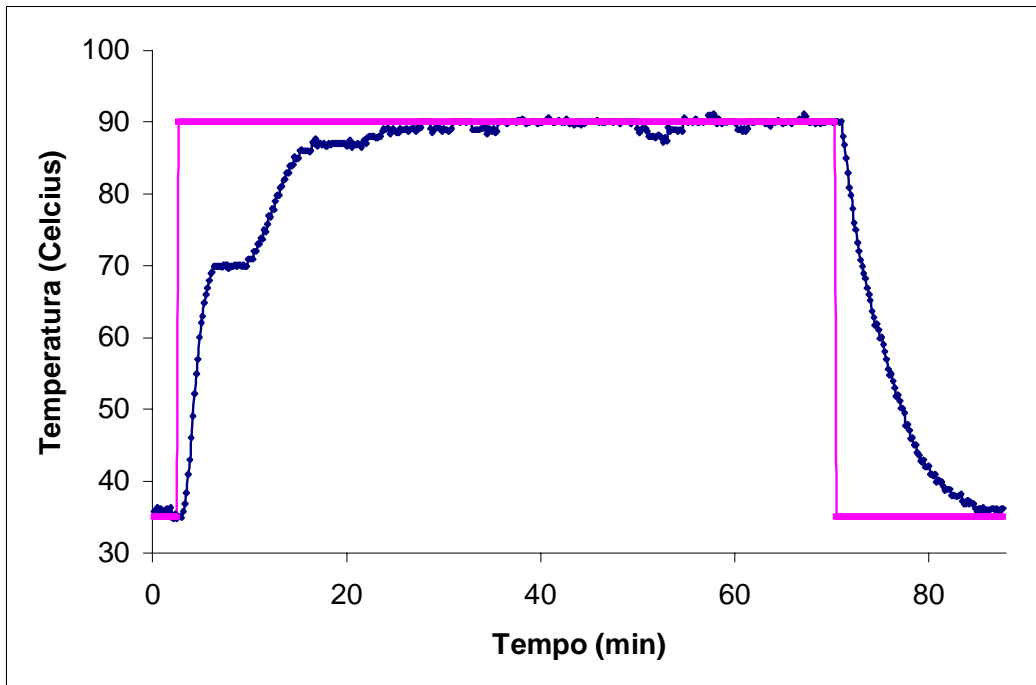


Figura 54 – Resposta do controlador 1 com adaptação por GA

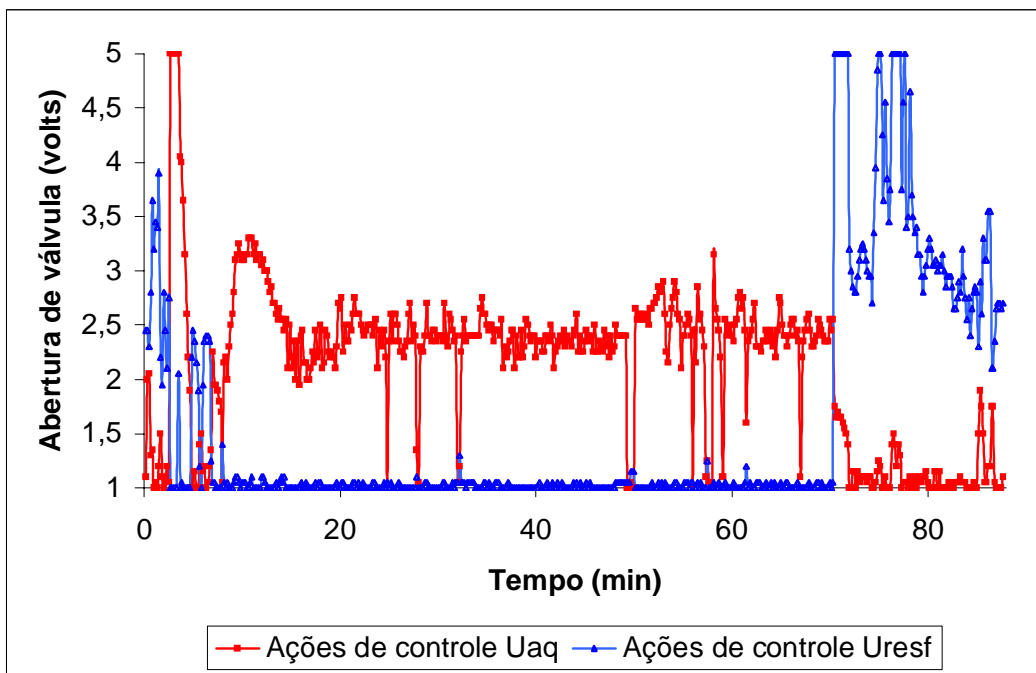


Figura 55 – Ações de controle tomadas pelo controlador 1 com adaptação por GA

Podemos ver que o método adaptativo neste caso eliminou o off-set (positivo) ocorrido com o controlador 1 sem adaptação. O experimento das Figuras 54 e 55 durou 525 amostras, ou seja, cerca de 87,5 minutos.

A seguir foram realizados experimentos reduzindo um pouco a penalização pela trajetória de referência, objetivando deixar o controlador com respostas mais rápidas. Para tanto se procederam testes com os parâmetros descritos para o **controlador 2**. Primeiramente este foi avaliado no sistema de controle de temperatura somente contendo água no reator, sem adaptação, e em seguida foi realizado experimento com este mesmo **controlador 2** no reator com água, com adaptação por GA; os parâmetros utilizados para o método adaptativo neste caso foram: **Janela de Pontos = 20, Numero de Iterações = 110, variação superior e inferior dos pesos = 0,001 e tamanho da população = 60.**

Os resultados obtidos para o controlador 2 no caso sem adaptação com somente água no reator são apresentados nas Figuras 56 e 57. Para o caso com adaptação os resultados obtidos para o controlador 2 são apresentados nas Figuras 58 e 59.

Podemos observar ainda pela Figura 54 que a atuação do controlador frente a transição com a utilização do método adaptativo ocorre suavemente, sendo mais lenta que aplicando o controlador sem adaptação. Isto pode ser atribuído ao elevado valor para o coeficiente da trajetória de referência ( $\alpha = 0,96$ ), que induz a uma transição suave para a referência.

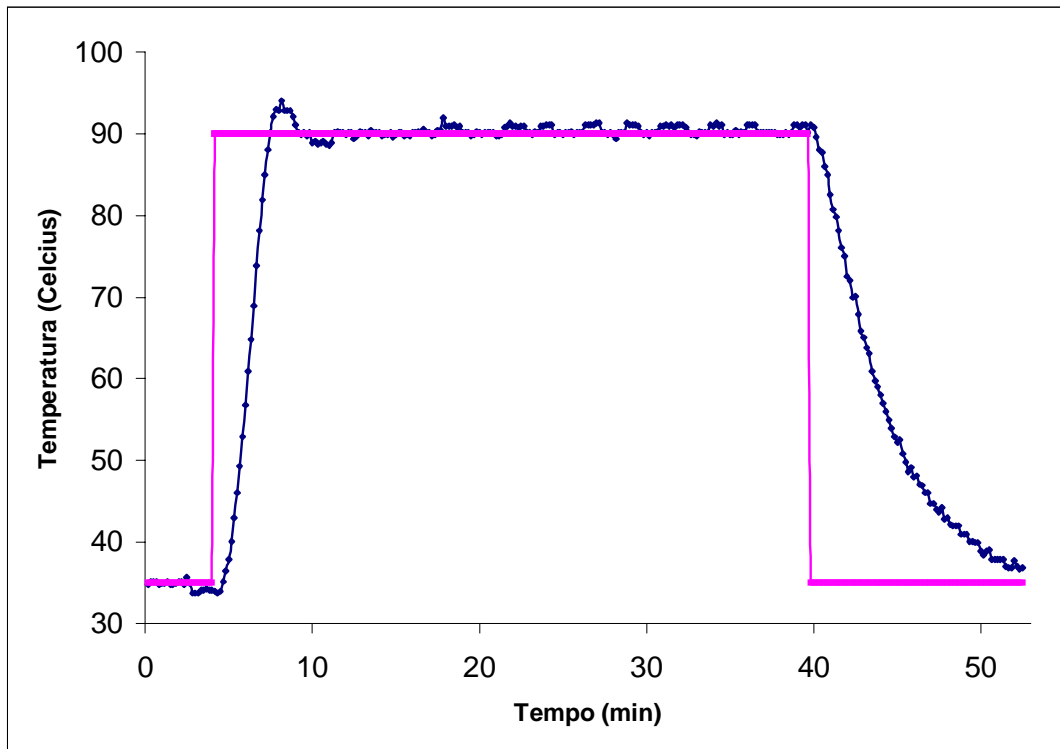


Figura 56 - Curva de resposta do controlador MPC para controlador 2 , sem adaptação (314 amostras cerca de 52 minutos),  $\alpha = 0,75$

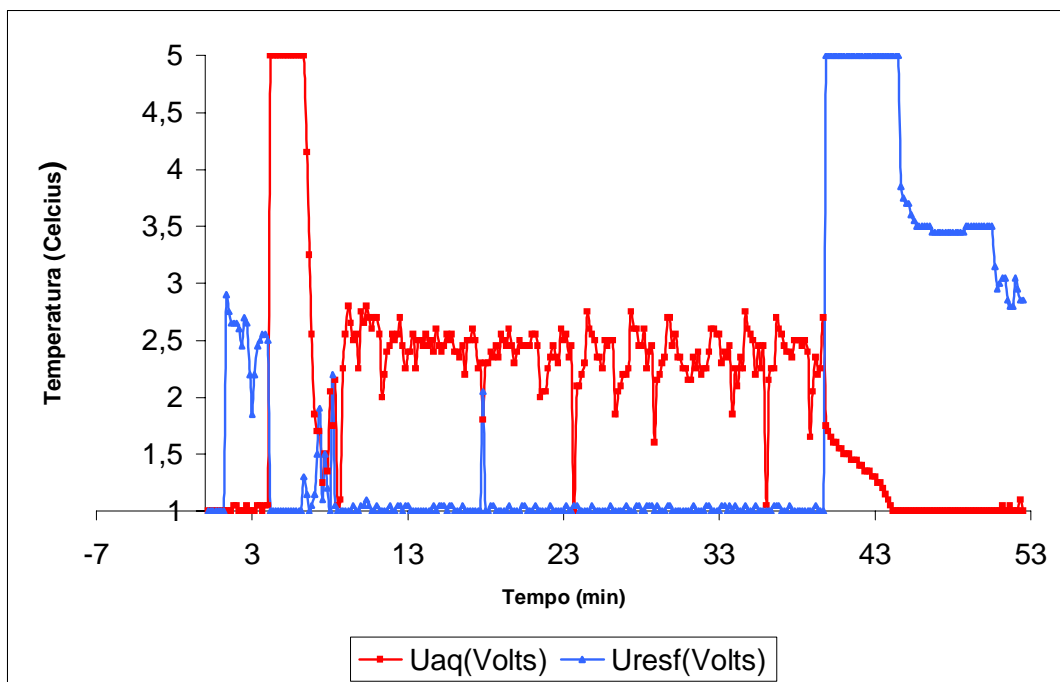


Figura 57 – Ações de controle tomadas pelo controlador 2 para o caso sem adaptação



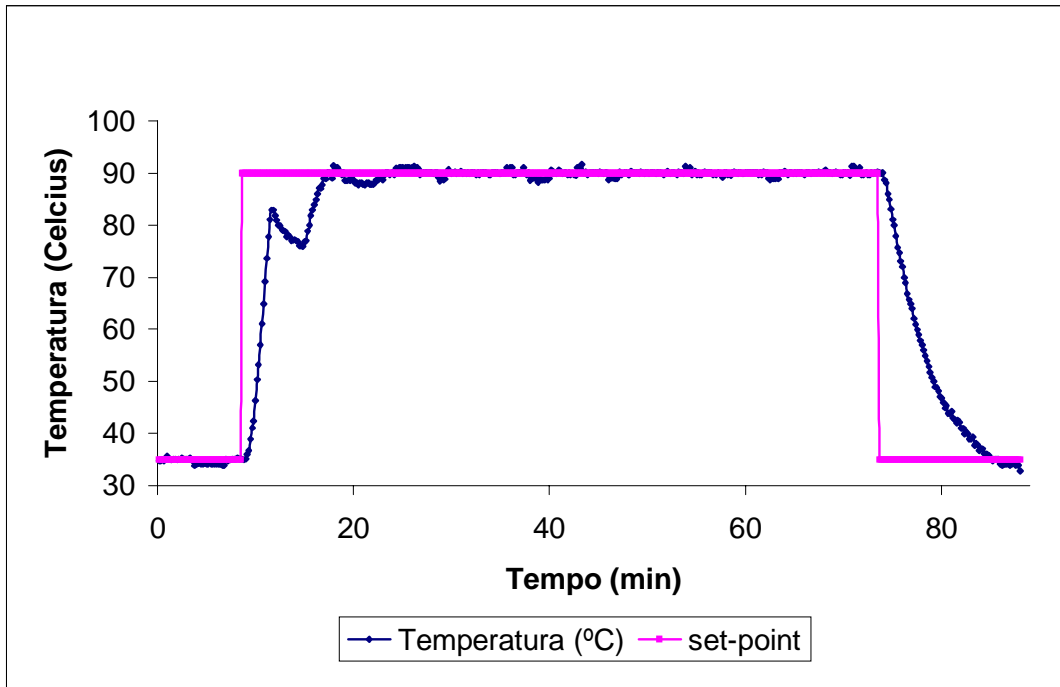


Figura 58 – Curva de resposta do controlador 2 para o caso com adaptação

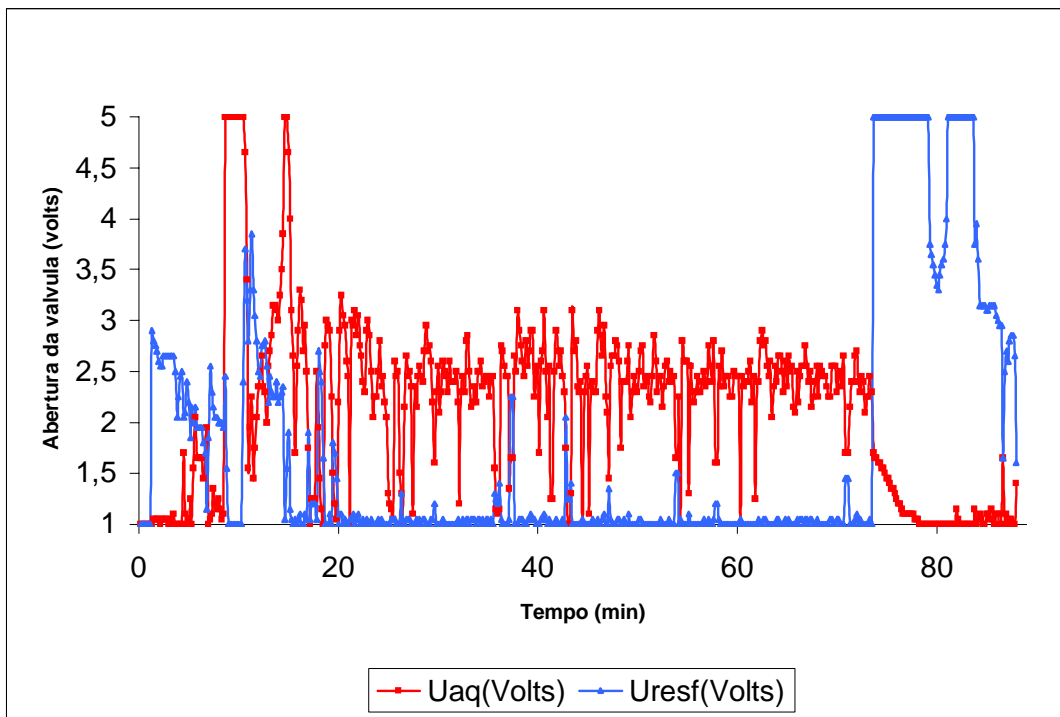


Figura 59 - Ações de controle tomadas pelo controlador 2 para o caso com adaptação

Podemos notar na **Figura 58** que o método adaptativo possibilitou que o controlador eliminasse o over-shoot ocorrido com o MPC sem adaptação (**Figura 56**). O experimento sem adaptação durou 314 amostras, cerca de 52,3 minutos. O experimento com adaptação (**Figura 58**) durou 527 amostras, cerca de 87,8 minutos.

Em seguida foi conduzido um experimento com reação química de produção de poliestireno em suspensão, com *holdup* de 0,3. Este experimento foi conduzido para os parâmetros do controlador 2 sem adaptação por GA, onde poderemos notar que a mudança na dinâmica do processo em função da reação química que transcorre em suspensão aquosa faz com que o controlador não tenha um desempenho muito bom, causando off-set ao longo do tempo de operação. A reação química é conduzida por cerca de 6 horas para cada caso. No caso da aplicação da adaptação via GA para o experimento com reação de poliestireno, utilizou-se os mesmos parâmetros do controlador 2, e os parâmetros utilizados para o GA adaptativo foram: **Janela de Pontos = 10, Penalização da Variação de pesos = 0,001, Número de iterações = 110, tamanho da População = 30**. Os resultados obtidos para os experimentos com reação de polimerização de poliestireno são mostrados nas **Figuras 60 e 61** para o caso sem adaptação e nas **Figuras 62 e 63** para o caso com adaptação por Algoritmo Genético.

