

**GUSTAVO MAIA DE ALMEIDA**

**CONTROLE PREDITIVO SINTONIZADO  
VIA ALGORITMO GENÉTICO APLICADO  
EM PROCESSOS SIDERÚRGICOS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Doutor em Engenharia Elétrica.

Orientador: Prof. Dr. José Leandro Félix Salles.

Co-orientador: Prof. Dr. José Denti Filho.

VITÓRIA

2011

Dados Internacionais de Catalogação na publicação (CIP)  
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

---

A447c Almeida, Gustavo Maia de, 1979-  
Controle preditivo sintonizado via algoritmo genético aplicado em processos siderúrgicos / Gustavo Maia de Almeida. - 2011. 171 f. : il.

Orientador: José Leandro Félix Salles.

Coorientador: José Denti Filho.

Tese (Doutorado em Engenharia Elétrica) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Controle Preditivo. 2. Algoritmos genéticos. 3. Identificação de sistemas. I. Salles, José Leandro Félix. II. Denti Filho, José. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 621.3

---

**GUSTAVO MAIA DE ALMEIDA**

**CONTROLE PREDITIVO SINTONIZADO  
VIA ALGORITMO GENÉTICO APLICADO  
EM PROCESSOS SIDERÚRGICOS**

Tese submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Doutor em Engenharia Elétrica.

Aprovada em 04 de março de 2011.

**COMISSÃO EXAMINADORA**

---

Prof. Dr. José Leandro Félix Salles  
Universidade Federal do Espírito Santo  
Orientador

---

Prof. Dr. José Denti Filho  
Universidade Federal do Espírito Santo  
Co-orientador

---

Profa. Dra. Jussara Farias Fardin  
Universidade Federal do Espírito Santo

---

Prof. Dr. Alessandro Mattedi  
Universidade Federal do Espírito Santo

---

Prof. Dr. Antônio Augusto Rodrigues Coelho  
Universidade Federal de Santa Catarina

---

Prof. Dr. Marcelo Lucas Pereira Machado  
Instituto Federal do Espírito Santo

*Dedico esta tese a Tatiana.*

*Por sua inestimável ajuda durante esses longos anos de estudo, especialmente neste último, por seu enorme carinho, sua generosidade incomum e sua paciência para entender a minha rabugice com tanto amor e bom humor e, sobretudo por sempre conseguir enxergar o que há de melhor em todas as coisas da vida.*

## Agradecimentos

Agradeço primeiramente à Deus por sempre me dar força para desenvolver o trabalho.

Agradeço também ao meu amigo e orientador José Leandro, que durante esses anos de pesquisa proporcionou os meios necessários para que eu pudesse concluir a minha tese.

Agradeço a minha esposa Tatiana e as minhas filhas Ana Júlia e Gabriela por me confortar com amor e carinho em todos os momentos de minha vida.

Aos meus pais Celso e Wilma, irmãos Celso, Pedro e Karynna por todo amor e incentivo que sempre recebi.

A todos os meus amigos que sempre me incentivaram durante todos esses anos de árduo e rigoroso trabalho.

A todos aqueles que de forma direta ou indireta contribuíram para o desenvolvimento deste trabalho.

## RESUMO

As técnicas de Controle Preditivo Baseado em Modelos (MPC) são cada vez mais aplicadas na indústria, pois geralmente exibem um bom desempenho e robustez, desde que os parâmetros do controlador sejam sintonizados corretamente. Nesta tese utiliza-se o Algoritmo Genético (AG) para realizar a sintonia dos parâmetros do controlador preditivo mono e multivariável em modelos lineares e não lineares.

A técnica existente na literatura para sintonia do Controlador Preditivo por Matriz Dinâmica (DMC), a qual é utilizada para controlar sistemas que apresentam resposta ao degrau estável em malha aberta, será comparada com a sintonia via AG. No caso em que o processo em malha aberta apresenta resposta ao degrau instável, não existe um método analítico para a sintonia. Por isto, observa-se na literatura que a utilização do MPC em alguns sistemas instáveis em malha aberta, lineares ou não (como estes a serem estudados nesta tese) é inexistente, pois a sintonia baseada em tentativa e erro é as vezes impraticável. Portanto, neste trabalho estuda-se a aplicação do MPC sintonizado por Algoritmo Genético em dois processos instáveis em malha aberta, os quais são muito importantes na indústria Siderúrgica.

O primeiro é o Laminador constituído por 3 cadeiras, onde se deseja minimizar a variação da espessura da tira da última cadeira devido aos distúrbios que afetam o processo tais como: temperatura da tira e variações da espessura na primeira cadeira. Neste caso utiliza-se o modelo linear e multivariável do Laminador para desenvolver o Controlador Preditivo Generalizado (GPC) cujos parâmetros são sintonizados por AG. O segundo processo instável em malha aberta é o nível do Molde do Lingotamento Contínuo, o qual possui um modelo não linear e portanto será controlado através de técnicas de controle preditivo não linear usando redes neurais e modelo Hammerstein. É feita uma comparação entre estes controladores para analisar a estabilidade e a robustez quando o molde é afetado por distúrbios de *Bulging*, *Clogging* e Argônio.