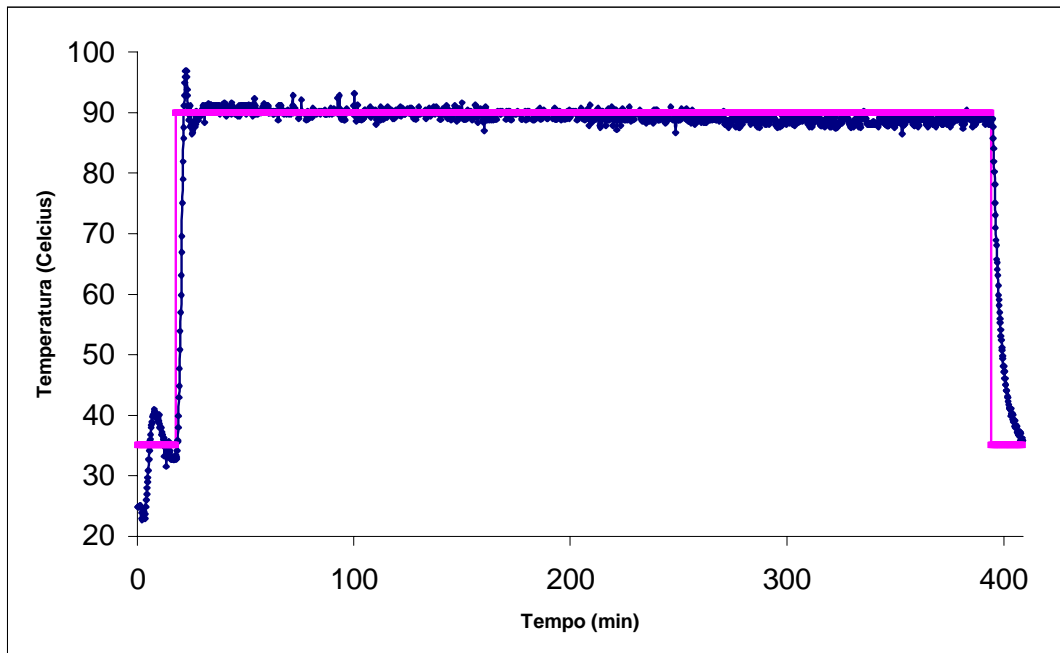


Figura 60 – Curva de resposta para o controlador 2 em experimento com reação de polimerização sem adaptação, somente utilizando a RNA recorrente como modelo do MPC

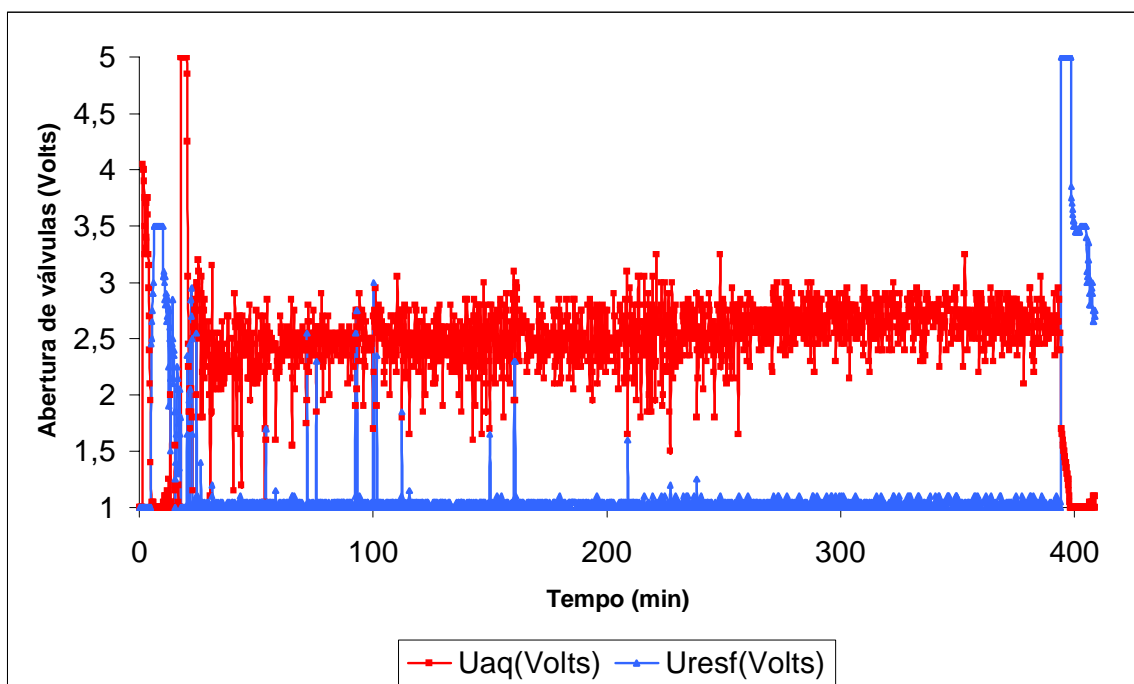


Figura 61– Ações de controle para o controlador 2 sem adaptação com reação química

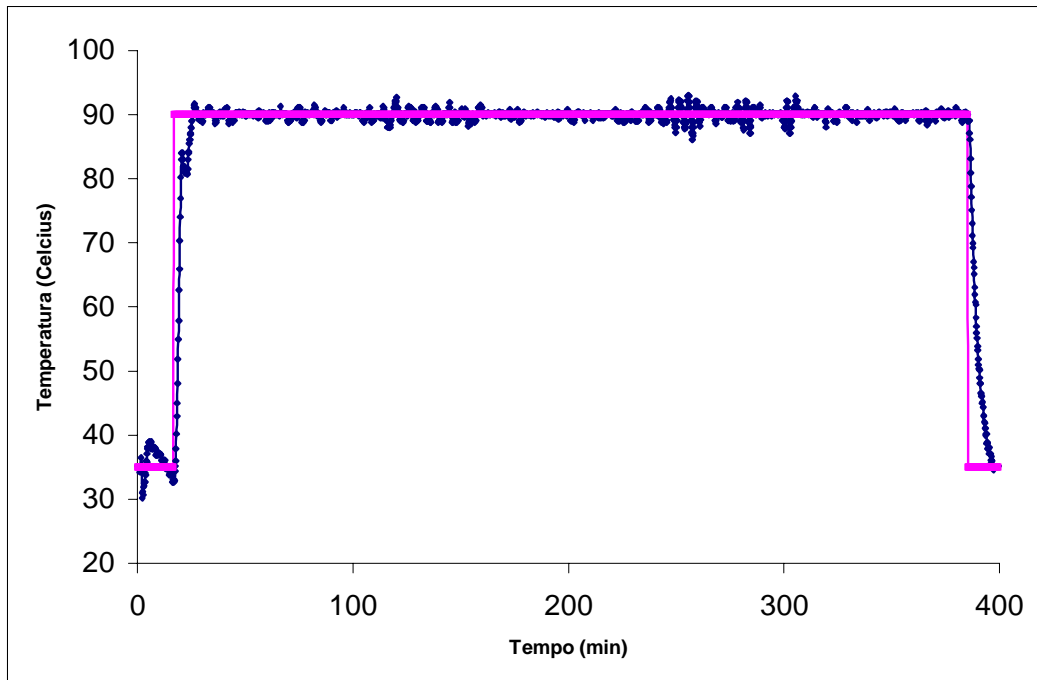


Figura 62 – Curva de resposta do controlador 2 para experimento com reação química e com adaptação por GA.

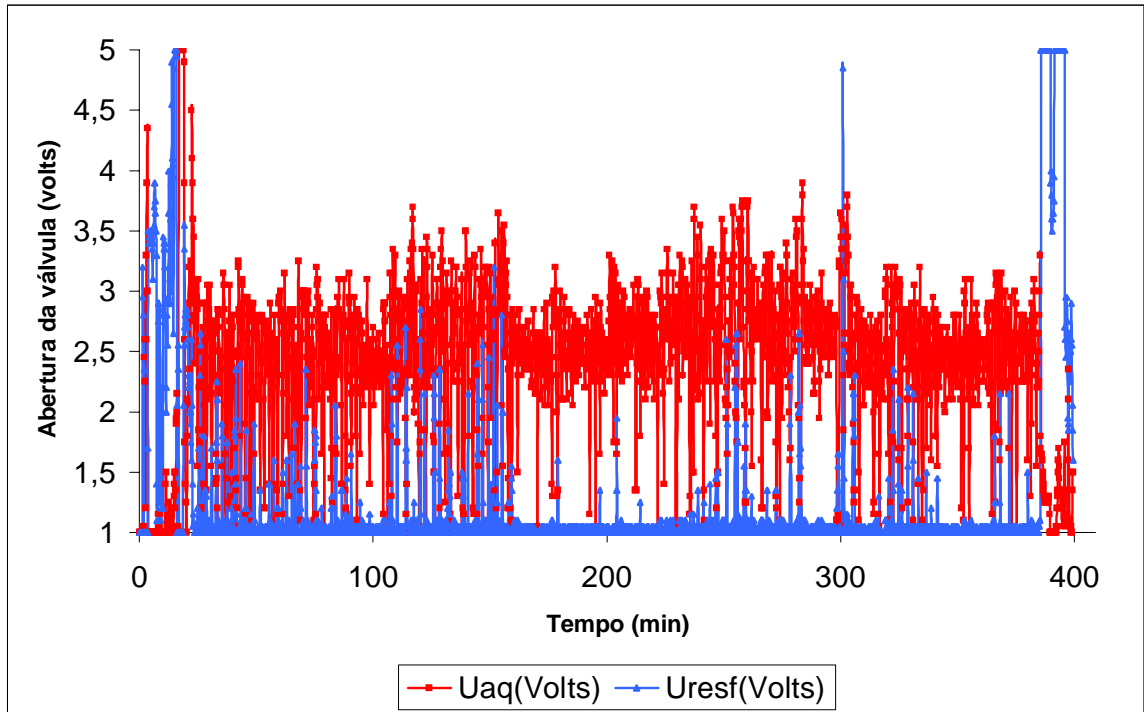


Figura 63 – Ações de controle para experimento com reação química e com adaptação por GA para o controlador 2

Podemos observar pela **Figura 60** que a aplicação do controlador MPC sem o método adaptativo e somente utilizando a RNA recorrente como modelo para o processo, apresenta um bom resultado. Porém, ocorre um *overshoot* que não é desejável, pois pode comprometer a qualidade do produto (poliestireno), bem como ao longo do transcorrer da reação química, observa-se a ocorrência de off-set, mas de modo geral a resposta é satisfatória. A ocorrência de off-set e o *overshoot* podem ser entendidos como desvios da dinâmica do processo em relação ao treinamento da rede neural, onde com reação química temos liberação de calor e ao invés de somente água temos uma suspensão de estireno/água, onde podemos entender que ocorre um comportamento dinâmico diferente do que foi utilizado para treinar a rede originalmente e a RNA não possui informação a respeito destas alterações ou perturbações ao qual o sistema está sujeito durante a reação química. O experimento em questão durou 2452 amostras ou 6,8 horas.

Contudo, observando a **Figura 62** vemos que a metodologia de adaptação dos pesos e bias da camada de saída da RNA apresenta-se como uma excelente estratégia a fim de corrigir estes desvios de modelagem da RNA frente a estas mudanças de dinâmica do sistema ou simplesmente perturbações inerentes a reação química efetuada. A respeito da aplicabilidade este tipo de controlador para tratar problemas com dinâmica variável e tempo morto de difícil avaliação fica demonstrada. O experimento durou 2396 amostras ou 6,6 horas.

Podemos observar que além das diferenças encontradas na unidade experimental, algumas considerações foram tomadas um pouco diferentes das de *CANCELIER(2004)* para a condução deste trabalho, como já citado, o tamanho da população utilizado para treinamento da RNA (off-line), o número de gerações (bem maior – 40000), sem contar que foi utilizado outro conjunto de dados para treinamento da RNA onde não se levou em conta comportamento split-range das válvulas de atuação, deixando-as livres já que se trata de uma estratégia de controle MISO (múltiplas entradas e uma saída). Outro fator relevante é que o microcomputador onde foi realizada a aplicação deste trabalho foi um Pentium-S 200 MHZ e conforme descrito no trabalho de *CANCELIER (2004)* utilizou um Pentium III 500Mhz, que tinha mais recursos computacionais, maior velocidade e capacidade de processamento, mesmo assim o controlador se apresentou com atuação satisfatória.

O fato de se utilizar valores para a penalização pela trajetória de referência maiores que os de Cancelier e uma maior penalização do consumo de vapor, bem como

um horizonte de predição menor (7 intervalos de amostragem), pode-se dizer que se deve ao fato de que as alterações físicas na planta experimental fizeram com que a dinâmica do sistema seja um pouco diferente da original e estes foram os parâmetros que satisfizeram a atuação do controlador MPC para a RNA utilizada como modelo para predição.

Outro fato relevante é que o controlador apresentou um desempenho satisfatório em uma faixa de transição de temperatura maior que a testada por CANCELIER (2004) onde o degrau para 90 graus celcius neste trabalho foi a partir de 35 graus celcius e no trabalho de cancelier foi a partir de 50 – 60 graus celcius.

Por outro lado seria de se esperar que a condução do experimento com adaptação por GA resultasse em uma resposta mais rápida que o MPC sem adaptação, o que não ocorreu. Isto pode se dar ao fato de que para a RNA utilizada não se pôde encontrar no método de adaptação uma faixa de variação de pesos maior, talvez também devido a limitação computacional; o que pode ter limitado um pouco a aplicação do método, mas ele apresentou uma resposta muito melhor que o MPC sem adaptação como se pôde observar.

De maneira geral, o controlador MPC atendeu as expectativas e a metodologia de treinamento off-line da RNA utilizando um algoritmo híbrido utilizando algoritmos genéticos no treinamento dos pesos dos neurônios e bias da camada intermediária e mínimos quadrados para o treinamento dos pesos e bias da camada de saída da RNA bem como a aplicação da adaptação on-line via algoritmos genéticos para atualizar os pesos e bias dos neurônios da camada de saída da rede puderam ser validadas satisfatoriamente na planta experimental.

Podemos observar na **Figura 64** que o controlador teve boa atuação conduzindo reação química de polimerização mesmo com elevadas **perturbações** decorrentes de ruído (variante entre 3 a 10 unidades de temperatura). Apesar do ruído elevado, o controlador adaptativo pôde suportar bem a perturbação e manter a transição de set-point bem como a manutenção da temperatura de referência. O ruído pode ser um problema em instalações industriais, constante, intermitente ou ocasional, e um controlador que possa suportar este tipo de perturbação é interessante. A **Figura 65** apresenta as ações de controle para o caso com ruído de sinal(controlador 2). O experimento durou 2447 amostras ou 6,8 horas.

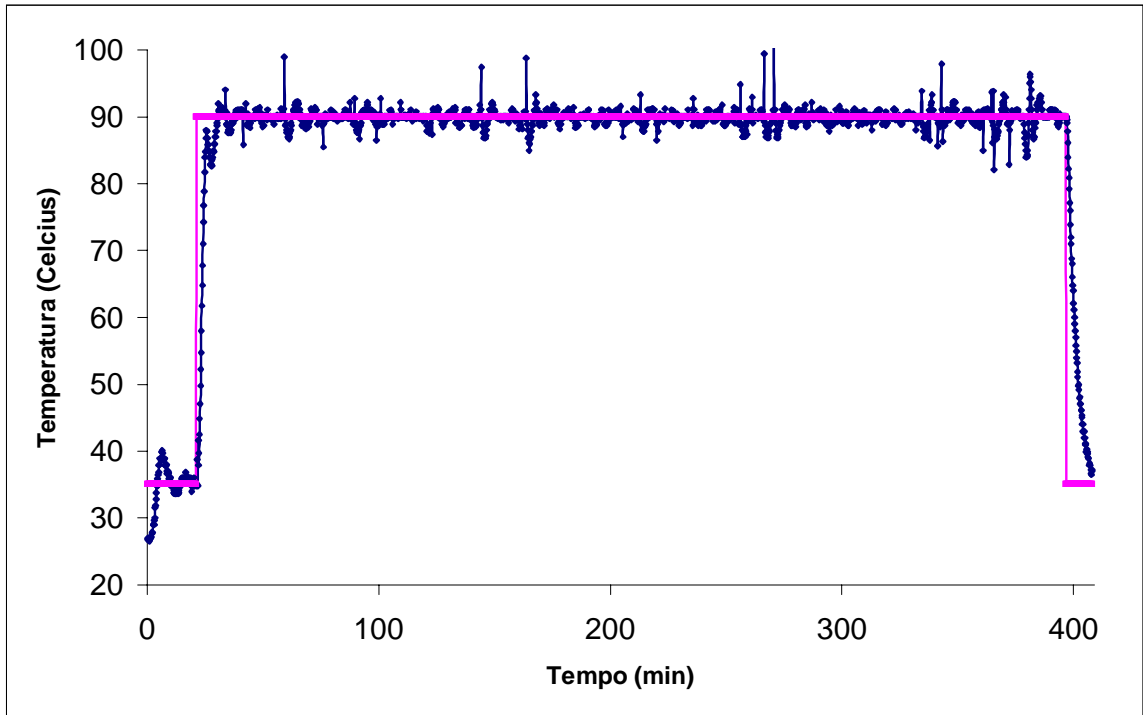


Figura 64- Curva de resposta para o caso com ruído

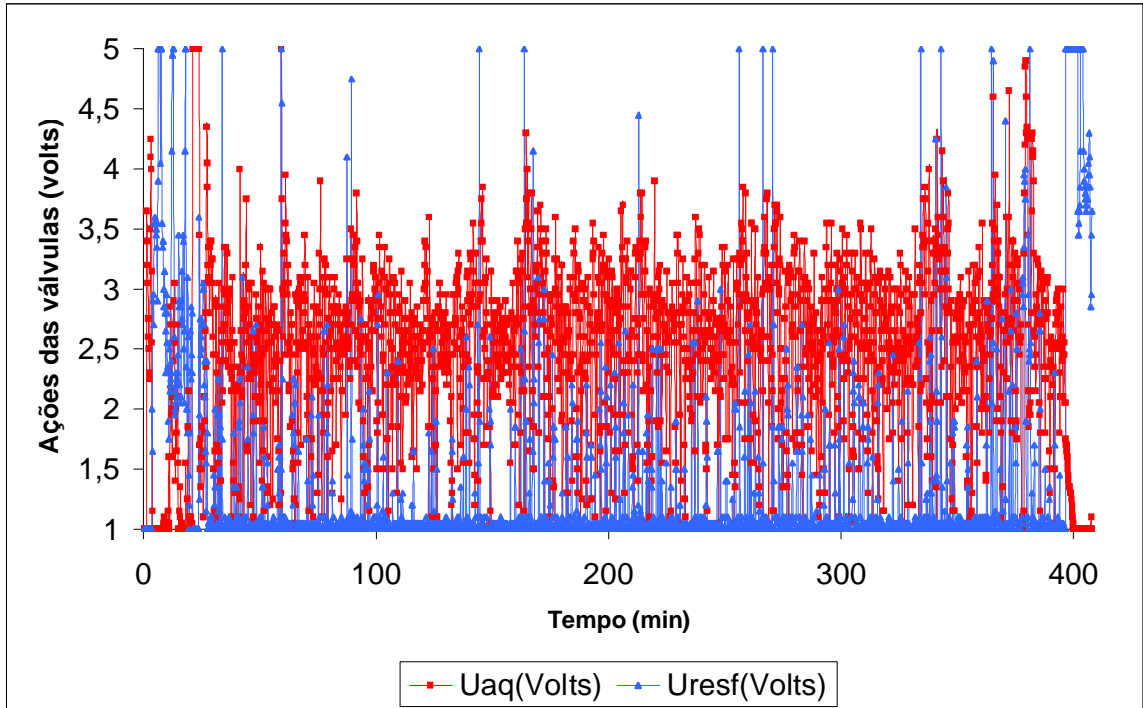


Figura 65- Ações de controle para o caso com ruído

A fim de demonstrar a atuação do método adaptativo sobre o modelo empírico utilizado (RNA Feedforward) no controlador MPC, apresenta-se a variação do valor da função objetivo do controlador a qual é minimizada pelo Algoritmo Genético, a **Figura 66** apresenta um comparativo entre a função objetivo da rede original (sem adaptação) e a rede adaptada por algoritmos genéticos para o caso do **controlador 2** conduzindo reação química de polimerização com utilização do método adaptativo (sem ruído). Nota-se que o método atuou muito bem mesmo com a taxa de variação dos pesos adaptados pequena (0.001) minimizando a função objetivo do controlador e como já mostrado mantendo a a temperatura satisfatoriamente na referência.

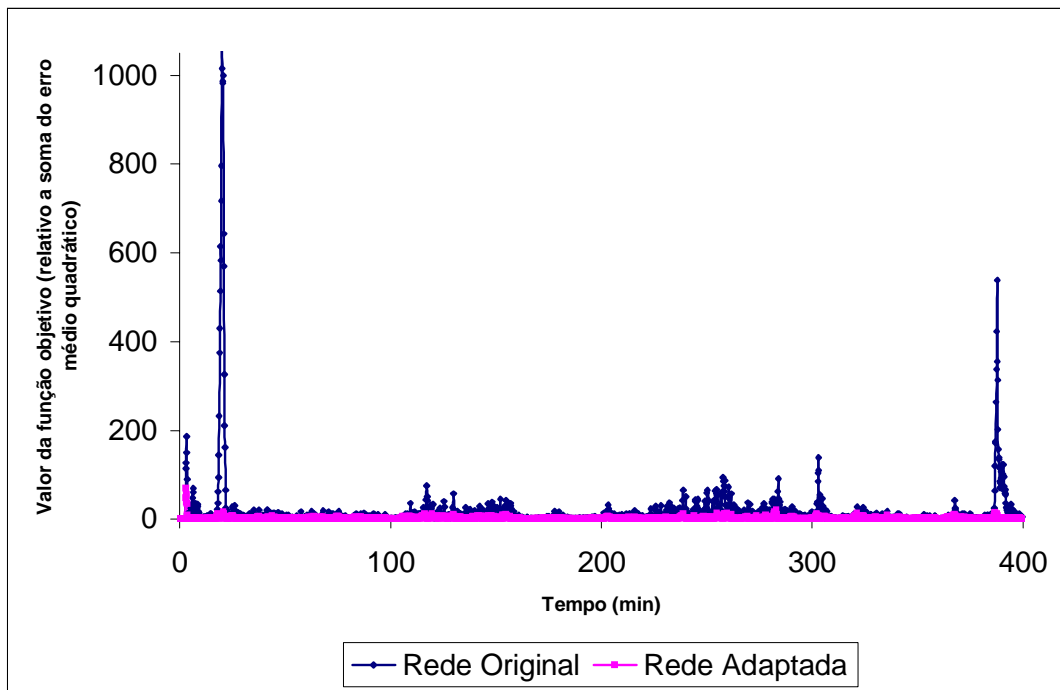


Figura 66 - Variação do valor da Função objetivo para experimento com adaptação por GA

Podemos ver na **Figura 66** que o método adaptativo mantém o valor da função objetivo do controlador (soma do erro médio quadrático) numa faixa bastante inferior que os valores da rede original (não adaptada). Também podemos observar que a rede adaptada apresenta menor variação dos valores da função objetivo que a rede à parâmetros fixos (não adaptada), o que garante robustez ao sistema de controle.

Cabe registrar que o comportamento do valor da função objetivo para os outros experimentos apresentados com adaptação foi similar ao apresentado e os resultados foram suprimidos para evitar repetição.



## 6. Conclusões

Este estudo foi direcionado para investigação de metodologias de ajuste de controladores, bem como aplicação destes em um sistema de polimerização objetivando avaliar a performance frente ao controle de temperatura. Uma boa performance é importante como no caso do controle de temperatura da reação de polimerização de poliestireno, pois desvios quanto a manutenção da temperatura de referência podem comprometer o produto final. Esta observação é válida para maioria dos casos encontrados na indústria.

A complexidade do processo de controle de temperatura, frente às dificuldades quanto a modelagem do sistema, tempo morto, dinâmica variável, e perturbações de ruído dentre outras que não podem ser identificadas foi tratada de maneira a explorar recursos os quais possam minimizar estes problemas de modelagem.

A tentativa de aproximar o sistema a um sistema de primeira ordem com tempo morto e aplicação de um controlador linear PID ajustado à metodologias clássicas foi investigada, não apresentando desempenho satisfatório para o caso de conduzir reação química de polimerização, onde as dificuldades implícitas ao processo demonstram ter um peso grande na performance do controlador, donde a necessidade de uma metodologia de modelagem mais acurada que possa levar em consideração estas não-linearidades, e variações do sistema.

É importante registrar a característica das válvulas de controle, tipo igual percentagem abre falha fecha, que representa uma válvula com comportamento não-linear, boa para baixas vazões mas a medida que se aumenta a vazão o desempenho da válvula muda.

Frente esta necessidade, a aplicação de um controlador preditivo baseado em modelo é investigada, onde se propõe a utilização de um modelo empírico baseado em uma Rede Neural Feedforward a fim de representar o sistema, treinada de forma off-line e de forma recorrente (saída revista da rede é realimentada para o treinamento), e mais; a aplicação deste modelo a um controlador MPC (Model Predictive Control), a fim de se suprir o controlador de informações quanto aos tempos mortos envolvidos bem como quanto a dinâmica do processo, que são incorporados a RNA e utilizados pelo Preditor (controlador). Esta aplicação apresentou resultados satisfatórios para o controlador a parâmetros fixos, após ajuste preliminar dos parâmetros e testes com diferentes RNA's treinadas a partir de uma série temporal de dados gerados na unidade experimental, para

o sistema sem alterações, ou seja; utilizando o mesmo sistema com o qual foram gerados os padrões de treinamento da RNA. A RNA utilizada possui 4 neurônios na camada de entrada, 5 neurônios na camada intermediária e 1 neurônio na camada de saída, mais os *bias* da camada intermediária e de saída.

Contudo, para o caso com a aplicação conduzindo uma reação química de polimerização de estireno, o controlador MPC baseado em um modelo de RNA a parâmetros fixos não obteve um desempenho satisfatório.

O resultado de reação exotérmica ocorrida no reator e não modelada, representa uma alteração na dinâmica do sistema e a ocorrência de off-set ao longo da reação que também pode ser explicado por alterações na dinâmica do sistema em decorrência do transcorrer da reação química. Esta problemática traz a necessidade de se investigar uma aplicação que possa corrigir estes desvios de dinâmica e restaurar a robustez do controlador; e para tanto foi aplicada uma metodologia de ajuste on-line dos parâmetros da RNA utilizada como modelo do processo.

Esta metodologia de ajuste on-line dos parâmetros da RNA utilizada como modelo do sistema no controlador MPC objetiva a minimização da função objetivo do controlador que trata da soma do erro médio quadrático em relação ao valor predito pelo MPC e o valor real do sistema para a temperatura. A metodologia de ajuste de parâmetros recalcula os valores dos pesos e *bias* da camada de saída da RNA objetivando minimizar o valor da função objetivo (tendendo a zero), aplicando para isto uma metodologia de Algoritmos Genéticos. A estratégia foi investigada primeiramente com o controlador MPC com a unidade experimental contendo água, onde se obteve resultados satisfatórios para todos os casos estudados, controladores que apresentaram pequeno off-set, ou *overshoot* na transição de *set-point*; estes causados por algum pequeno desvio do modelo de RNA utilizada como modelo para a dinâmica real do sistema ou por perturbações não modeladas; tiveram um desempenho visivelmente melhorado com a utilização do processo de adaptação on-line. Com isto se passou a aplicação para o caso com reação química, onde o controlador com adaptação on-line por GA obteve um resultado melhor que o controlador a parâmetros fixos tanto para transição do *set-point* (eliminando *over-shoot*), quanto para manutenção da referência eliminado off-set. Portanto a aplicação do método adaptativo apresentou-se satisfatória para os casos estudados.

Pode-se observar que o método de adaptação dos parâmetros realizou a manutenção dos valores da função objetivo do controlador satisfatoriamente, mantendo-

os em uma faixa de variação pequena e bem mais próximos de zero que os valores calculados para o controlador a parâmetros fixos.

Também pode-se concluir com este trabalho que todas as etapas envolvidas têm a mesma importância, desde a identificação dos tempos mortos envolvidos no sistema até a formação dos padrões de treinamento da RNA e sua aplicação no MPC. A aplicação do MPC necessita ajuste de alguns parâmetros ligados a lei de controle do mesmo e não existe regra pra tal ajuste, as penalizações de variação das ações de controle, o horizonte de predição que não deve ser muito elevado a fim de não deteriorar o modelo utilizado, a penalização quanto a trajetória de referência, que esta ligada ao tempo de resposta do controlador a transições de *set-point*, a penalização do consumo de vapor, que esta ligada ao consumo de vapor e também de água fria, pois se este parâmetro é “liberado”, ou seja, quanto menor, mais consumo de vapor se terá, com isto a válvula de água fria vai trabalhar mais também; e todos estes parâmetros apesar de interligados não possuem regra de ajuste, devem ser ajustados intuitivamente e seu valor dependerá o sistema em questão, dos limites de operação e da eficiência desejados. Mesmo com uma RNA utilizada no modelo do controlador muito bem treinada, um ajuste ruim destes parâmetros pode comprometer o controlador, gerando resultados não satisfatórios, ou seja, o controlador pode não “controlar” o processo para o qual esta sendo aplicado. Com isto é necessário uma série de testes preliminares a fim de se determinar o conjunto de parâmetros a serem utilizados pelo controlador; lembrando que estes também vão depender do modelo (no caso RNA) que esta sendo utilizado.

Ainda a ocorrência de ruído elevado no sinal de temperatura, o que poderia desestabilizar o controlador mas conforme apresentado não ocorreu, sendo que estes desvios foram absorvidos pela adaptação *on-line* da RNA, permitindo ao controlador desempenho satisfatório frente a ruído elevado no sinal de referência.

Contudo, a aplicação do controlador MPC em um sistema onde não há estados estacionários, com não-linearidades e ocorrência de perturbações; com aplicação de adaptação *on-line* (dinâmica) apresentou-se como bom como preditor de estados futuros do processo, com desempenho satisfatório.

## 7. Sugestões

Pode-se proceder a um estudo mais aprofundado das metodologias aplicadas neste trabalho no tocante aos seguintes itens:

- a) Quanto à metodologia de modelagem: pode ser investigada uma estratégia de treinamento da RNA de maneira a fornecer resultados tão bons quanto a utilizada de maneira a convergir mais rapidamente, ou pode ser investigada outra maneira de modelar o sistema para aplicação junto ao preditor;
- b) Quanto ao controlador utilizado: pode-se investigar melhor a lei de controle utilizada nesta aplicação a fim de se encontrar uma expressão que possibilite um número de parâmetros menor a serem ajustados manualmente.
- c) Quanto a metodologia de ajuste de parâmetros: Pode-se investigar outras estratégias de ajuste de parâmetros da RNA (ou modelo utilizado), ou utilizando mais recursos computacionais e se ampliando os limites utilizados pelo Algoritmo Genético. Ou então aplicando-se estudos sobre estas metodologias a fim de se propor um algoritmo com maior velocidade de convergência ou que possa atualizar mais parâmetros do controlador e não somente os pesos e bias da camada de saída da rede neural, ou que trabalhe juntamente com um conjunto de regras pré-programados referentes ao processo que possam determinar melhor o momento da atualização dos parâmetros e não que os mesmos sejam atualizados a cada intervalo de amostragem mas sim atualizados caso uma determinada condição, além da minimização da soma do erro quadrático somente; seja satisfeita.
- d) A aplicação e investigação do desempenho do controlador estudado em outros sistemas também se apresenta como um estudo interessante.

## 8. Bibliografia

ABOU-JEYAB R.A., GUPTA Y.P., GERVAIS J. R., BRANCHI P.A., WOO S.S. **Constrained multivariable control of a distillation column using a simplified model predictive control algorithm.** Journal Process Control, 2001, Vol. 11, pg. 509-517.

AGUIRRE, LUIS ANTONIO. **Introdução à identificação de Sistemas: técnicas lineares e não-lineares aplicadas a sistemas reais.** Ed. UFMG – BH – 2000.

ASTRÖM, K.J. , HÄGGLUND T. **The future of PID Control.** Control Engineering Practice, 2001, Vol. 9, pg. 1163-1175.

ASTRÖM, KARL J. Hägglund, TORE. **PID controllers, teory, design, and tunning.** 2<sup>nd</sup> ed. Instrument Society of America. 1995.

ASTRÖM, KARL J. ; WITTENMARK, B. **Computer Controlled Systems.** Ed. Prentice Hall, Inc. – 1984.

BARCZAK, CZESLAU L. **Controle Digital de Sistemas Dinâmicos: Projeto e Análise.** Ed. Edgard Blücher LTDA – 1995.

BITTENCOURT, GUILHERME. **Inteligência Artificial: ferramentas e teorias.** 2 ed.- Ed. Da UFSC – Florianópolis – SC. 2001.

BLOEMEN, H.H.J. , CHOU C. T., van den BOOM T.J.J., VERDULT V. , VERHAEGEN M. BACKX T. C. **Winer model indentification and predictive control for dual composition control of a distillation column.** Journal Process Control, 2001, Vol 11, pg. 601-620.

CAIRO LÚCIO NASCIMENTO JR. ; TAKASHI YONEYAMA. **Inteligência Artificial em Controle e Automação.** Ed. Edgard Blücher LTDA. 2002.

CAMACHO, E. F.; BORDONS, C. **Model predictive control.** London: Springer, 1998. 280p.

CANCELIER, A. **Controle preditivo de reatores semi-batelada**. 1997. Dissertação (Mestrado em engenharia química) - Departamento de Engenharia Química e Engenharia de Alimentos, Universidade Federal de Santa Catarina - UFSC, Florianópolis.

CANCELIER, A. **Otimização e Controle do Processo de Produção do Poliestireno em Suspensão**. 2004. Tese de Doutorado. Departamento de Engenharia de Produção, universidade federal de Santa Catarina – UFSC, Florianópolis.

CANTÚ, MARCO. **Dominando o Delphi 5 “A Bíblia”**. Ed. Person Education do Brasil – SP – 2000.

CARDOSO. LUIZ CLÁUDIO DOS SANTOS. **Logística do Petróleo: Transporte e Armazenamento**. Ed. Interciência – RJ – 2004.

CHAMPMAN, STEPHEN J. **Programação em Matlab para engenheiros**. Ed. Pioneira Thomson Learning – 2003.

CLAUMANN, C. A. **Modelagem dinâmica e controle de processos não lineares: Uma aplicação de algoritmos genéticos para treinamento de redes neurais recorrentes**. 1999. Dissertação (Mestrado em engenharia química) - Departamento de Engenharia Química, Universidade Federal de Santa Catarina - UFSC, Florianópolis.

CORRÊA, OTON LUIZ SILVA. **Petróleo: noções sobre exploração, perfuração, produção e microbiologia**. Interciência – 2003.

CRUZ, JOSÈ JAIME da. **Controle Robusto Multivariável**. Ed. da Universidade de São Paulo – EDUSP. 1996.

DIEGUEZ, JOSÉ PAULO P. **Métodos numéricos computacionais para engenharia**. Ed. Interciência –RJ – 1992. VI e VII.

DORF, RICHARD C.; BISHOP, ROBERT H. **Modern Control Systems**. 9<sup>th</sup> ed. Prentice Hall, Inc. 2001.

EDGARD CAMPOS FURTADO, EDUARDO M. A. M. MENDES, ERIVELTON G. NEPOMUCEMO, VALCERES V. R. e SILVA. **Identificação de sistemas dinâmicos não-lineares contínuos utilizando modelos NARMAX: Estudo de caso de um forno a arco elétrico**. ANAIS do XIV Congresso Brasileiro de Automática, Natal – RN – 2002.

EMAD ALI. **Heuristic on-line tuning for non-linear predictive controllers using fuzzy logic**. Elsevier – Journal Process Control, 2003, Vol. 13, pg. 383-396.

FOUST, ALAN S. **Princípio das Operações Unitárias**. 2 ed. Editora LTC – RJ – 1982.

FOGLER, H. S. **Elements of chemical reaction engineering**. 3<sup>th</sup> ed. New Jersey: Prentice Hall, 1999. 967p.

GARCIA, ROBERTO. **Combústíveis e Combustão Industrial**. Ed. Interciência – RJ – 2002.

GOLDBERG, D. E. **Genetic Algorithms in search, optimization, and machine learning**. New York: Addison-Wesley, 1989.

GOLDEN, RICHARD M. **Mathematical methods for neural network analysis and design**. Massachusetts Institute of Technology. 1996.

GRUYITCH, L. ; RICHARD J-P. ; BORNE P. AND GENTINA J-C. **Stability Domains**. CRC Press Company/CHAPMAN & HALL – New York – 2004.

HAYKIN, SIMON. **Redes Neurais: princípios e prática**. 2ed. Porto Alegre, Ed. Bookmann. 2001.

HANSELMAN, DUANE; LITTLEFIELD, BRUCE. **Matlab 5: Versão do Estudante**. Makron Books – SP – 1999.

HENSON, MICHEL A. ; SEBORG, DALE E. **Nonlinear process control**. Prentice – Hall – New Jersey – 1997.

INGIMUNDARSON, ARI ; HÄGGLUND, TORE. **Robust tuning procedures of dead-time compensating controllers**. Control Engineering Practice, 2001, Vol. 9, pg. 1195-1208.

JEAN GABRIEL GANASCIA. **Inteligência Artificial (L'intelligence artificielle)**. Ed. Lis Gráfica e Editora LTDA – 1997.

KAYA, IBRAHIM. **Obtaining controller parameters for a new PI-PD Smith predictor using autotuning**. Journal Process Control, 2003, Vol. 13, pg.465-472.

LYUBEN, William L. **Process modeling, simulation, and control for chemical engineers**. 2<sup>nd</sup> ed. McGraw-Hill Book Co.1990.

MACHADO, R. A. F. **Otimização e Controle de Reatores de Polimerização em Batelada**. 1996. Dissertação (Mestrado em engenharia química) - Departamento de Engenharia Química, Universidade Federal de Santa Catarina - UFSC, Florianópolis.

MACHADO, R. A. F. **Modelagem e Simulação da Distribuição do Tamanho das Partículas em Sistemas de Polimerização em Suspensão**. 2000. Tese (Doutorado em engenharia química) - PEQ/COPPE/UFRJ, Rio de Janeiro – Brasil.

MARLIN, THOMAS E. **Process Control: designing processes and control systems for dynamic performance**. 2<sup>nd</sup>. ed. McGraw-Hill Inc. 2000.

MATOS, MARIVALDO P. **Séries e Equações Diferenciais**. SP/P. Hall – 2001.



MAZZUCCO, M. M. **Implementação de um controlador preditivo baseado em um modelo neural associado a um sistema especialista.** 1996. Dissertação (Mestrado em engenharia química) - Departamento de Engenharia Química e Engenharia de Alimentos, Universidade Federal de Santa Catarina - UFSC, Florianópolis.

MAZZUCCO, M. M. **Um sistema difuso para o controle de temperatura de unidades de processamento em batelada,** 2003. 297f. Tese (Doutorado em Engenharia de Produção) - Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina - UFSC. Florianópolis.

OGATA, KATSUHIKO. **Engenharia de Controle Moderno.** Editora Prentice-Hall do Brasil – RJ – 1998.

OGUNNAIKE, BABATUNE A. **Process dynamics, modeling, and control.** Oxford University Press, Inc. 1994.

RUSSEL, STUART J. **Inteligência Artificial : tradução da segunda edição / Stuart Russel, Peter Norving.** Elsevier Editora Ltda. – RJ – 2004.

SEBORG, D. E.; EDGAR, T. F.; MELLICHAMP, D. A. **Process dynamics and control.** New York: John Wiley & Sons, 1989.

SHINSKEY, F.G. **PID-deadtime control of distributed processes.** Control Engineering Practice, 2001, Vol. 9, pg. 1177-1183.

WALTER GANDER; JIRÍ HREBÌCEK. **Como Resolver Problemas em Computação Científica Usando Maple® e Matlab®.** Tradução da Terceira Edição Alemã. Ed. Edgard Blücher LTDA. 1997.