## ABSTRACT

Techniques of Model Based Predictive Control (MPC) are increasingly applied in industry because they generally exhibit a good performance and robustness, since the parameters of controller are tuned correctly. This thesis use the Genetic Algorithm (GA) to perform the tuning of parameters of the predictive controller to control mono and multivariable, linear and nonlinear models.

The existing technical literature for tuning Predictive Dynamic Matrix Controller (DMC), which use a step response to control systems that are open loop stable will be compared with the tuning by GA. In the event that the process is unstable in open-loop, there is not an analytical method for the tuning. Therefore, it is noted in the literature that the use of MPC in some open-loop unstable systems, linear or not (Such as these to be studied in this thesis) is lacking, because the tuning procedure is based on trial and error and sometimes is impractical. Therefore, this study focuses the application of the MPC tuning by Genetic algorithm for two open-loop unstable processes, which are very important in the Steel Industry.

The first is composed by Rolling Mill Stands, where we wish to minimize the variation of strip thickness the last stand due to disturbances that affect the process such as temperature and strip thickness variations in the first stand. In this case we use the linear and multivariable model to develop the Generalized Predictive Controller (GPC) whose parameters are tuned by GA. The second process, unstable in open-loop, is the level of Mold of a Continuous Casting, which has a nonlinear model and is therefore controlled by techniques of nonlinear predictive control using neural networks and Hammerstein model. A comparison is made between these controllers to analyze the stability and robustness when the mold is affected by disturbance of Bulging, Clogging and Argon.

# Sumário

|   |           |
|---|-----------|
| <b>1 - Introdução</b>   | <b>1</b>  |
| 1.1 - Aplicações do MPC   | 5         |
| 1.2 - Métodos de Sintonia da Literatura   | 11        |
| 1.3 - Objetivos da Tese   | 12        |
| 1.4 - Organização da Tese   | 13        |
| <b>2 - Revisão Bibliográfica</b>  | <b>15</b> |
| 2.1 - Modelo Hammerstein  | 15        |
| 2.2 - Identificação do Modelo Hammerstein através do Método dos Mínimos Quadrados | 17        |
| 2.3 - Redes Neurais   | 20        |
| 2.3.1 - Redes Perceptron Multicamadas (MLP)                                       | 21        |
| 2.4 - Identificação em Redes Neurais Artificiais                                  | 25        |
| 2.4.1 - Seleção da Estrutura de Modelos Neurais                                   | 25        |
| 2.4.2 - Estimação de Modelos Neurais  | 28        |
| 2.5 - Validação de Modelos  | 31        |
| 2.6 - Algoritmo Genético  | 32        |
| 2.6.1 - Configuração do Algoritmo Genético  | 34        |
| 2.6.2 - Método de Seleção   | 36        |
| 2.6.3 - Operadores Genéticos  | 38        |
| 2.6.4 - Configuração dos Parâmetros   | 41        |
| 2.6.5 - Convergência  | 42        |
| 2.7 - Conclusão do Capítulo   | 43        |
| <b>3 - Controle Preditivo Baseado em Modelos Lineares</b>                         | <b>45</b> |

|   |           |
|---|-----------|
| 3.1 - Controle por Matriz Dinâmica (DMC)  | 45        |
| 3.1.1 - DMC Monovariável  | 46        |
| 3.1.2 - Solução do DMC Irrestrito   | 48        |
| 3.1.3 - Caso Restrito Monovariável  | 49        |
| 3.1.4 - DMC Multivariável   | 50        |
| 3.2 - Controle Preditivo Generalizado (GPC)   | 53        |
| 3.2.1 - GPC Monovariável  | 53        |
| 3.2.2 - GPC Multivariável   | 59        |
| 3.3 - Conclusão do Capítulo   | 62        |
| <b>4 - Sintonia do MPC Linear</b>   | <b>63</b> |
| 4.1 - Sintonia através do Guia Proposto em <i>Dougherty e Cooper 2003</i> [34]        | 64        |
| 4.2 - Sintonia do MPC através do Algoritmo Genético                                   | 65        |
| 4.3 - Comparação entre as Técnicas de Sintonia  | 67        |
| 4.3.1 - DMC Monovariável Restrito   | 67        |
| 4.3.2 - DMC Multivariável Restrito  | 75        |
| 4.4 - GPC do Laminador Sintonizado através do AG                                      | 77        |
| 4.5 - Conclusão do Capítulo   | 87        |
| <b>5 - MPC Não-Linear com Modelos Hammerstein e Neural</b>                            | <b>89</b> |
| 5.1 - GPC com Perturbações Mensuráveis para o Modelo Hammerstein (HGPC)               | 90        |
| 5.2 - Controle Preditivo com Redes Neurais  | 92        |
| 5.2.1 - MPC Neural com Regra Baseado no Gradiente (PNRG)                              | 96        |
| 5.2.2 - Controle Preditivo Neural com Linearização Instantânea)                       | 100       |
| 5.3 - MPC Não-Linear do Nível do Molde do Lingotamento Contínuo                       | 102       |
| 5.3.1 - Equacionamento Básico do Nível do Molde                                       | 103       |
| 5.3.2 - Representação dos Distúrbios de <i>Bulging</i> , <i>Clogging</i> e de Argônio | 105       |
| 5.3.3 - O Modelo Neural do Nível do Molde   | 112       |