YOUSHEN XIA, JUN WANG. **A Discrete-Time Recurrent Neural Network for Shortest-Path Routing.** IEEE Transactions on Automatic Control, 2000, Vol. 45, nº11, pg. 2129-2134.

## **9. Anexos & Apêndices**

# APÊNDICE A

Descrição do método de Identificação de Sundaesan para o caso Sobreamortecido

## O caso sobreamortecido ( $\xi > 1$ )

### Exemplo 1

Neste ponto vamos tratar de um sistema com resposta do tipo sobre amortecido

com função transferência do tipo  $H(s) = \frac{1}{s^2 + 4s + 1}$ , esta função não pode ser dividida

em uma função do tipo  $H(s) = \frac{e^{-\tau_d s}}{(\tau_1 s + 1)(\tau_2 s + 1)}$  mas poderia ser aproximada a uma do

tipo  $H(s) = \frac{1}{\{(\tau_1 s + 1)(\tau_2 s + 3)\} - 2}$ . Vamos utilizá-la inicialmente para testar a

flexibilidade do método apresentado a sistemas com pequenas diferenças das funções que se leva em consideração para mostrar que mesmo assim pode-se obter uma boa aproximação para o sistema.

Utilizando o Simulink™ podemos obter a curva de resposta para o sistema a ser analisado; como mostrado no diagrama abaixo.

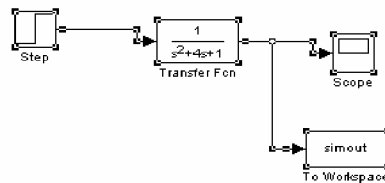


Diagrama A.1 – Esquema do Simulink™ para obtenção da Figura A.1

Esta função de transferência fornece uma resposta em malha aberta a um degrau unitário como apresentado na **Figura A.1**.

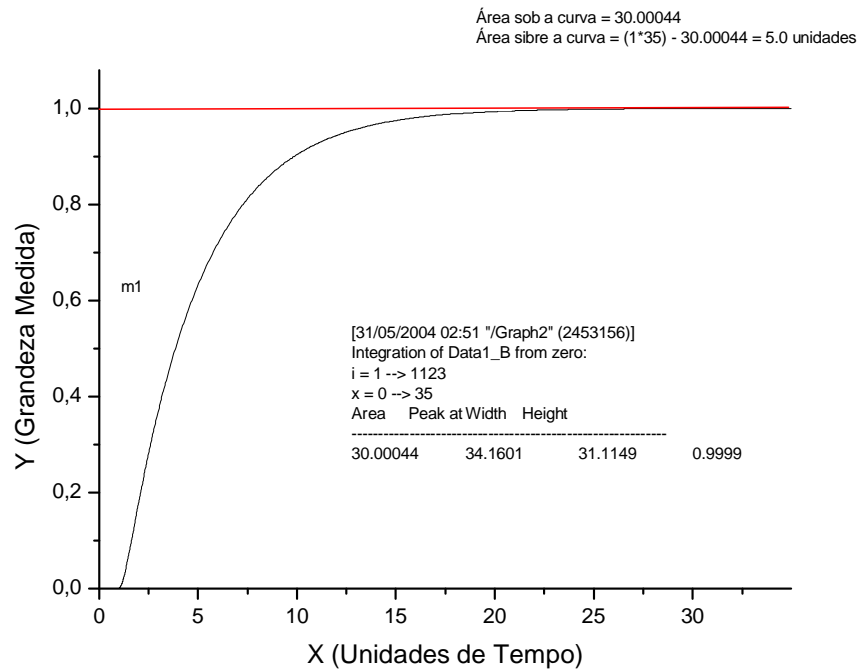


Figura A.1 – Resposta do sistema a um degrau unitário

Utilizando a definição de integral podemos obter a área sob a curva de resposta do sistema, sendo que nos interessa a área **sobre** a curva ( $m_1$ ), que seria a subtração da área do retângulo de dimensões ( $x = 35$ ,  $y = 1$ ), ou seja 35 unidades de área, da área sob a curva ou seja, a integral definida no intervalo de 0 a 35 da aproximação polinomial à curva de resposta que representaria a função a ser integrada. Este procedimento foi realizado utilizando-se o software Origin™, e obteve-se:

**[31/05/2004 02:58 "/>**

**Integration of Data1\_B from zero:**

**i = 1 --> 1123**

**x = 0 --> 35**

**Area Peak at Width Height**

-----  
**30.00044 34.1601 31.1149 0.9999**

Com 1123 pontos obtidos através do Simulink™ Matlab™.

A área **sobre** a curva encontrada foi de **5 unidades de área**, tomando-se a área total do retângulo de base 35 unidades e altura 1 unidade e subtraindo-se esta da área sob a curva já calculada de 30 unidades.

Traçando uma reta visualmente **tangente** ao ponto de **inflexão** da curva de resposta podemos obter a sua **inclinação** e com isto evoluir no exemplo.

Temos como mostra a **Figura A.2** o procedimento de se traçar a reta tangente ao ponto de inflexão, e com isto através do software Origin™ podemos obter um par de pontos desta reta e com isto calcular a equação da reta com seus respectivos coeficientes angular (inclinação da reta) e linear.

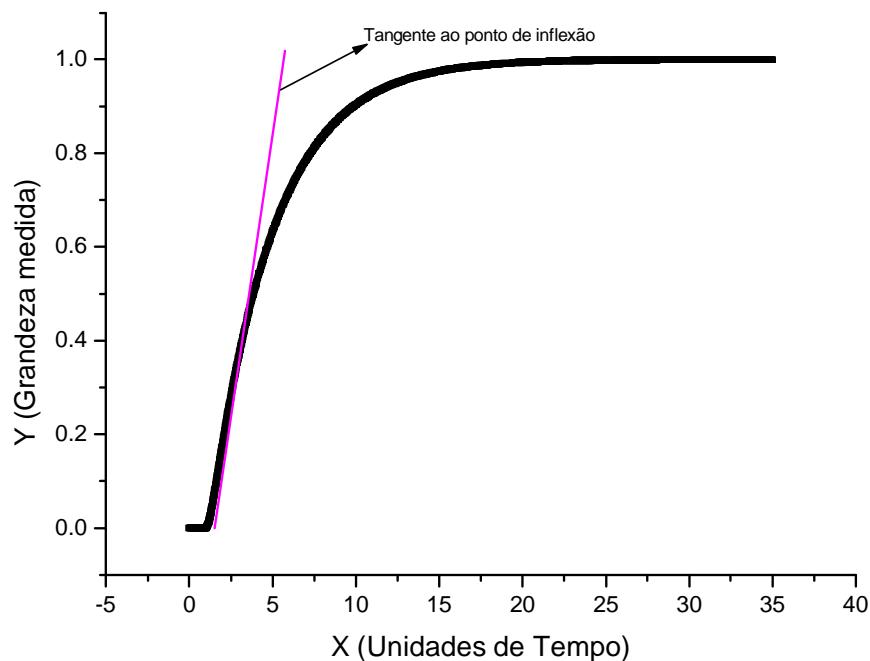


Figura A.2 – Reta tangente ao ponto de inflexão da curva de resposta do sistema

Com isto podemos obter a equação da reta tangente e os parâmetros A e B já que  $Y = A + B * X$ ; temos como mostra a **Figura A.3**.

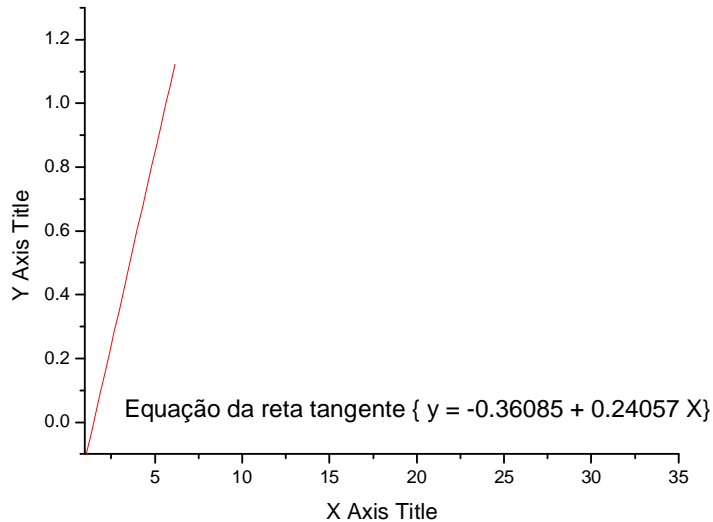


Figura A.3 – Regressão linear utilizando o software Origin™ para a reta tangente

Com isto podemos seguir os passos para desenvolver o método e temos:

- 1 Determinar o **ganho em regime permanente**, que é a variação do sinal de saída pela amplitude do degrau aplicado na entrada; é igual a **1** neste caso.
- 2 Determinar a área **sobre a curva**  $m_1 = 5$  **unidades de área**.
- 3 Determinar a **inclinação** da reta tangente ao ponto de inflexão do sinal de saída. Este valor é chamado **Mi** e é igual a **0,24057**.
- 4 Determinar  $t_m$ , que é o valor de **X** na reta tangente quando **Y = 1** (no caso);  $t_m = 5,656$ ; ou seja, a interseção da reta tangente com o valor em  $y(t)$  em regime permanente.
- 5 Determinar  $\lambda = (t_m - m_1) * Mi$ ; temos,  $\lambda = (5,656 - 5) * 0,24057 = 0,158$ .
- 6 Determinar  $\eta$  a partir do gráfico de  $\lambda = \chi e^{-\chi}$ , sendo que  $\chi = \ln \eta / (\eta - 1)$ . Temos que  $\lambda = \{\ln \eta / (\eta - 1)\} * (e^{-\ln \eta / (\eta - 1)})$ ; a solução desta equação nos fornece o valor de  $\eta$  para o valor já encontrado de  $\lambda$ .

Seguindo-se estes passos podemos agora calcular os parâmetros.

O valor de  $\eta$  pode ser calculado utilizando um programa desenvolvido em Delphi™ a partir do valor de  $\lambda$  e o valor encontrado é de **0,15**, como mostra a **Figura A.4**. O Código do programa pode ser encontrado no **Apêndice F**.

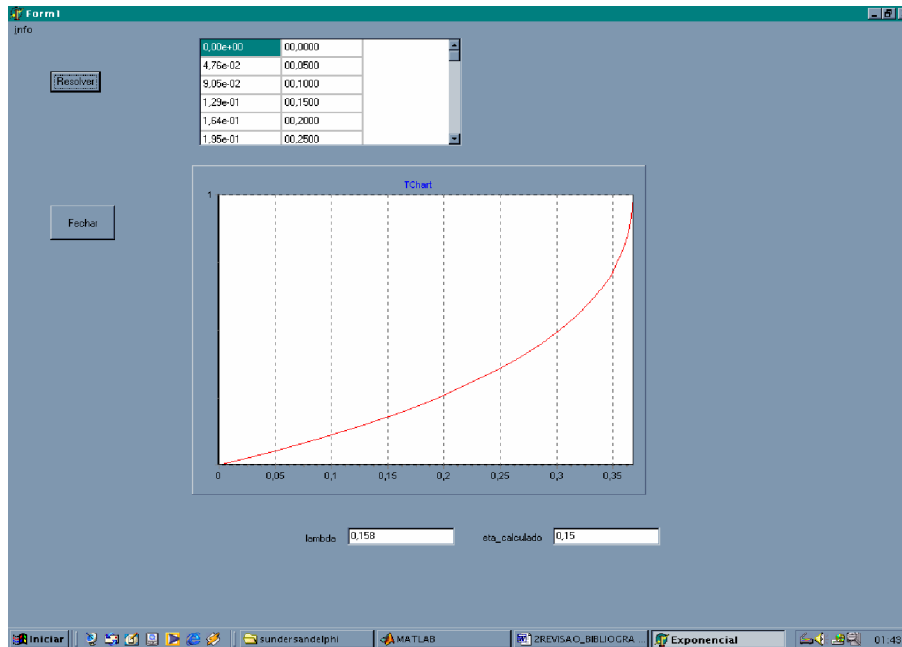


Figura A.4 – Determinação do valor de  $\eta$

Com o valor de  $\eta = 0,15$  podemos calcular os valores dos parâmetros para estimar a função transferência para o sistema em questão.

Procedemos como segue:

$$\tau_1 = \frac{\eta}{Mi^{1-\eta}}$$

$$\tau_2 = \frac{1}{Mi^{1-\eta}}$$

$$\tau_d = m_1 - \tau_1 - \tau_2$$

Com isto obtemos valores para os parâmetros:  $\tau_1 = 2,974$  ;  $\tau_2 = 0,446$  e  $\tau_d = 1,58$ ; vemos que representa uma boa aproximação da função transferência original sendo que a comparação entre as duas, a estimada e a real (simulada) segue na **Figura A.5** a seguir.

A função obtida apresenta resposta próxima da original:

$$H(s) = \frac{e^{-1,58s}}{(2,974s + 1)(0,446s + 1)} = H(s) = \frac{e^{-1,58s}}{(1,326s^2 + 3,42s + 1)}$$

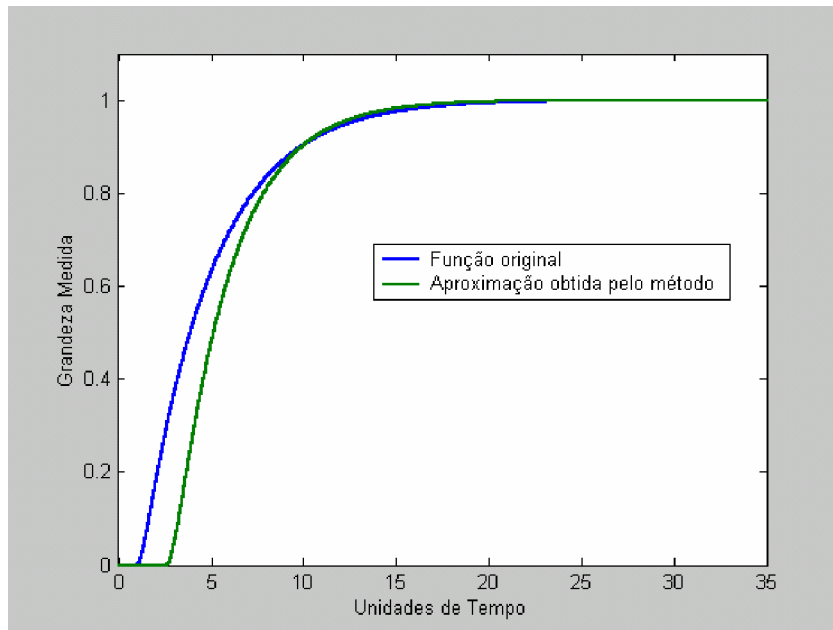


Figura A.5 – Comparação entre a função original e a estimada pelo método

### Exemplo 2

Agora utilizando uma função transferência do tipo exato de

$$H(s) = \frac{e^{-\tau_d s}}{(\tau_1 s + 1)(\tau_2 s + 1)}; \text{ e vamos considerar uma função com } \tau_1=1, \tau_2=1,5 \text{ e } \tau_d=4;$$

sendo:

$$H(s) = \frac{e^{-4s}}{(1s + 1)(1,5s + 1)} = H(s) = \frac{e^{-4s}}{(1,5s^2 + 2,5s + 1)}.$$

Esta função transferência fornece uma resposta a um degrau unitário em malha aberta do tipo da **Figura A.6** a seguir.



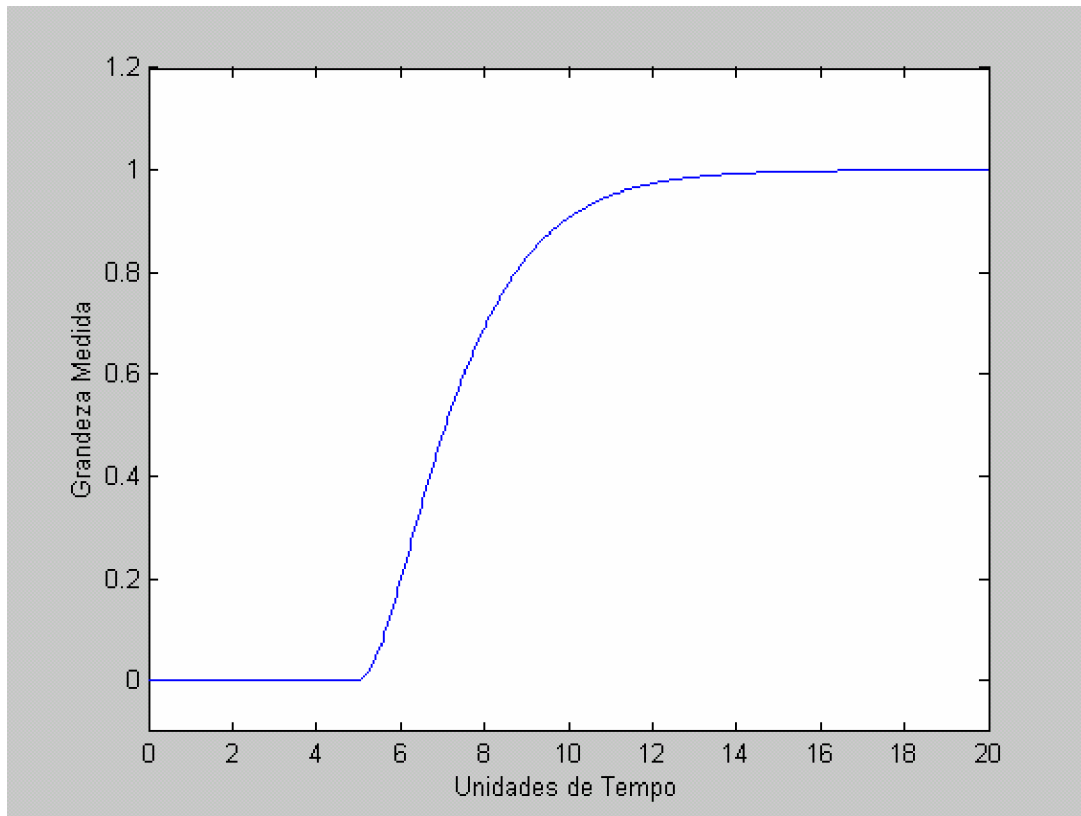


Figura A.6 – Resposta do sistema a um degrau unitário

Note que esta é a resposta típica de um sistema de segunda ordem com tempo morto; subamortecido ( $\xi > 1$ ).

Vamos agora calcular o ganho do sistema que para um degrau unitário (amplitude = 1) que é de **1 unidade**.

Vamos agora calcular a área **m<sub>1</sub>** sobre a curva de resposta utilizando o software Origin™, como apresentado na **Figura A.7**.

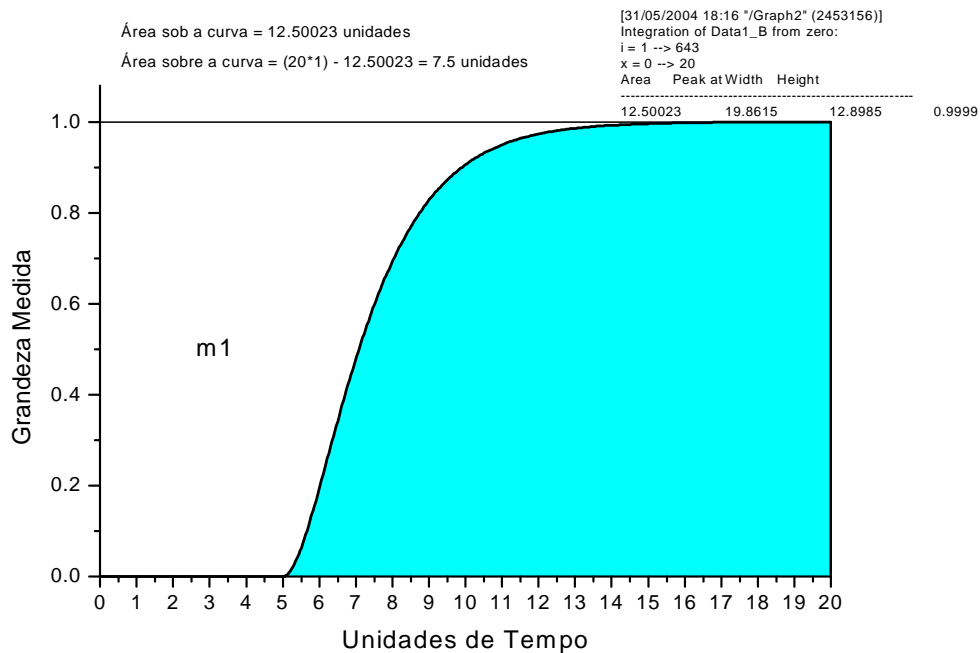


Figura A.7 – Cálculo da área  $m_1$

Encontramos o valor da área = **7,5 unidades de área**.

Agora vamos traçar uma reta visualmente que seja **tangente ao ponto de inflexão** da curva de resposta; e teremos a inclinação da reta tangente **Mi** e o valor de  **$t_m$**  ( ponto em que a reta tangente ao ponto de inflexão com o valor em regime permanente); podemos visualizar isto na **Figura A.8**.

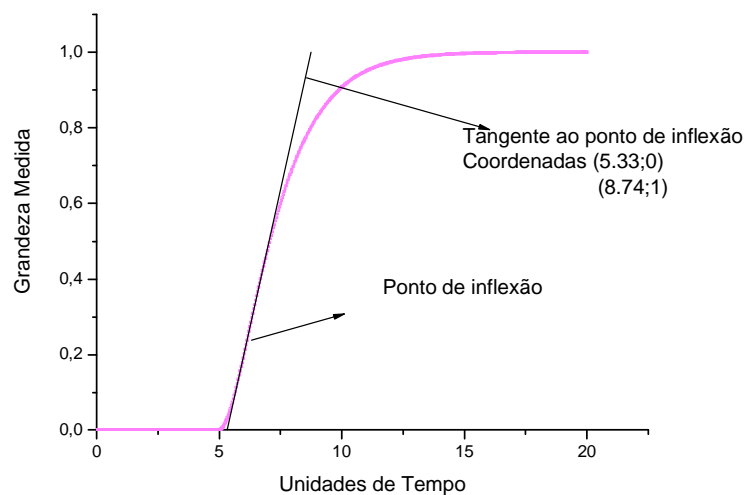


Figura A.8 – Reta tangente ao ponto de inflexão

Temos já o valor de  $t_m = 8,74$ .

Com a reta traçada podemos obter os parâmetros da reta tangente no software Origin™, a **Figura A.9** apresenta esta regressão.

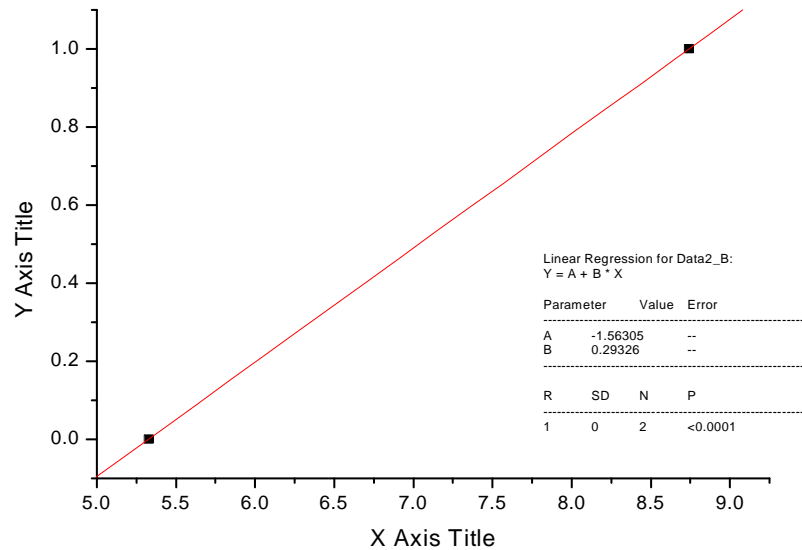


Figura A.9 – Regressão linear da reta tangente ao ponto de inflexão

Podemos agora confirmar o valor de  $t_m$  pela equação da reta:

$Y = -1,56305 + 0,29326 X$ ; para  $Y = 1$  temos  $X = 8,74 = t_m$ .

$M_i = 0,29326 =$  inclinação da reta tangente.

Agora calculamos o valor de  $\lambda = (t_m - m_1) * M_i = (8,74 - 7,5) * 0,29326 = 0,3636$ .

Com este valor de  $\lambda$  podemos calcular o valor de  $\eta$  pelo gráfico de  $\lambda = \chi e^{-\chi}$  com  $\chi$  variando de **0 a 1**. Daí temos o valor de  $\eta = \ln(\eta) / (\eta - 1)$ , e este valor é utilizado para calcularmos os parâmetros da função transferência que se está estimando.

Utilizando um programa desenvolvido em Delphi™; podemos traçar a curva  $\lambda = \chi e^{-\chi}$  com  $\chi$  variando de **0 a 1** e com o valor de  $\lambda$  calculado podemos obter um valor aproximado para  $\eta$  como mostra a **Figura A.10** a seguir.

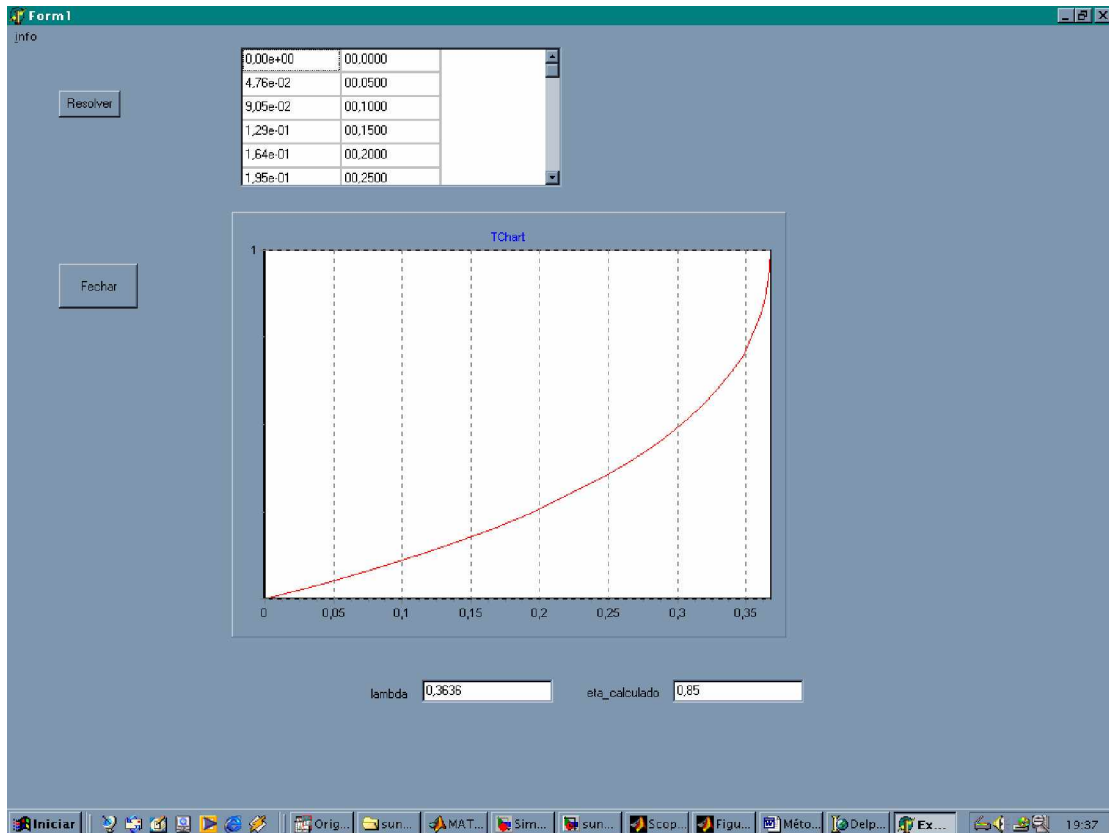


Figura A.10 – Obtenção do valor de  $\eta$

Temos um valor de  $\eta = 0,85$  unidades.

Agora podemos calcular  $\tau_1$ ,  $\tau_2$  e  $\tau_d$  estimados pelo método.

$$\tau_1 = \frac{\eta^{\frac{1}{1-\eta}}}{Mi}$$

$$\tau_2 = \frac{\frac{1}{1-\eta}}{Mi}$$

$$\tau_d = m_1 - \tau_1 - \tau_2$$

Temos :  $\tau_1 = 1,36$  ;  $\tau_2 = 1,15$  e  $\tau_d = 4,99$ ; vemos que representa uma boa aproximação da função transferência original sendo que a comparação entre as duas, a estimada e a real (simulada) segue na **Figura A.11** a seguir.

A função obtida é muito próxima da utilizada como vemos:

$$H(s) = \frac{e^{-4,99s}}{(1,36s + 1)(1,15s + 1)} = H(s) = \frac{e^{-4,99s}}{(1,564s^2 + 2,51s + 1)}$$

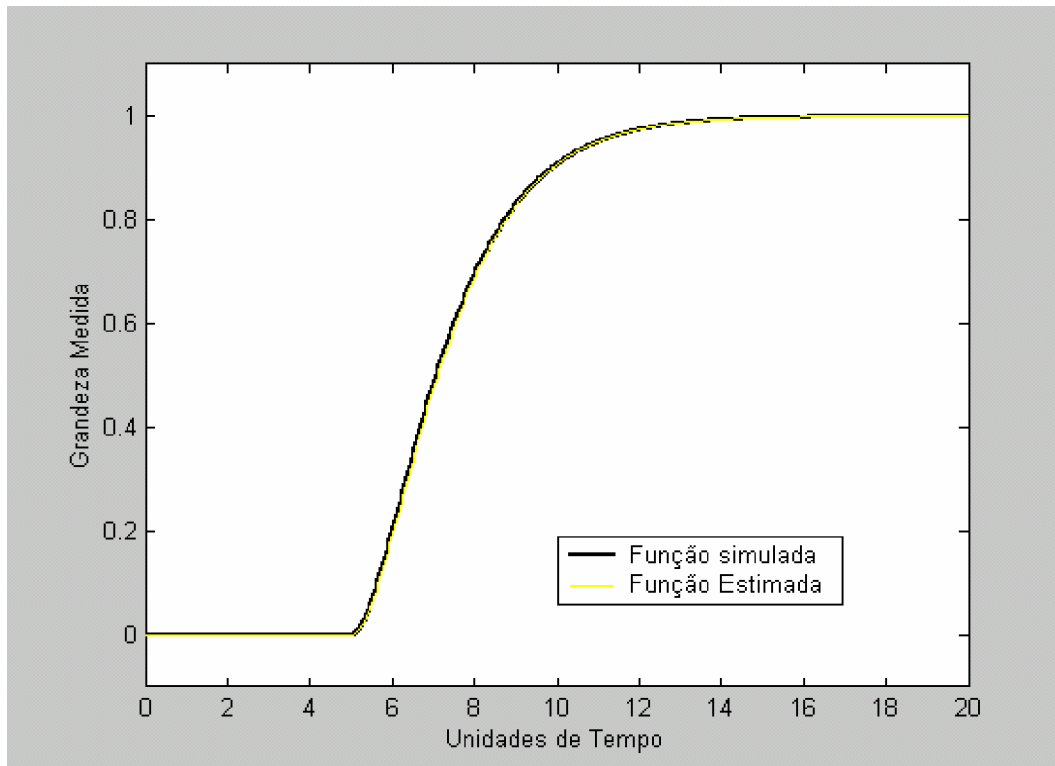


Figura A.11 – Comparação entre a função transferência simulada e a estimada

Podemos observar que para o caso dos exemplos apresentados o método foi satisfatório no tocante a aproximar as funções transferência utilizadas.

Para o caso de sistemas com resposta do tipo sub amortecido o método é similar com algumas mudanças pequenas nas equações envolvidas e no tratamento da área  $m_1$  da curva. Maiores detalhes podem ser encontrados em AGUIRRE(2000).

# APÊNDICE B

## Modificações no algoritmo do controlador PID

A versão clássica do PID não é muito usada na prática dado que melhores resultados são obtidos modificando a estrutura básica normalmente usada. A função de transferência do controlador (de  $e$  para  $u$ ) é:

$$G(s) = K_p \left( 1 + \frac{1}{sT_i} + sT_d \right) = K_p + \frac{K_i}{s} + sK_d \quad \text{ou seja;}$$

$$G(s) = P + I(s) + D(s); \text{ onde } P = K_p; I(s) = \frac{K_p}{sT_i} \text{ e } D(s) = K_p T_d s.$$

## Limitação da ação derivativa

A ação derivativa da forma expressa da maneira clássica pode gerar dificuldades se o sinal de erro estiver contaminado com ruído de alta frequência, dado que o ganho da parte derivativa cresce indeterminadamente com a frequência. Portanto é preciso limitar o ganho da ação derivativa para altas frequências. Isto é possível aproximando a ação derivativa por uma função de transferência de primeira ordem com segue:  $D(s) =$

$$\left[ \frac{sKT_d}{1 + sT_d / N} e \right].$$

Esta função, com constante de tempo igual a  $T_d/N$ , atua como uma derivada para baixas frequências e em altas frequências seu ganho é limitado a  $K_p N$ . valores típicos de  $N$  podem ser entre 3 e 20.

## Ponderação do sinal de referência

Pode-se obter uma estrutura mais flexível tratando o sinal e a saída do processo separadamente. Neste caso, definem-se diferentes sinais de erro para cada termo do controlador:

$$u(t) = K \left[ e_p(t) + \frac{1}{T_i} \int_0^t e(s) ds + T_d \frac{s}{1 + sT_d / N} e_d \right] \text{ podemos escrever como:}$$

$$u(t) = K \left[ e_p(t) + b \int_0^t e(s) ds + c e_d \right]$$

O erro na parte proporcional é expresso por:  $e_p = by_{sp} - y$ .

O erro na parte derivativa:  $e_d = cy_{sp} - y$ .

A expressão do erro na parte integral deve-se manter o mesmo que na expressão clássica do controlador PID, para poder garantir que este erro tenda sempre para zero:

$$e = y_{sp} - y.$$

Os controladores obtidos para diferentes valores de  $b$  e  $c$  somente afetarão a resposta do sistema no caso de mudanças no sinal de referência. Entretanto não será afetado o comportamento do sistema frente a distúrbios de carga, distúrbios na entrada do processo ou ruído de medição. O overshoot devido a uma mudança no sinal de referência é menor quanto menor for o valor de  $b$ . Entretanto, a redução de  $b$  também diminui a velocidade do controlador.

O parâmetro  $c$  é normalmente igual a 0(zero) para evitar grandes mudanças no sinal de controle devido a rápidas mudanças no set-point. O controlador com  $b = 1$  e  $c = 0$  é chamado algumas vezes de controlador PI – D. Outras representações do controlador PID podem ser encontradas na literatura.

# APÊNDICE C

## Discretização – Aplicação do controlador PID Digital

### Aproximação da ação proporcional:

Este termo é aproximado substituindo a variável contínua pela versão amostrada.

$P(t_k) = K_p (by_{sp} - y(t_k))$ , onde  $t_k$  representa os instantes de amostragem.

### Aproximação da ação integral:

A forma original :  $I(t) = \frac{K_p}{T_i} \int_0^t e(s) ds$  pode ser representada de maneira

equivalente como segue:  $\frac{dI}{dt} = \frac{K_p}{T_i} e$ .

Existem várias formas de aproximar esta maneira equivalente em forma discreta.

Abaixo apresenta-se três formas de aproximação:

	<b>Ação Integrativa</b>
<b>Forward Difference</b>	$I(t_{k+1}) = I(t_k) + \frac{K_p h}{T_i} e(t_k)$
<b>Backward Differencce</b>	$\frac{I(t_k) - I(t_{k-1})}{h} = \frac{K_p}{T_i} e(t_k)$
<b>Aproximação de Tustin</b>	$I(t_{k+1}) = I(t_k) + \frac{K_p h}{T_i} \frac{e(t_{k+1}) + e(t_k)}{2}$

### Aproximação da Ação Derivativa:

Da mesma forma, pode-se citar diferentes formas de aproximação do termo derivativo. A expressão modificado do termo derivativo apresentada anteriormente pode

ser expressa da forma:  $\frac{T_d}{N} \frac{dD}{dt} + D = -KT_d \frac{dy}{dt}$ . A seguir apresenta-se três formas de

aproximação desta expressão para o domínio discreto.



	<b>Ação Derivativa</b>
<b>Forward Difference</b>	$D(t_{k+1}) = \left(1 - \frac{Nh}{T_d}\right)D(t_k) - K_p N(y(t_{k+1}) - y(t_k))$
<b>Backward Difference</b>	$D(t_k) = \frac{T_d}{T_d + Nh} D(t_{k-1}) - \frac{K_p T_d N}{T_d + Nh} (y(t_k) - y(t_{k-1}))$
<b>Aproximação de Tustin</b>	$D(t_k) = \frac{2T_d - Nh}{2T_d + Nh} D(t_{k-1}) - \frac{2K_p T_d N}{2T_d + Nh} (y(t_k) - y(t_{k-1}))$

O parâmetro  $T_d$  deveria satisfazer algumas restrições para se garantir a estabilidade no caso da aproximação **Forward Difference**. A aproximação por diferença em atraso em geral oferece bons resultados para todos os valores de  $T_d$ . Portanto, esta aproximação é muito usada na discretização do termo derivativo.

# APÊNDICE D

## Outras metodologias de ajuste de controladores

### Método da Sensibilidade

Este método consiste em determinar o ganho crítico  $K_c$  e o período crítico  $T_c$  do sistema. Considere a malha de controle mostrada na **Figura D.1**.

Aplica-se um degrau na referência  $r$  e ajusta-se o ganho  $K$  até que o sistema alcance o limite de estabilidade, ou seja, a oscilação permanente, como mostrado na **Figura D.2**.

O ganho crítico  $K_c$  é o valor de  $K$  quando esta nesse limite. O período crítico  $T_c$  é igual ao período de oscilação neste estado. Os parâmetros do controlador são encontrados com o auxílio da **Tabela D.1**.