|  |            |
|--|------------|
| 5.3.4 - Controle Preditivo Neural            | 117        |
| 5.3.5 - GPC para o Modelo Hammerstein (HGPC) | 122        |
| 5.3.6 - HGPC com Ruído de Argônio            | 127        |
| 5.4 - Conclusão do Capítulo                  | 129        |
| <b>6 - Conclusões Gerais</b>                 | <b>132</b> |
| <b>- Referências Bibliográficas</b>          | <b>135</b> |
| Apêndice A                                   | 147        |
| Apêndice B                                   | 151        |

# LISTA DE FIGURAS

|   |    |
|---|----|
| 1.1 Horizonte de Predição   | 2  |
| 1.2 Estrutura de um Controlador Preditivo                             | 3  |
| 1.3 Laminador de 3 Cadeiras   | 6  |
| 1.4 Curva de Controle Dimensional na Laminação                        | 7  |
| 1.5 Esquema da Torre Giratória  | 9  |
| 1.6 Representação da Máquina de Lingotamento Contínuo                 | 10 |
| 1.7 Representação da Passagem de Aço do Distribuidor ao Molde         | 11 |
| 2.1 Modelo de Hammerstein   | 15 |
| 2.2 Rede MLP com Camada Intermediária                                 | 22 |
| 2.3 Fluxo de Processamento do Algoritmo de Retropropagação do Erro    | 23 |
| 2.4 Rede Neural com estrutura NNARX                                   | 28 |
| 2.5 Identificação de uma planta utilizando RNA                        | 29 |
| 2.6 Estrutura de identificação série-paralelo                         | 30 |
| 2.7 Configuração de um Simplex Algoritmo Genético                     | 35 |
| 2.8 Operação de Crossover Realizada em um Ponto                       | 39 |
| 2.9 Operação de Crossover Multi Pontos                                | 40 |
| 2.10 Operação de Mutação  | 40 |
| 2.11 Operação de Reprodução   | 41 |
| 4.1 Sinal de Resposta $y, u$ e $\Delta u$ para $G_1$ com $\mu = 0, 1$ | 69 |
| 4.2 Sinal de Resposta $y, u$ e $\Delta u$ para $G_1$ com $\mu = 3$    | 70 |
| 4.3 Sinal de Resposta $y, u$ e $\Delta u$ para $G_1$ com $\mu = 10$   | 70 |

|  |     |
|--|-----|
| 4.4 Sinal de Resposta $y, u$ e $\Delta u$ para $G_2$   | 71  |
| 4.5 Sinal de Resposta $y, u$ e $\Delta u$ para $G_3$ com $\mu = 0,2$   | 73  |
| 4.6 Sinal de Resposta $y, u$ e $\Delta u$ para $G_3$ com $\mu = 0,7$   | 73  |
| 4.7 Sinal de Resposta $y, u$ e $\Delta u$ para $G_4$   | 74  |
| 4.8 Sinal de Resposta $y_1, u_1$ e $\Delta u_1$  | 76  |
| 4.9 Sinal de Resposta $y_2, u_2$ e $\Delta u_2$  | 76  |
| 4.10 Cadeiras de Laminação   | 77  |
| 4.11 Variação da espessura e temperatura das duas Cadeiras de Laminação.   | 85  |
| 4.12 Variação da Espessura de saída das duas Cadeiras de Laminação com FFF e GPC   | 86  |
| 4.13 Espessura de saída das duas Cadeiras de Laminação.  | 87  |
| 5.1 Diagrama do Controle Preditivo Neural  | 93  |
| 5.2 A Estrutura Usada em uma TDNN  | 94  |
| 5.3 Diagrama de Bloco do Controle Preditivo Neural   | 96  |
| 5.4 Configuração Básica do Sistema de Controle   | 102 |
| 5.5 Movimento e Formato da Válvula Gaveta do Distribuidor.   | 104 |
| 5.6 Diagrama de Blocos da Malha Completa de Controle de Nível do Molde   | 105 |
| 5.7 Comportamento do efeito <i>Clogging</i> na malha de controle   | 107 |
| 5.8 a) Diagrama de bloco com a inclusão do efeito de clogging; b) Representação gráfica da função de redução da área $A_{clg}$ | 108 |
| 5.9 Efeito Bulging na MLC03  | 110 |
| 5.10 Ponto de Inclusão do Efeito Bulging no Processo   | 110 |
| 5.11 Simulação do borbulhamento de Argônio em um molde com água  | 111 |