Tabela D.1 – Relações para encontrar os parâmetros do controlador

	$K_p$	$T_i$	$T_d$
<i>P</i>	$0,5 K_c$	—	—
<i>PI</i>	$0,45 K_c$	$T_c/1,2$	—
<i>PID</i>	$0,6K_c$	$T_c/2,0$	$T_c/8,0$

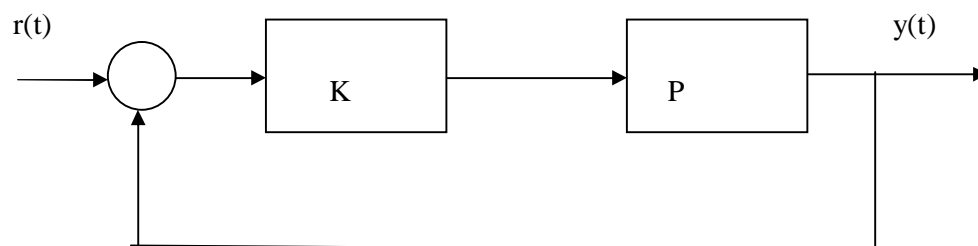


Figura D.1 – Malha usada no Método de Ziegler e Nichols

(MARLIN, EDWARD – Auto Sintonia Robusta de Controladores PID Usando Alg.Genéticos – RJ – M.Sc.

- 1997 – COPPE/UFRJ)

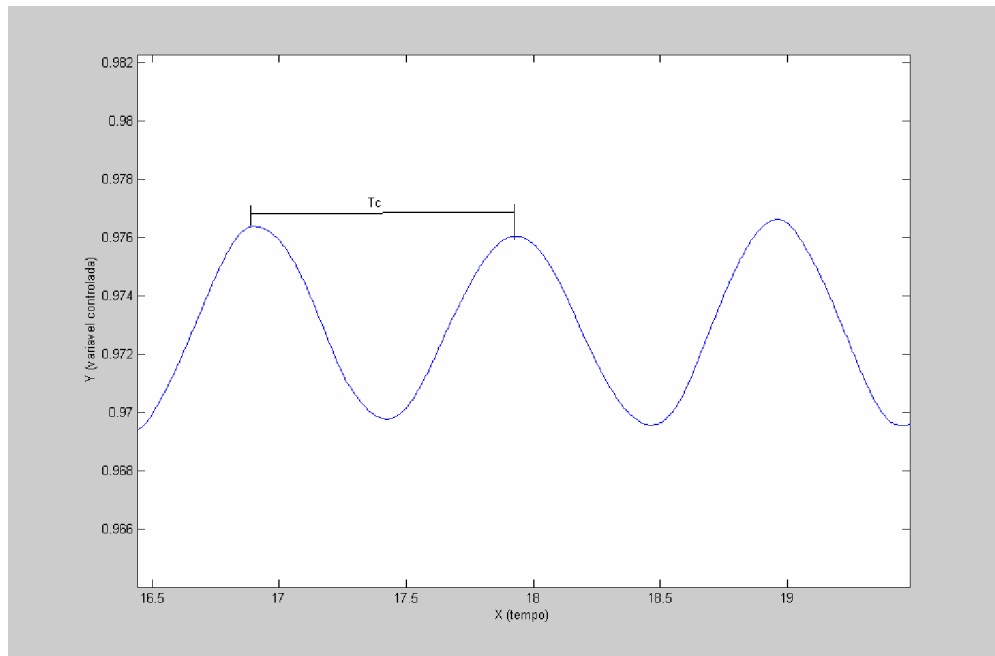


Figura D.2 – Resposta oscilatória

Este método gera no sistema uma resposta com decaimento aproximado de  $\frac{1}{4}$  de onda em cada ciclo de oscilação.

O método de Ziegler e Nichols apresenta as seguintes características:

- As respostas são muito oscilatórias.
- Diferentes regras de sintonia são necessárias para a resposta ao set-point ou para resposta a perturbações.
- As regras fornecem resultados pobres para sistemas com tempo morto normalizado grande.
- Não existe parâmetro (índice) de sintonia.

As metodologias apresentadas oferecem uma estimativa inicial dos parâmetros do controlador PID, necessitando-se para certos casos se utilizar outra metodologia a fim de melhorar o desempenho do controlador obtido.

# APÊNDICE E

Resultados obtidos para identificação do sistema de aquecimento pelo Método de Sundaresan

## Experimento 1 – Aquecimento do reator

**Metodologia:** Utilizou-se o reator de aço do LCP contendo água (cerca de 5 litros) sob agitação a fim de se identificar o sistema, obteve-se curvas de resposta para um degrau unitário de 25 unidades na válvula de aquecimento (V2) do reator sem se promover variação na válvula de resfriamento (V1). Chamou-se o conjunto de dados de Aquecimento\_03OUT.xls (planilha de dados do excel). A Curva obtida para o aquecimento do sistema é apresentada a seguir na **Figura E.1**.

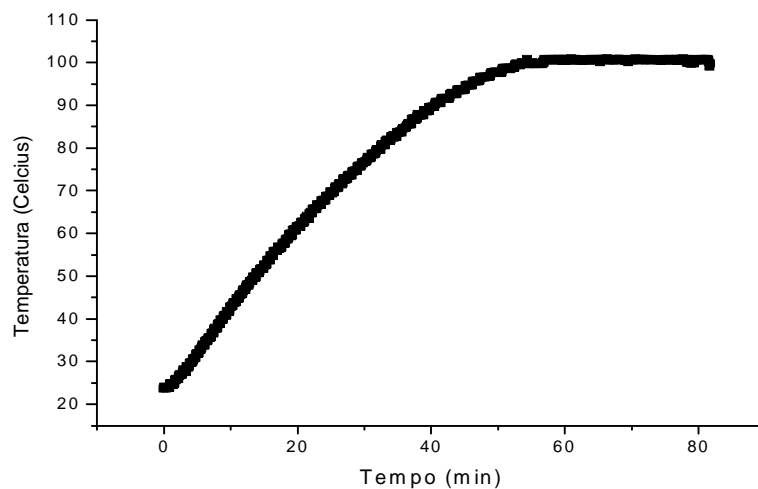


Figura – E.1 - Resposta Obtida do Sistema

**Podemos notar que esta se apresenta como o comportamento típico de um sistema de primeira ordem ou talvez um sistema de segunda ordem sobreamortecido. Vamos eliminar os pontos correspondentes ao regime estável mantendo somente os de interesse para o cálculo.**

Neste ponto vamos abordar o método de **Sundaresan** a fim de investigar a aproximação deste sistema como um sistema de segunda ordem sobreamortecido a partir de um método gráfico. Vamos normalizar os dados da temperatura a fim de manter o valor máximo em 1 para fins de cálculo.

O ganho do sistema (K) pode ser calculado pela variação da temperatura/Amplitude do Degrau =  $(99,7505 - 23,6459)/((25+25)/25 \text{ volts}) = 38,0523$  Celcius/volts.

Utilizando-se um método de integração numérica podemos obter a integral sob a curva de resposta obtida, assim obter a área sob a curva e conseqüentemente sobre a curva; que é a área  $m_1$  de interesse o método em questão podemos obter invertendo os eixos e aplicando o método de integração.

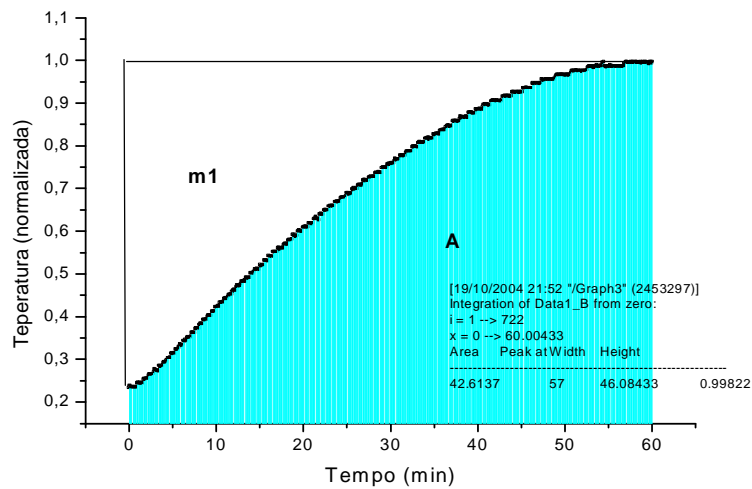


Figura E.2 – Cálculo da área

Com isto; podemos obter a área de 17,22075 unidades.

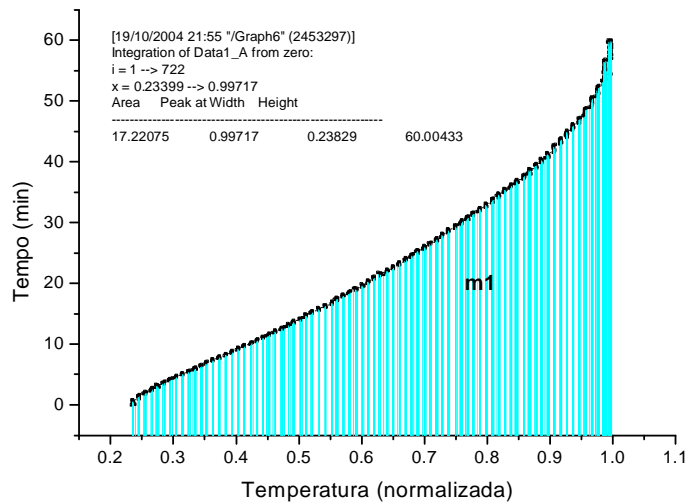
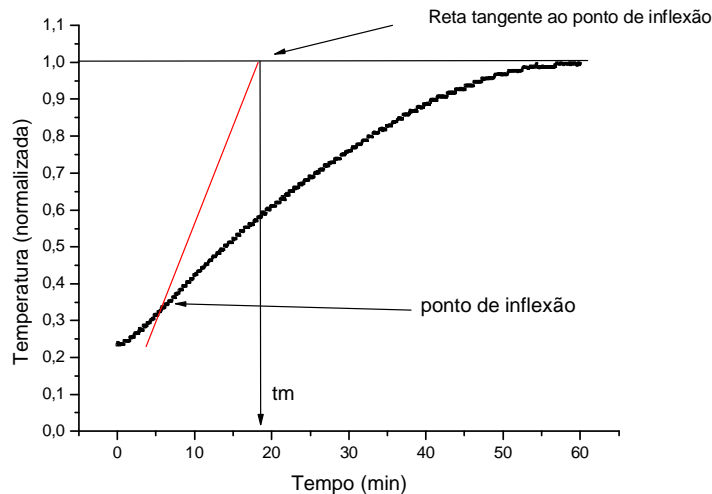
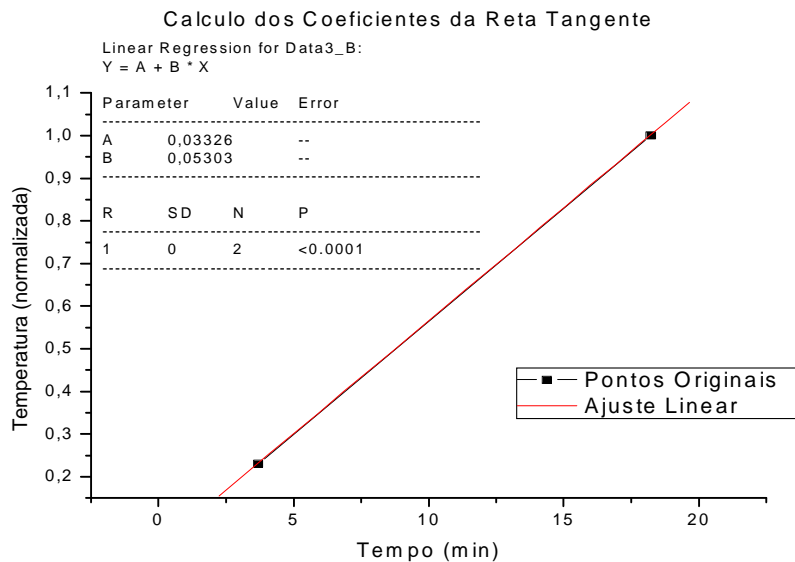


Figura E.3 - Resultados da área

Agora vamos proceder a regressão linear da reta tangente ao ponto de inflexão da curva de resposta, como mostrado nas **Figuras E.4 e E.5** a seguir foram obtidos valores para  $t_m$  e  $M_i$ :



**Figura E.4 – reta tangente ao ponto de inflexão da curva de resposta do sistema**



**Figura E.5 – Obtenção dos coeficientes da reta tangente**

Temos a equação da reta como sendo:  $Y = 0,03326 + 0,05303 * X$ .

Vamos ter o valor de  $t_m = X$  quando  $Y = 1$ ; teremos;  $t_m = 18,23$  unidades.

$M_i = 0,05303$ .

Contudo temos todos os valores necessários para determinar a função transferência para este sistema, no caso de sistema de segunda ordem sobreamortecido pelo método gráfico de Sundaresan.

Temos:  $\lambda = (t_m - m_1) * M_i$  ;  $\lambda = (18,23 - 17,22075) * 0,05303$ ; portanto  $\lambda = 0,0535$ .

Neste caso temos  $\eta < 1$ ; o sistema pode ser representado como sobreamortecido sendo que este método se aplica a sistemas sobreamortecidos ou subamortecidos.

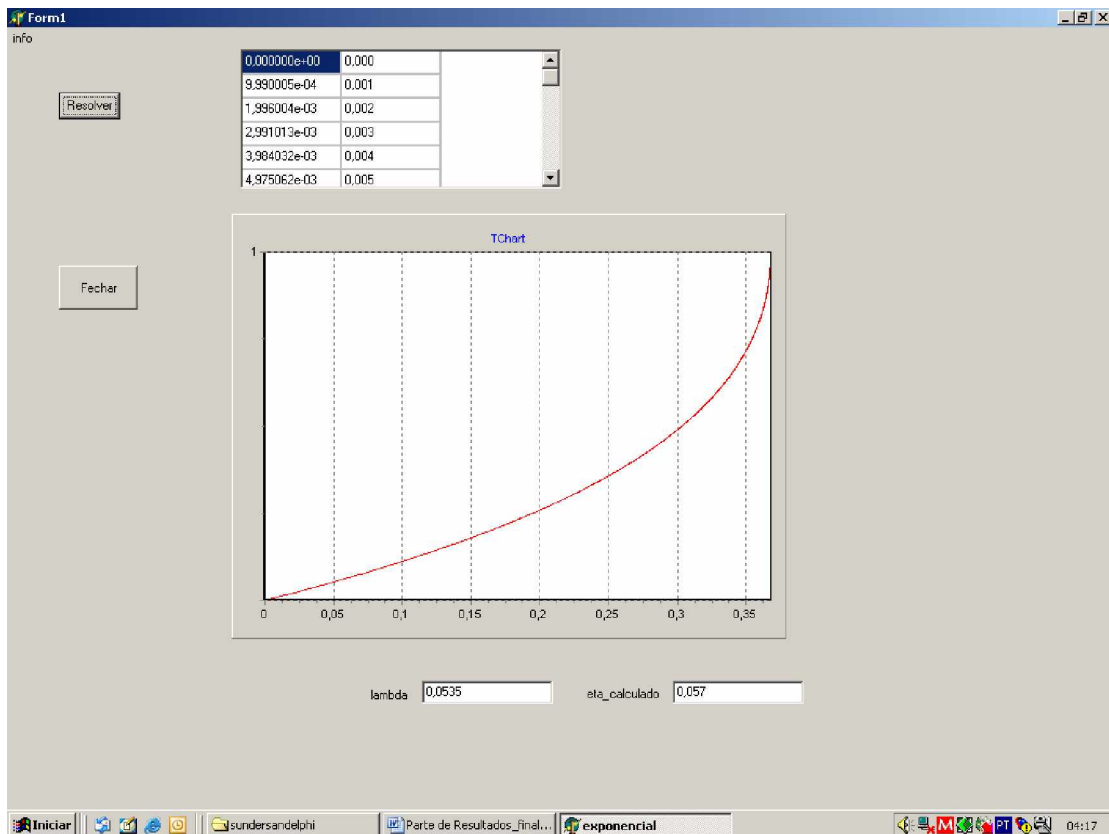


Figura - E.6 – Determinação de  $\eta$

Temos um valor de  $\eta = 0,057$  unidades.

Agora podemos calcular  $\tau_1$ ,  $\tau_2$  e  $\tau_d$  estimados pelo método.

$$\tau_1 = \frac{\eta}{M_i^{1-\eta}}$$

$$\tau_2 = \frac{1}{M_i^{1-\eta}}$$

$$\tau_d = m_1 - \tau_1 - \tau_2$$

Temos:  $\tau_1 = 15,85$  ;  $\tau_2 = 0,9039$  e  $\tau_d = 0,4668$  ; vemos que representa uma aproximação para o processo original.

A função obtida é como vemos:

$$H(s) = \frac{38,0523e^{-0,4668s}}{(15,85s + 1)(0,9039s + 1)} = H(s) = \frac{38,0523e^{-0,4668s}}{(14,32s^2 + 16,75s + 1)}.$$

Vemos que esta metodologia apresenta dificuldade no tocante a identificar o ponto de inflexão da curva de resposta.

Utilizando o Software Matlab™ podemos simular a resposta desta função a uma perturbação degrau de duas unidades (2volts) e verificar a resposta. Após isto podemos tomar os pontos experimentais, e fazendo uma pequena adaptação; tomando os pontos do eixo da temperatura e reduzindo de 23,88 para que a curva se inicie em 0 (zero) a fim de podermos compará-la com a curva simulada. E podemos ter uma idéia da eficiência do método aplicado.

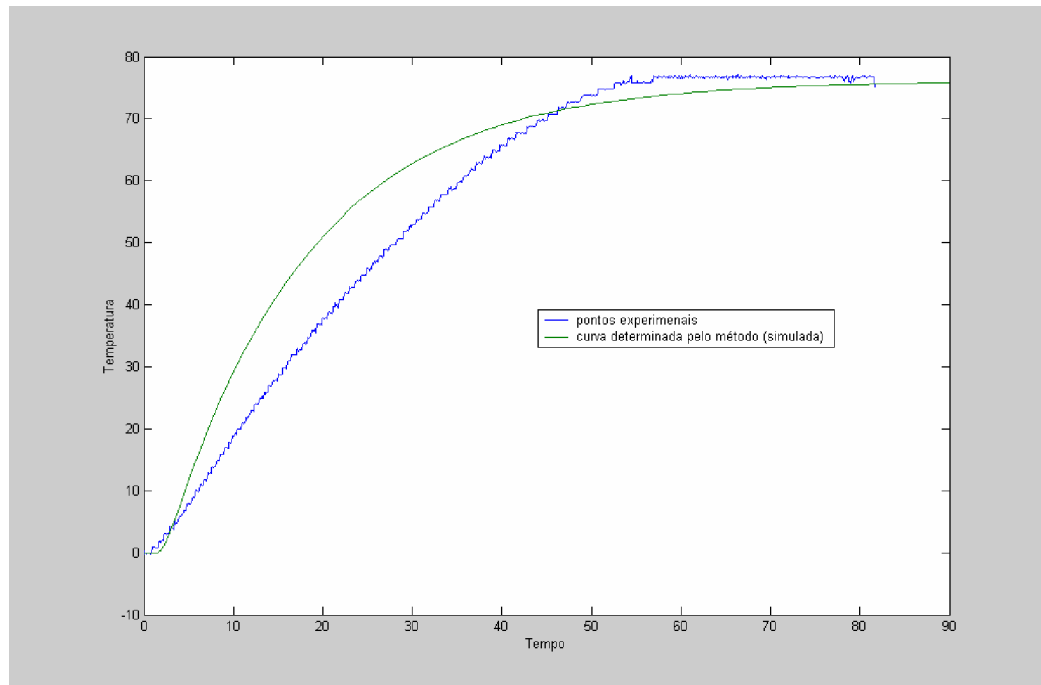


Figura E.7 – Comparação entre a curva de resposta original e a função obtida pelo método de Sundaresan

Será utilizado um outro conjunto de dados a fim de se procurar determinar uma função transferência para o processo a partir deste método procurando encontrar uma função a qual seja mais satisfatória para descrever o sistema quanto ao procedimento de aquecimento do reator.

Vamos a seguir tomar outro conjunto de dados para o aquecimento do reator procurar proceder a identificação do sistema.

## Experimento 2- Aquecimento do reator em região mediana de temperatura



**Metodologia:** Utilizou-se o reator de aço do LCP contendo água (cerca de 5 litros) sob agitação a fim de se identificar o sistema, obteve-se curvas de resposta para um degrau unitário de 20 unidades na válvula de aquecimento (V2) do reator sem se promover variação na válvula de resfriamento (V1). Neste procedimento houve uma alteração do estado inicial do sistema, donde se iniciou o sistema com ambas as válvulas em 30 % e promoveu-se o degrau. Notemos que este procedimento toma a identificação do sistema em uma região diferente do anterior.

Chamou-se o conjunto de dados de AQUEC\_2\_MEIO.xls (planilha de dados do excel). A Curva obtida para o aquecimento do sistema é apresentada a seguir.

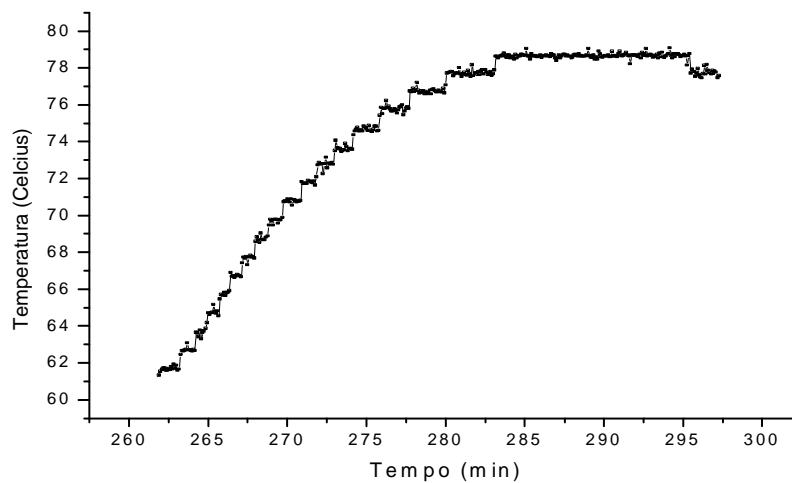


Figura E.8 – Curva de resposta obtida

Neste ponto vamos abordar o método de Sundaresan a fim de investigar a aproximação deste sistema como um sistema de segunda ordem sobreamortecido a partir de um método gráfico. Vamos normalizar os dados da temperatura a fim de manter o valor máximo em 1 para fins de cálculo.

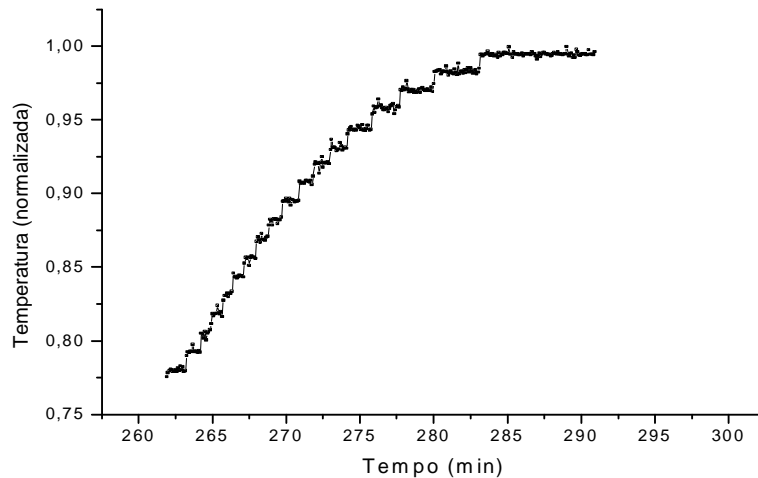


Fig E.9 – Curva de resposta normalizada

O ganho do sistema (K) pode ser calculado pela variação da temperatura/Amplitude do Degrau =  $(77,56882 - 61,31929)/(20+25/25 \text{ volts}) = 9,0275$  Celcius/volts.

Utilizando-se um método de integração numérica podemos obter a integral sob a curva de resposta obtida, assim obter a área sob a curva e conseqüentemente sobre a curva; que é a área  $m_1$  de interesse o método em questão podemos obter invertendo os eixos e aplicando o método de integração.

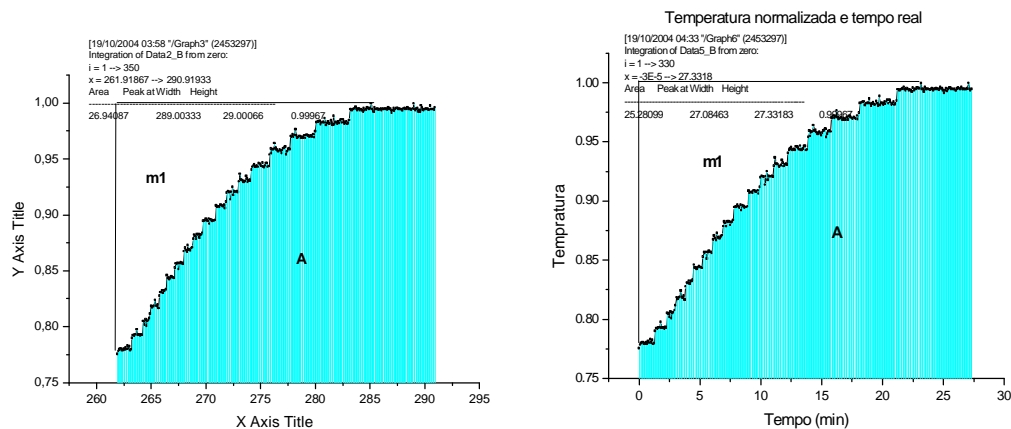


Figura E.10 - a ; b – Área de Interesse ( $m_1$ )

Tanto E.10.a quanto E.10.b representam a mesma área somente os dados foram normalizados e adequados em E.10.b para uma melhor apresentação.

A **Figura E.11** apresenta o cálculo da área  $m_1$ .

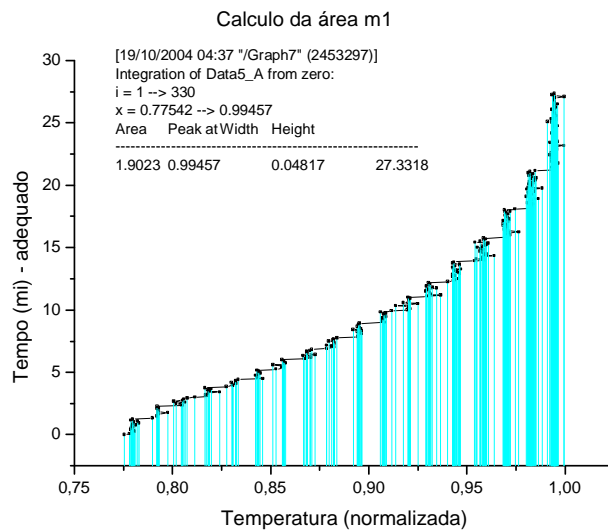


Figura E.11 – Área de interesse  $m_1$

Com isto a área  $m_1$  é de 1,9023 unidades.

Agora vamos proceder a regressão linear da reta tangente ao ponto de inflexão da curva de resposta, como mostrado na **Figura E.12** a seguir foram obtidos valores para  $t_m$  e  $M_i$ .

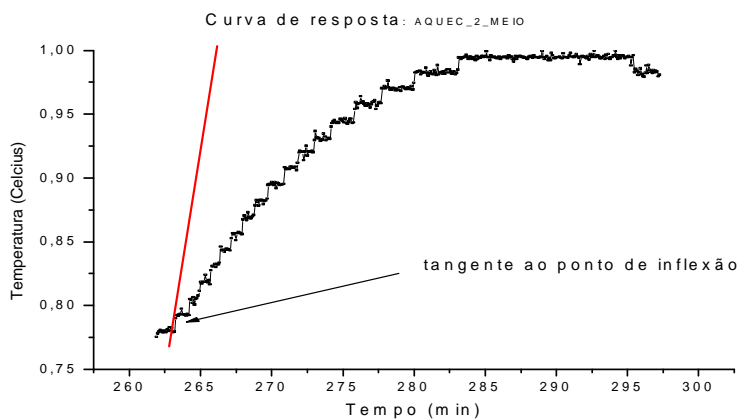


Figura E.12 – Determinação da reta tangente ao ponto de inflexão

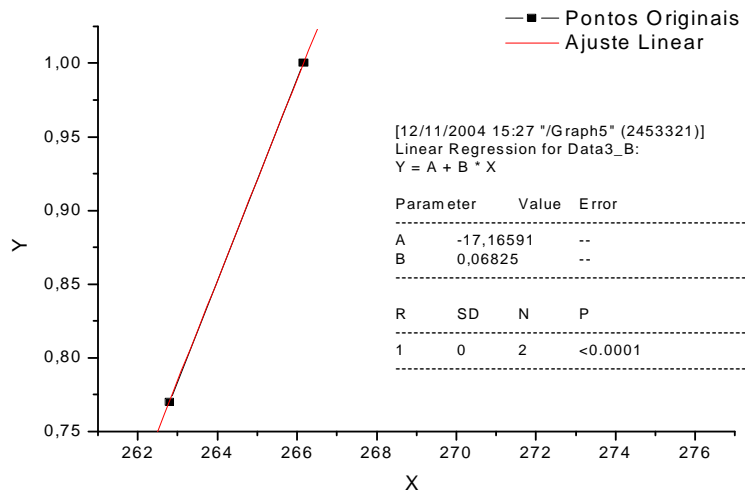


Figura E.13 – Determinação dos coeficientes da reta tangente

Temos a equação da reta como sendo:  $Y = -17,16591 + 0,06825 * X$ .

Vamos ter o valor de  $t_m = X$  quando  $Y = 1$ ; teremos;  $t_m = 266,17$  unidades. Mas, o tempo não inicia em zero então temos que descontar o valor do tempo no início do experimento para acertar a escala. Temos que  $t_m = 266,17 - 261,9187 = 4,2513$  unidades.

$$M_i = 0,06825.$$

Contudo temos todos os valores necessários para determinar a função transferência para este sistema, no caso de sistema de segunda ordem sobreamortecido pelo método gráfico de Sundaresan.

Temos:  $\lambda = (t_m - m_1) * M_i$  ;  $\lambda = (4,2513 - 1,9023) * 0,06825$  ; portanto  $\lambda = 0,160$ .

Com isto podemos obter os valores dos coeficientes do sistema (caso de sistema de segunda ordem sobreamortecido) ; utilizando um programa desenvolvido em Delphi™; podemos traçar a curva  $\lambda = \chi e^{-\chi}$  com  $\chi$  variando de **0 a 1** e com o valor de  $\lambda$  calculado podemos obter um valor aproximado para  $\eta$  como mostra a **Figura E.14** a seguir; calculando o valor de  $\eta$ .

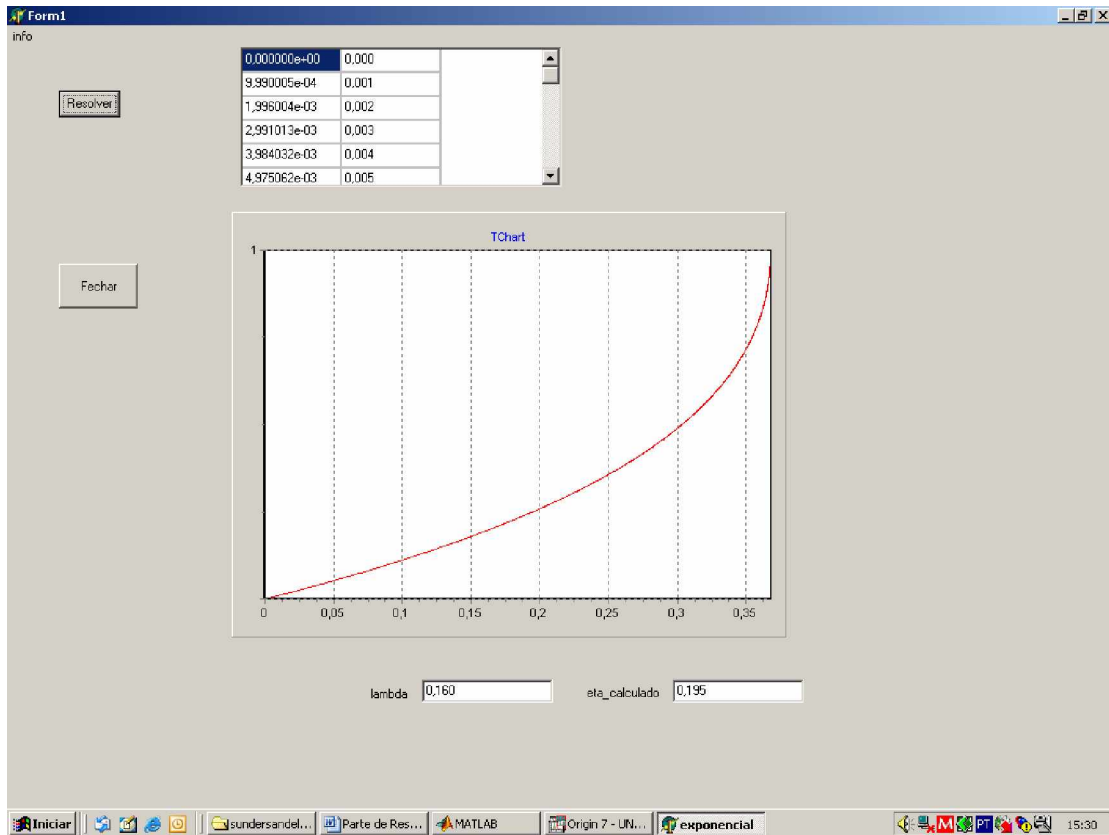


Figura E.14 – Determinação do valor de  $\eta$

Temos um valor de  $\eta = 0,195$  unidades.

Agora podemos calcular  $\tau_1$ ,  $\tau_2$  e  $\tau_d$  estimados pelo método.

$$\tau_1 = \frac{\eta^{\frac{1}{1-\eta}}}{Mi}$$

$$\tau_2 = \frac{\frac{1}{1-\eta}}{Mi}$$

$$\tau_d = m_1 - \tau_1 - \tau_2$$

Temos:  $\tau_1 = 9,86$  ;  $\tau_2 = 1,92$  e  $\tau_d = - 9,88$ ; vemos que representa uma aproximação para o processo original.

A função obtida é como vemos:

$$H(s) = \frac{9,0275e^{9,88s}}{(9,86s + 1)(1,92s + 1)} = H(s) = \frac{9,0275e^{9,88s}}{(18,93s^2 + 11,78s + 1)}$$

O valor negativo encontrado para o tempo morto do sistema nos impossibilita avaliar esta função; podemos notar que uma das principais limitações deste método é localizar o ponto de inflexão da curva de resposta e traçar a reta tangente. Também para este caso se “forçou” a utilização do método mesmo notando que a curva de resposta não apresenta comportamento muito semelhante ao tipo de função a que se tentou aproximar.

O procedimento de localização do ponto de inflexão é muito importante para o sucesso do método em questão e pequenas diferenças no procedimento podem ocasionar erros como o obtido através deste método.

### **Experimento 3 - Resfriamento do reator**

**Metodologia:** Utilizou-se o reator de aço do LCP contendo água (cerca de 5 litros) sob agitação a fim de se identificar o sistema, obteve-se curvas de resposta para um degrau unitário de 25 unidades na válvula de resfriamento (V1) do reator sem se promover variação na válvula de aquecimento (V2). Este procedimento é o inverso do procedimento 1, quando o reator aquecido é resfriado. Chamou-se o conjunto de dados de Resfriamento\_03OUT.xls (planilha de dados do excel). A Curva obtida para o resfriamento do sistema é apresentada a seguir na **Figura E.15**.

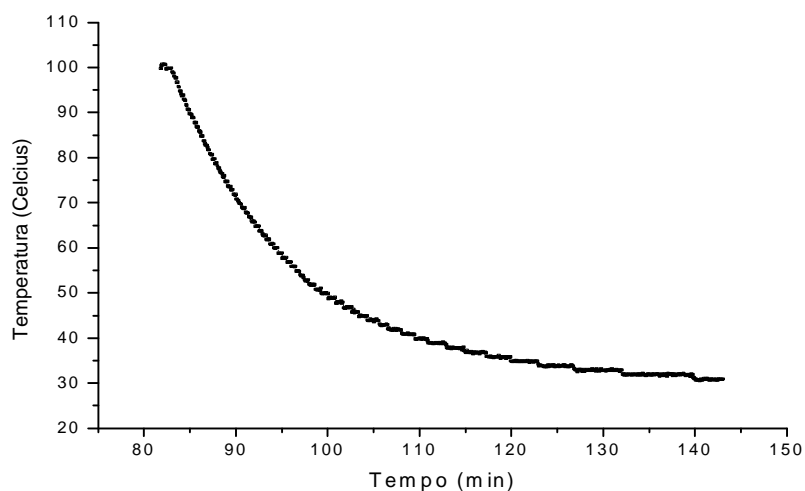


Figura E.15 – Curva de resposta obtida para o resfriamento

Este sistema pode ser observado também com um comportamento similar ao sistema de aquecimento; entretanto, é de se esperar uma pequena mudança nos

coeficientes da função transferência para este caso. Contudo, podemos esperar que o sistema seja um sistema de primeira ordem ou mesmo um sistema de segunda ordem sobreamortecido.

Nesta etapa vamos abordar o método de Sundaresan para identificação de modelos com comportamento de segunda ordem sobreamortecidos. Vamos calcular o ganho do sistema e posteriormente normalizar os eixos da temperatura e do tempo a fim de proceder os cálculos.

Notemos que em comparação com a curva anterior temos uma curva “invertida”, mas isto não nos impede de proceder o método; somente devemos observar que a dinâmica buscada é a de resfriamento neste caso, ou seja, o ganho do processo é negativo.

O ganho do sistema (K) pode ser calculado pela variação da temperatura/Amplitude do degrau aplicado =  $(30,8669 - 100,4086)/((25+25)/25 \text{ volts}) = -34,77085$  (ganho negativo – resfriamento). Para fins de cálculo da área  $m_1$  de interesse vamos proceder a normalização dos eixos do gráfico. Note que neste caso a área  $m_1$  é a própria área sob a curva de resposta. A **Figura E.16** apresenta a curva de resposta normalizada.