|      |   |     |
|------|---|-----|
| 5.12 | Espectro de frequência de borbulhamento de argônio num molde real         | 112 |
| 5.13 | Dados de entrada e saída utilizados na identificação do modelo.           | 113 |
| 5.14 | Curso x Área Efetiva da Válvula Gaveta.                                   | 114 |
| 5.15 | Efeito da ordem dos regressores no índice de avaliação proposto por [57]. | 115 |
| 5.16 | Validação do Modelo Neural  | 116 |
| 5.17 | Saída real e saída prevista pela rede neural                              | 117 |
| 5.18 | Resposta ao Degrau com o Controlador PNLI                                 | 118 |
| 5.19 | Resposta ao Degrau com o Controlador PNRG                                 | 119 |
| 5.20 | Simulação com Controlador PNLI e Distúrbio de <i>Bulging</i>              | 120 |
| 5.21 | Simulação com Controlador PNRG e Distúrbio de <i>Bulging</i>              | 120 |
| 5.22 | Simulação com Controlador PNLI e Clogging com $f=0,8$                     | 122 |
| 5.23 | Simulação com Controlador PNRG e Clogging com $f=0,75$                    | 122 |
| 5.24 | Saída real e saída identificada do Modelo Hammerstein                     | 123 |
| 5.25 | Saída real e saída prevista com o modelo Hammerstein                      | 124 |
| 5.26 | Resposta ao Degrau com Controlador HGPC                                   | 125 |
| 5.27 | Resposta do Sistema com o Distúrbio de <i>Bulging</i>                     | 125 |
| 5.28 | Resposta do Sistema com o Distúrbio de Clogging                           | 126 |
| 5.29 | Resposta do Sistema com o Distúrbio de Argônio sem FKE                    | 129 |
| 5.30 | Resposta do Sistema com o Distúrbio de Argônio com FKE                    | 129 |

## LISTA DE TABELAS

|   |     |
|---|-----|
| 3.1 Aplicações Comerciais de MPC                            | 46  |
| 4.1 Parâmetros do DMC para $G_1$ .                          | 69  |
| 4.2 Parâmetros do DMC para $G_2$                            | 71  |
| 4.3 Parâmetros do DMC para $G_3$ .                          | 72  |
| 4.4 Parâmetros do DMC para $G_4$                            | 74  |
| 4.5 Parâmetros do DMC para o sistema MIMO                   | 75  |
| 5.1 Aplicações Comerciais de NMPC                           | 89  |
| 5.2 Comparação do Desempenho dos Controladores Não Lineares | 130 |

# LISTA DE ABREVIACÕES

|           |  |
|-----------|--|
| MPC       | Controle Preditivo por Modelo                        |
| NMPC      | Controle Preditivo por Modelo Não-Linear             |
| AMT       | Aercelor Mittal Tubarão                              |
| GPC       | Controle Preditivo Generalizado                      |
| FFF       | Force Feed Forward                                   |
| DMC       | Controle por Matriz Dinâmica                         |
| AG        | Algoritmo Genético                                   |
| SISO      | Sistemas Monovariáveis                               |
| MIMO      | Sistemas Multivariáveis                              |
| $N_u$     | Horizonte de Controle                                |
| $N_2$     | Horizonte de Controle                                |
| $\alpha$  | Constante da trajetória de referência                |
| $\delta$  | Ponderação do erro de previsão                       |
| $\lambda$ | Fator de supressão                                   |
| RN        | Redes Neurais  |
| MLP       | Perceptron de Multicamadas                           |
| PDP       | Processamento Paralelo Distribuído                   |
| CARMA     | Controller Auto-Regressive Moving-Average            |
| CARIMA    | Controlled Auto-Regressive Integrated Moving Average |
| IAE       | Integral do valor absoluto do erro                   |

|                  |  |
|------------------|--|
| ISE              | Integral do erro quadrático                                |
| ITAE             | Integral do valor absoluto do erro multiplicado pelo tempo |
| FOPDT            | Primeira Ordem com Tempo Morto                             |
| NR               | Número de Restrições                                       |
| HGPC             | Controle Preditivo Generalizado com Modelo Hammerstein     |
| NPC              | Controle Preditivo Neural                                  |
| PNRG             | Controle Preditivo Neural Baseado na Regra do Gradiente    |
| PNLI             | Controle Preditivo Neural com Linearização Instantânea     |
| FK               | Filtro de Kalman   |
| FKE              | Filtro de Kalman Estendido                                 |
| TP               | Tempo de Processamento                                     |
| $H_{max}$        | Altura Máxima de Controle                                  |
| $R_{max}$        | Degrau Máximo Aplicado                                     |
| $\Delta H_{max}$ | Varição Máxima de Altura                                   |
| $f_{max}$        | Entupimento Máximo   |

# Capítulo 1

## INTRODUÇÃO

Controle Preditivo por Modelo (MPC) originou-se nos anos 70 e teve um desenvolvimento considerável desde então. O termo MPC não designa uma estratégia de controle específica, mas uma ampla variedade de métodos de controle que fazem uso explícito do modelo de um processo para obter o sinal de controle pela minimização de uma função objetivo [14]. Os vários algoritmos de MPC diferem-se apenas quanto ao uso da representação do modelo do processo, dos ruídos e da função custo a ser minimizada. Este tipo de controle é de natureza aberta, dentro do qual muitos trabalhos têm sido desenvolvidos, tais como: (Controle Preditivo Generalizado, [19]), (Controle por Matriz Dinâmica, [25]), (Controle Auto Adaptativo Estendido Preditivo, [68]), (Controle Preditivo Funcional, [96]), (Controle Adaptativo com Horizonte Estendido, [119]), (Algoritmo de Controle por Modelo, [101]) entre outros, sendo amplamente utilizados tanto pela área acadêmica quanto pela indústria.

A metodologia de todos os controladores pertencentes à família MPC é caracterizada pela seguinte estratégia, representada na Figura 1.1.

- a) As saídas futuras para um determinado horizonte de predição,  $N_2$  denominado horizonte de predição, são estimadas a cada instante de tempo  $t = k$  usando o modelo do processo.
- b) O conjunto de sinais das ações futuras do controlador é calculado por otimização, atendendo a um determinado critério, de forma a manter o processo o mais perto possível da trajetória de referência  $w(t)$  a qual pode ser o próprio *set point* ou uma aproximação do mesmo. Este critério toma, em geral, a forma de

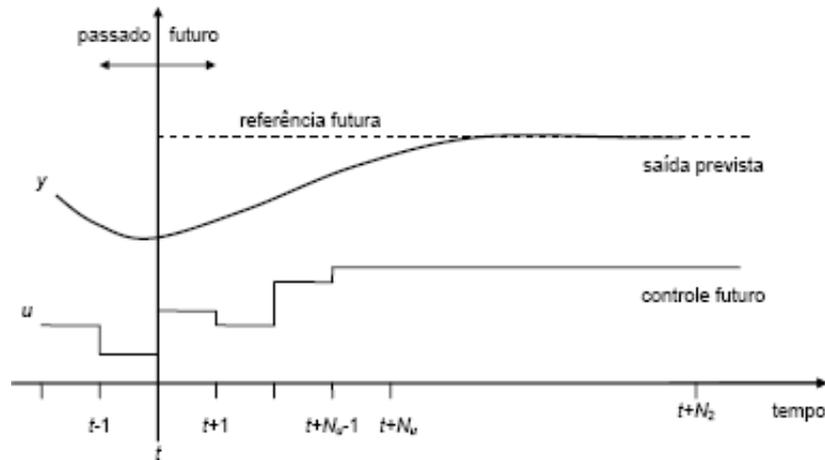


Figura 1.1: Horizonte de Predição.

uma função quadrática do erro entre o valor predito para a saída do processo e a trajetória de referência proposta. O esforço do controlador é, muitas vezes, incluído na função objetivo. Uma solução explícita pode ser obtida se o critério é quadrático, o modelo é linear e não existem restrições; caso contrário um método de otimização iterativo tem que ser usado.

- c) O primeiro sinal da sequência das ações de controle  $u(t)$  é enviado ao processo, enquanto que o restante da sequência é descartada.

Para implementar esta estratégia, a estrutura básica apresentada na Figura 1.2 é usada. Um modelo é necessário para prever as saídas futuras da planta, baseado nos seus valores presente e passado e nas ações de controle futuras propostas pela otimização. Estas ações são calculadas minimizando a função custo com as restrições impostas ao sistema.

O modelo do processo tem um papel decisivo no desempenho do controlador, devendo ser capaz de capturar a dinâmica do processo de forma a prever precisamente as saídas futuras, assim como ser simples na sua implementação e entendimento. O MPC é uma técnica que engloba diferentes metodologias, pois utiliza vários tipos de modelos, nas mais diferentes formulações.

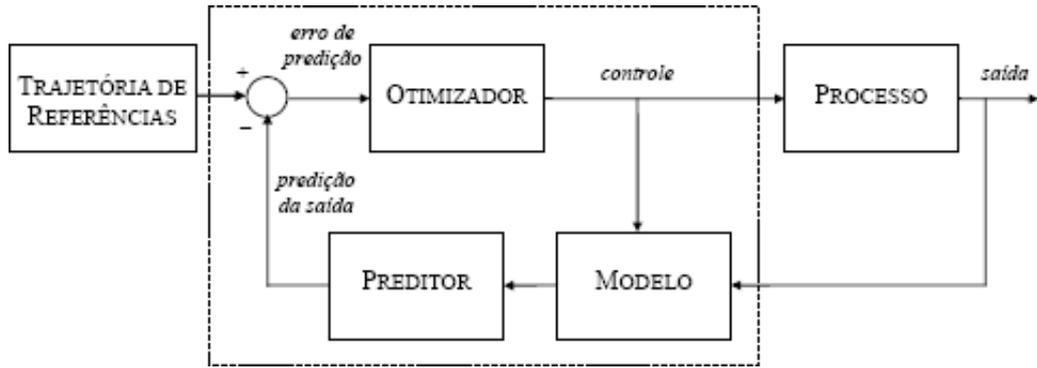


Figura 1.2: Estrutura de um Controlador Preditivo.