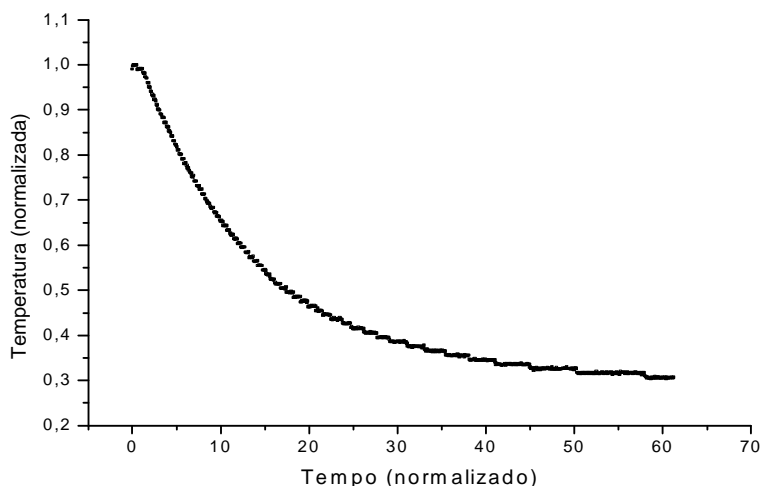


Figura E.16 – Curva de resposta normalizada

Com isto calculamos a área  $m_1$ ; a **Figura E.17** a seguir apresenta o resultado encontrado.

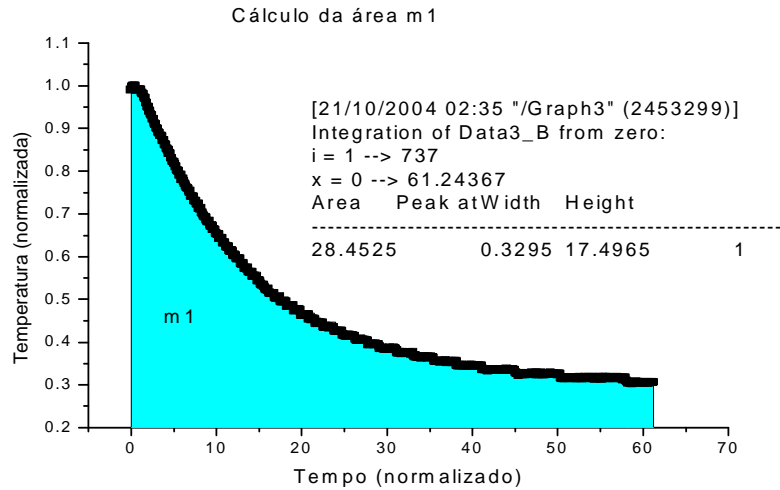


Figura E.17 – resultado obtido para cálculo da área m 1

A área encontrada foi de **28,4525 unidades**.

Agora vamos proceder a regressão linear da reta tangente a curva de resposta no ponto de inflexão, como mostrado na **Figura E.18**, a fim de determinarmos  $M_i$ ; e  $t_m$ .

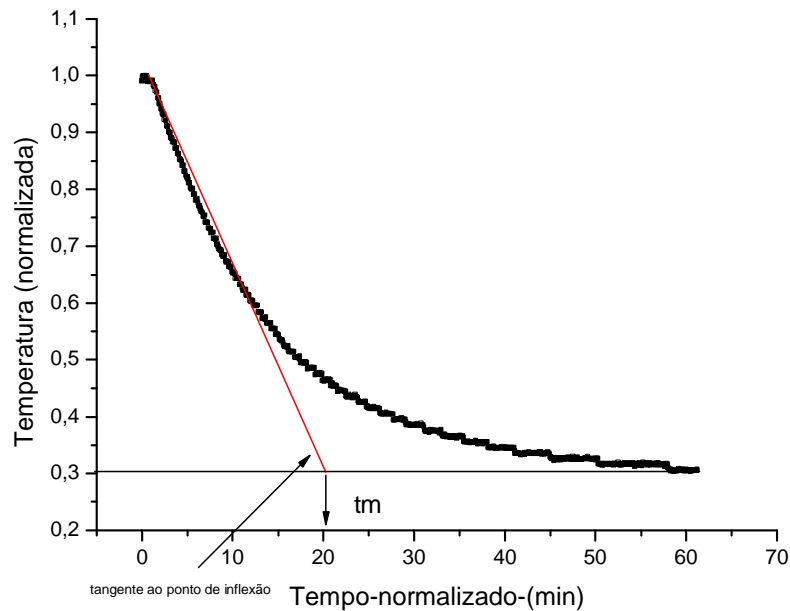


Figura E.18 – Determinação da reta tangente ao ponto de inflexão



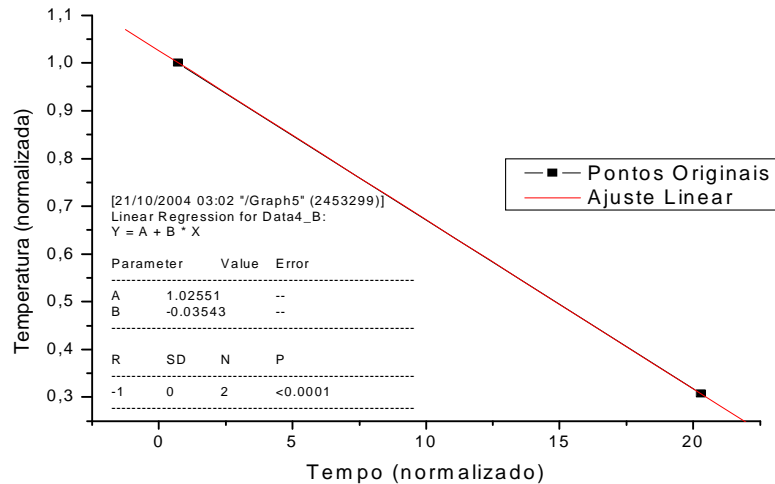


Figura E.19 – Determinação dos coeficientes da reta tangente ao ponto de inflexão

Temos que a equação da reta sendo:  $Y = 1,02551 - 0,03543 * X$ .

Com isto o valor de  $t_m$  fica sendo quando  $Y = 0.30658$ ; temos:  $t_m = 20,29$  unidades.

$M_i = -0,03543$ .

Contudo temos todos os valores necessários para determinarmos a função transferência para este sistema através do método de Sundaresan (sistema de segunda ordem sobre amortecido).

Temos:  $\lambda = (t_m - m_1) * M_i$  ;  $\lambda = (20,29 - 28,4525) * -0,03543$ ; portanto  $\lambda = 0,2892$ .

Com isto podemos obter os valores dos coeficientes do sistema (caso de sistema de segunda ordem sobre amortecido); utilizando um programa desenvolvido em Delphi™; podemos traçar a curva  $\lambda = \chi e^{-\chi}$  com  $\chi$  variando de **0 a 1** e com o valor de  $\lambda$  calculado podemos obter um valor aproximado para  $\eta$  como mostra a **Figura E.20** a seguir; calculando o valor de  $\eta = \mathbf{0,459}$ .

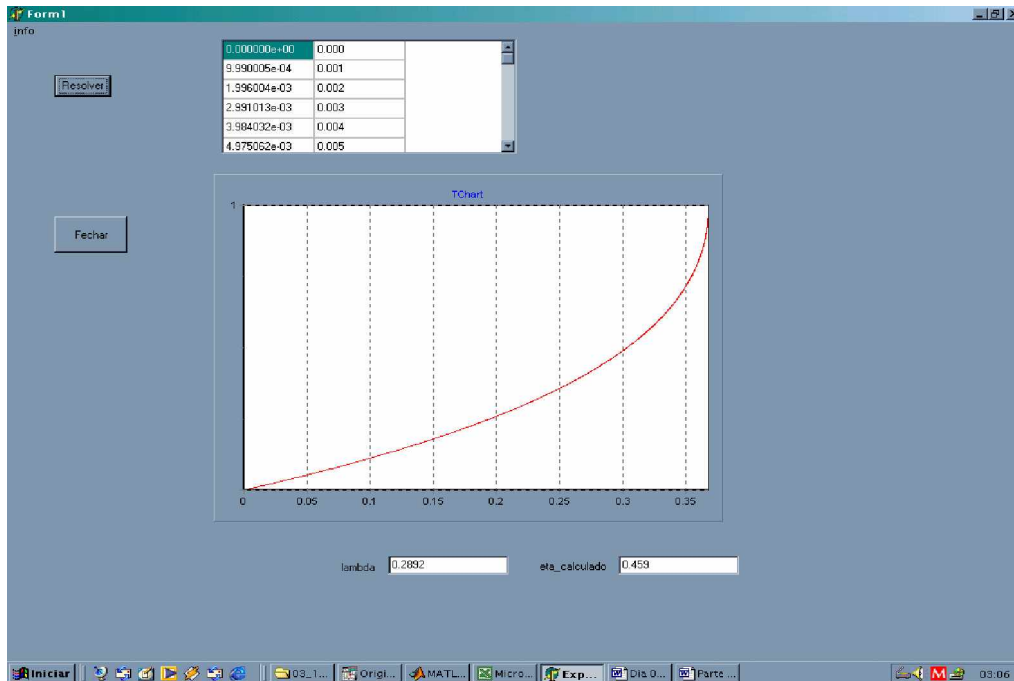


Figura E.20 – Cálculo de  $\eta$

Agora podemos calcular  $\tau_1$ ,  $\tau_2$  e  $\tau_d$  estimados pelo método.

$$\tau_1 = -\frac{\eta}{Mi} \quad (\text{pois o coeficiente da reta neste caso é negativo})$$

$$\tau_2 = -\frac{1}{Mi} \quad (\text{pois o coeficiente da reta neste caso é negativo})$$

$$\tau_d = m_1 - \tau_1 - \tau_2$$

Temos :  $\tau_1 = 14,578$  ;  $\tau_2 = 6,6913$  e  $\tau_d = 7,1832$ ; vemos que representa uma aproximação para o processo original.

A função obtida é como vemos:

$$H(s) = \frac{-34,77e^{7,1832s}}{(14,578s + 1)(6,6913s + 1)} = H(s) = \frac{-34,77e^{7,1832s}}{(97,5457s^2 - 21,2693s + 1)}$$

Neste caso pelo fato da curva ter inclinação negativa os valores de  $\tau_1$  e  $\tau_2$  foram multiplicados por -1 para fornecer um resultado mensurável.

Desta vez; o valor encontrado para o tempo morto do sistema não é negativo; e apesar disto a função transferência encontrada não é estável e não pode ser avaliada. Mais uma vez se esbarra em limitações do procedimento gráfico, donde é muito

importante para o sucesso do método em questão o ponto de inflexão da curva de resposta e a determinação da reta tangente a este ponto; e pequenas diferenças no procedimento podem ocasionar erros como o obtido através deste método.

Podemos ter um comparativo da função estimada versus a curva de resposta no gráfico a seguir.

Simulando no Matlab™ e fazendo um rearranjo de eixos podemos comparar com o curva de resposta do sistema e função aproximada na **Figura E.21**.

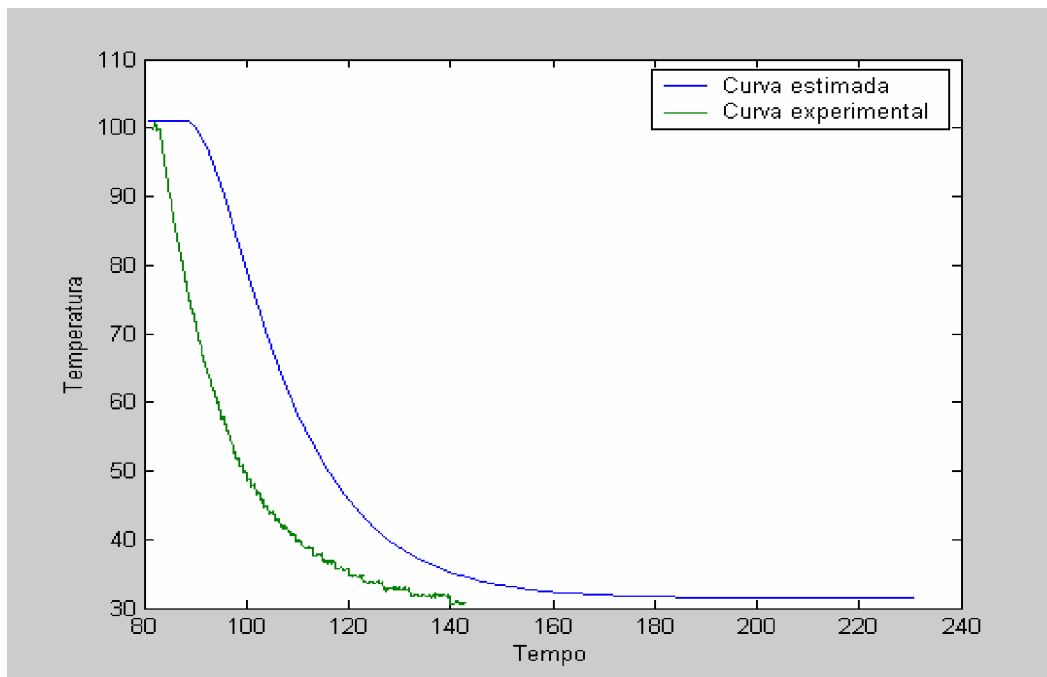


Figura E.21 – Comparação entre a curva de resposta estimada e a real

Notamos que para o sistema em questão torna-se difícil aproximar a curva de resposta para um sistema sobreamortecido de segunda ordem com tempo morto da forma com foi proposta, pois a aproximação obtida é muito distante da curva original.

Podemos notar que através desta metodologia é muito difícil determinarmos funções transferência as quais possam representar seguramente o processo em identificação. Portanto outras metodologias devem ser exploradas.

## APÊNDICE F

Códigos em Delphi® escritos na dissertação.

- 1) Código utilizado para cálculo dos parâmetros do controlador PID, Cohen-Coon e ITAE.
- 

```

unit Abertura;
interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Buttons;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    sele: TRadioGroup;
    BitBtn1: TBitBtn;
    controlador: TRadioGroup;
    procedure BitBtn1Click(Sender: TObject);
  private
    Procedure CC;//ctrl shift e c
    Procedure ITAE;
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
uses Resultados;
{$R *.DFM}

procedure TForm1.CC;
var K, TAU, TETA : Real;
var KC, TAUD, TAUI : Real;
begin
  K := strtofloat(Edit1.text);//lê os valores nas caixas de texto da form1
  TAU := strtofloat (Edit2.text);
  TETA := strtofloat (Edit3.text);

```

```

if controlador.ItemIndex = 0 then
begin
KC := (1/K) * (TAU/TETA)*(0.9+(TETA/(12*TAU)));
TAUI := (TETA*(30+3*(TETA/TAU))/(9+(20*(TETA/TAU)));
TAUD := 0;
end;
if controlador.ItemIndex = 1 then
begin
KC := (1/K) * (TAU/TETA)*(1+(TETA/(3*TAU)));
TAUI := 0;
TAUD := 0;
end;
if controlador.ItemIndex = 2 then
begin
KC := (1/K)*(TAU/TETA)*(((16*TAU)+(3*TETA))/(12*TAU));
TAUI := (TETA*(32+6*(TETA/TAU))/(13+(8*(TETA/TAU)));
TAUD := (4*TETA/(11+(2*TETA/TAU)));
end;
Form2.KCedit.text := floattostr(KC); //envia os resultados para as caixas de texto da
form2
Form2.TAUDedit.text := floattostr (TAUD);
Form2.TAUIedit.text := floattostr (TAUI);
Form2.ShowModal;//mostr a form2
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
if sele.ItemIndex = 0 then //testa se é cohen coon
begin
CC;//procedure
end;
if sele.ItemIndex = 1 then //testa se é ITAE
begin
ITAE;//procedure
end;
end;

procedure TForm1.ITAE; //para perturbação de set-point
var K, TAU, TETA : Real;
var KC, TAUD, TAUI : Real;
var C : Real;
Function X_TO_Y(x,y:real) : real;
Var r : real;
begin;
r:=y*Ln(x);
X_TO_Y:=Exp(r);
end;
begin
K := strtfloat(Edit1.text);
TAU := strtfloat (Edit2.text);

```

```

TETA := strtfloat (Edit3.text);
if controlador.ItemIndex = 0 then
begin
C:= (TETA/TAU);
KC := (0.586*(X_TO_Y(C,-0.916)))/K;
TAUI := TAU/(1.03{A para o parametro I com mud set point}-
0.165{B}*(TETA/TAU));
TAUD := 0;
end;
if controlador.ItemIndex = 1 then //O P não se faz por este método
begin
KC := 0;
TAUI := 0;
TAUD := 0;
end;
if controlador.ItemIndex = 2 then
begin
C:= (TETA/TAU);
KC := (0.965*(X_TO_Y(C,-0.85)))/K;
TAUI := TAU/(0.796{A para o parametro I com mud set point}-
0.1465{B}*(TETA/TAU));
TAUD := TAU*(0.308*(X_TO_Y(C,0.929)));
end;

Form2.KCedit.text := floattostr(KC);
Form2.TAUDedit.text := floattostr (TAUD);
Form2.TAUIedit.text := floattostr (TAUI);
Form2.ShowModal;
end;
end.

```

---

- 2) Código utilizado para determinação do valor de  $\eta$  pelo método de identificação de Sundaresan.

```

unit Unit1;
interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Grids, Buttons, TeEngine, Series, ExtCtrls, TeeProcs, Chart,
  Menus;
type
  TForm1 = class(TForm)
    Button1: TButton;
    BitBtn1: TBitBtn;
    StringGrid1: TStringGrid;
    Chart1: TChart;

```

```

Series1: TLineSeries;
MainMenu1: TMainMenu;
info1: TMenuItem;
Edit1: TEdit;
Label1: TLabel;
Edit2: TEdit;
Label2: TLabel;
procedure Button1Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Info1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
uses Unit2;
{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
var
  i:integer;
  p:array[0..101] of double;
  y:array[0..101] of double;
  lambda:real;
  eta:real;
  dif:real;
begin
  lambda:=strtofloat(Form1.Edit1.text);
  for i:= 0 to 20 do
  begin
    p[i]:=i/20;
    y[i]:=p[i]*exp(-(p[i]));
    dif:=y[i]-lambda;
    if dif <= 0.0000000001 then
    begin
      eta:=p[i];
      Form1.Edit2.text:=floattostr(eta);
    end;
    StringGrid1.CELLS[0,i]:=formatfloat('0.00e+00',y[i]);
    StringGrid1.CELLS[1,i]:=formatfloat('00.0000',p[i]);
  end;
  Series1.Clear;
  for i:=0 to 20 do
  begin
    Series1.addxy(StrtoFloat(StringGrid1.cells[0,i]),
      StrtoFloat(StringGrid1.cells[1,i]),clteecolor);
  end;
end;

```

```
end;  
  
procedure TForm1.BitBtn1Click(Sender: TObject);  
begin  
close;  
end;  
procedure TForm1.Info1Click(Sender: TObject);  
begin  
Form2.Showmodal  
end;  
end.
```

---