## A Função Objetivo

Os vários algoritmos MPC propõem diferentes funções custo para obter a lei de controle. O objetivo principal é que a predição da saída futura  $\hat{y}$  no horizonte considerado deve seguir um determinado sinal de referência  $w$  e, ao mesmo tempo, o esforço do controlador  $\Delta u$  necessário para isto deve ser penalizado. A expressão geral para tal função objetivo deve ser:

$$J = \sum_{k=N_1}^{N_2} \delta [\hat{y}(t+k) - w(t+k)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(t+j-1)]^2 \quad (1.1)$$

Em alguns algoritmos, o segundo termo, que considera o esforço do controlador, não é levado em conta, enquanto que em outros como no UPC (*Unified Predictive Control*) [111], os valores do sinal do controlador, e não o seu incremental, também aparecem diretamente na formulação da função custo. As considerações que podem ser efetuadas na função custo são apontadas a seguir.

## Parâmetros

$N_1$  e  $N_2$  são, respectivamente, o valor mínimo e o valor máximo do horizonte de predição da função custo e  $N_u$  é o horizonte de controle, o qual não é necessariamente igual ao horizonte máximo, como será visto depois.  $N_1$  marca o instante no qual é desejado que a saída do processo siga a referência. Nota-se que em processos com um tempo morto  $d$  não existe razão para  $N_1$  ser menor do que  $d$ , porque a saída não

começará a evoluir até o instante  $t + d$ . Além disto, se o processo possui resposta inversa, sistema de fase não mínima, este parâmetro irá permitir que os primeiros instantes da resposta inversa sejam eliminados da função objetivo. Os coeficientes  $\delta$  e  $\lambda$  são parâmetros que afetam a estabilidade e a robustez do controlador.

## Trajetória de Referência

Uma das vantagens do controle preditivo é que se a trajetória de referência futura é conhecida *a priori*, o sistema reage com antecedência às alterações da referência, seguindo a trajetória especificada com mais suavidade. A referência futura  $r(t)$  é especificada em muitas aplicações como robótica, processos servo ou bateladas. Na minimização, a maioria dos métodos usa uma trajetória de referência que não coincide necessariamente com a referência real. Ela é, normalmente, uma aproximação suave do valor real da saída  $y(t)$  em direção à referência conhecida por meio de um sistema de primeira ordem:

$$w(t) = y(t) \tag{1.2}$$

$$w(t) = \alpha w(t - 1) + (1 - \alpha)r(t) \quad t = 1, \dots, N \tag{1.3}$$

na qual  $\alpha$  é um parâmetro entre 0 e 1 que suaviza a resposta do sistema à medida que se aproxima de 1.

O modelo do processo é uma das partes mais importantes, pois deve ser capaz de capturar a dinâmica do processo para realizar uma predição precisa da saída futura, além de ser simples de implementar, analisar e entender.

O otimizador é outra parte fundamental para esta estratégia de controle. Se a função custo é quadrática, seu mínimo pode ser obtido como uma função explícita de entradas e saídas passadas e a trajetória da referência futura. Quando restrições são consideradas, a solução tem que ser obtida computacionalmente através de algoritmos de otimização.

Existem muitas aplicações de controle preditivo, não somente em processos industriais, mas também em uma diversidade de processos como: robôs manipuladores [71], colunas de destilação, plantas PVC, etc [94].

## 1.1 Aplicações do MPC

O Controlador Preditivo é encontrado tanto em processos lineares quanto não-lineares, mono e multivariáveis, com restrições e atrasos nas variáveis de entrada e saída (veja [14], [78] e [97]); desde aqueles com dinâmica estável, até aqueles de fase não mínima e instáveis em malha aberta. O MPC trabalha muito eficaz com restrições devido às limitações nos atuadores, sensores, sinais de controle, saída, as quais podem ser sistematicamente introduzidas durante o processo de otimização do controlador.

As aplicações de controle preditivo em processos químicos e petroquímicos [94] tem sido bem sucedido, tendo em vista que na maioria destes processos, a resposta ao degrau é estável em malha aberta, o que facilita a sintonia do controlador. Em sistemas cuja resposta ao degrau é instável, a sintonia é um fator que inibe a aplicação do Controlador Preditivo. Por isto, observa-se na literatura que a aplicação do MPC em sistemas instáveis em malha aberta, lineares ou não, é ainda pouco explorada.

Nesta tese será estudado a aplicação do MPC em dois processos instáveis em malha aberta, os quais são encontrados na indústria Siderúrgica, sendo eles: Lingotamento Contínuo e Laminação de tiras.

### O Laminador

O processo de laminação consiste em passar a tira entre dois conjuntos de cilindros rotativos afastados de uma distância inferior à espessura atual da tira, de modo a produzir uma redução da sua espessura.

Durante o processo de laminação, raramente passa-se o material somente uma vez entre os cilindros ou rolos de laminação, pois a redução de área almejada normalmente não pode ser conseguida em um só passe devido à necessidade de carga ser muito grande. Assim, o equipamento de laminação deve ser capaz de submeter o material a uma sequência de passes [59].

O controle de espessura de saída na laminação de tiras envolve ações de controle em subsistemas componentes do laminador como o subsistema do *gap* (abertura e fechamento do espaço entre os cilindros), ou de velocidade de laminação, ou

das tensões aplicadas a frente e a ré, ou outras atitudes que mantenham o valor da espessura de saída dentro dos limites de tolerância aceitáveis relativamente ao valor desejado.

Na Figura 1.3, tem-se um esquema de um laminador simplificado com seus subsistemas principais, tais como: de ajuste do *gap*, de debobinamento, de velocidade de laminação e de bobinamento.

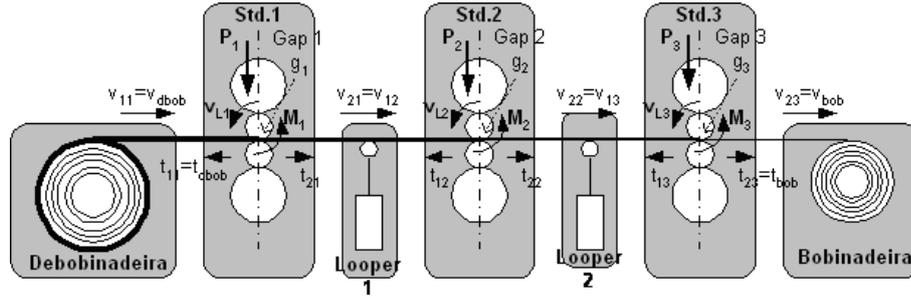


Figura 1.3: Laminador de 3 Cadeiras.

na qual:

- $P_j$  é a carga de laminação;
- $M_j$  é o conjugado de laminação;
- $v_{Lj}$  é a velocidade de laminação;
- $t_{ij}$  tensão na tira  $i = 1$  a ré e  $i = 2$  a frente, no *stand*  $j$ ;
- $v_{ij}$  velocidade da tira  $i = 1$  a ré e  $i = 2$  a frente, no *stand*  $j$

O processo de laminação se desenvolve segundo uma curva de operação, mostrada na Figura 1.4 que relaciona a carga de laminação e a espessura de saída conforme Equação 1.4.

$$P = E_{LM}(h_f - g) \quad (1.4)$$

no qual:

- $E_{LM}$  é o módulo de rigidez do laminador;

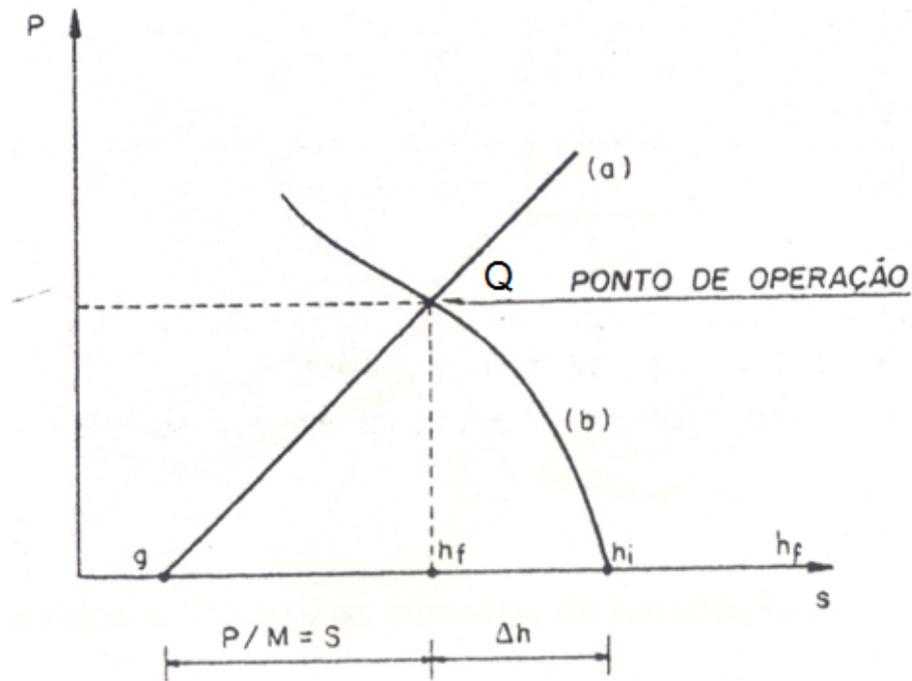


Figura 1.4: Curva de Controle Dimensional na Laminação: a) carga-deformação do laminador, b) carga-espessura final da chapa.

- $h_f$  é a espessura de saída;
- $g$  é o *gap*.

Esta curva é função dos parâmetros como coeficiente de atrito  $\mu$ , tensão de escoamento  $S$ , tensão a frente  $t_1$ , tensão a ré  $t_2$ , espessura de entrada  $h_i$ , velocidade de laminação  $v_l$  e outros. O ponto de funcionamento depende do *gap* entre os cilindros e da rigidez do laminador, o que é traduzida pela reta de carga  $r$  do sistema. Assim, com distúrbios nos parâmetros operacionais, ocorre a modificação na curva de operação e, conseqüentemente, na espessura de saída.

Há duas formas básicas de correção de  $h_f$ , os quais sejam: alteração do *gap* ou variação compensatória de parâmetros.

#### **Correção de $h_f$ por Ajuste no *Gap*:**

Esta ação consiste em modificar o *gap*, o que equivale a transladar a reta de carga de modo a recuperar o ponto de operação  $Q$ . Vários esquemas de controle

podem ser encontrados normalmente adotando realimentação da espessura de saída ou por modelo teórico como em [42], [44] e [45].

### **Correção de $h_f$ por Variação Compensatória de Parâmetros Operacionais:**

Esta ação consiste em modificar outro parâmetro operacional acessível e de comportamento conhecido de modo a recuperar o ponto de funcionamento  $Q$ . Também neste caso encontram-se vários esquemas de controle. Dentre as técnicas possíveis, tem-se [43] e [41], um método de predição do montante de ação corretiva.

Encontra-se também trabalhos como [48], no qual foi usado um controle preditivo *feed-forward* com Filtro de Kalman que estima a espessura local e a variação de temperatura no processo de laminação. O controle preditivo *feed-forward* também foi aplicado para o controle de temperatura do bobinamento em [76] e para o controle final de temperatura em [115] e [77].

Encontra-se na literatura vários trabalhos sobre a aplicação do MPC Multivariável no processo de Laminação, conforme pode ser visto em [13] e [17]. No entanto, nenhum deles utiliza o método de sintonia dos parâmetros realizada nesta tese, que é baseada em algoritmos genéticos. Além disto, será analisado o comportamento do laminador diante de distúrbios provocados pela variação de temperatura, cuja compensação foi estudada através de outras estratégias como redes neurais e controle ótimo (veja [98]).

## **O Molde do Lingotamento Contínuo**

O Controle do Nível do Molde do Lingotamento Contínuo é o outro processo que será estudado. O processo de lingotamento contínuo consiste, basicamente, da transformação do aço, inicialmente em seu estado líquido e temperatura superiores a  $1.600^{\circ}\text{C}$ , em placas de aço de formato, espessura, superfície, consistência e propriedades mecânicas conhecidas e controladas segundo padrões existentes.

O processo começa pela Torre Giratória que tem a função de receber a panela de aço cheia, liberada pelo convertedor, processo anterior de formação do aço a partir do Ferro Gusa, sustentando-a sobre o distribuidor. A panela vazia é liberada para a área de preparação através de um giro de  $180^{\circ}$ , permitindo a troca rápida de

panelas e dando continuidade ao processo. A Figura 1.5 mostra o detalhamento de uma Torre Giratória.

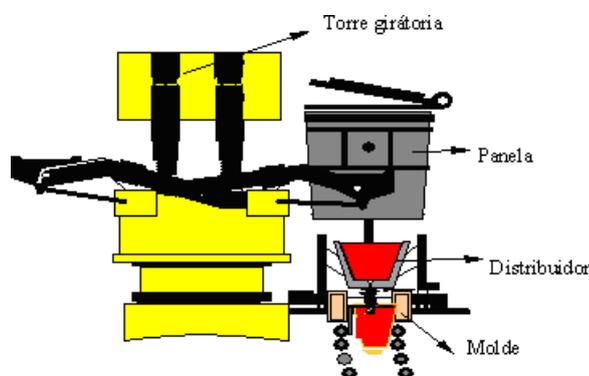


Figura 1.5: Esquema da Torre Giratória.

A Torre Giratória permite que o aço seja distribuído em dois moldes através de um dispositivo denominado Distribuidor. Sua função é manter a alimentação de aço para a máquina durante a troca de panelas, realizada por meio de peças refratárias, válvulas submersas, adaptadas ao distribuidor que evitam o contato do aço com o ar atmosférico e orientam o fluxo de aço para o interior dos moldes.

O objetivo do molde é extrair calor do aço promovendo a primeira solidificação a partir da superfície da placa, através do resfriamento primário feito pela circulação de água em circuito fechado em serpentinas internas às placas do molde e direcionar o aço aos segmentos.

A vazão de aço entre o distribuidor e o molde é regulada pela abertura de uma válvula própria para este objetivo. A temperatura interior do distribuidor para se iniciar o lingotamento deve ser de 1000 a 1100<sup>o</sup>C.

A máquina de lingotamento contínuo possui 14 segmentos constituídos de rolos superiores e inferiores por onde a placa é conduzida até a sua completa extração por meio de tracionamento por dois rolos motrizes em cada um destes segmentos. A função dos rolos é extrair e manter a forma da placa, além de suportar a pressão ferrostática <sup>1</sup> do interior ainda em solidificação e resfriar a placa através de jatos de água, resfriamento secundário e desencurvamento progressivo da placa.

<sup>1</sup>Pressão interna às paredes de aço solidificado na superfície da placa de aço, provocada pela força de expansão do aço em estado líquido em seu interior, proporcional à altura correspondente.

No percurso do aço entre os rolos dos segmentos, utilizam-se sprays de água e ar, fazendo com que a camada solidificada, pele, vá tornando-se mais espessa e ao chegar na saída do segmento 14, parte reta da máquina, o aço já se encontra totalmente solidificado. Após deixar o veio da máquina, o aço solidificado é cortado em dimensões predeterminadas através de um maçarico automático sendo agora denominado placa. A constituição básica da máquina está evidenciada na Figura 1.6.

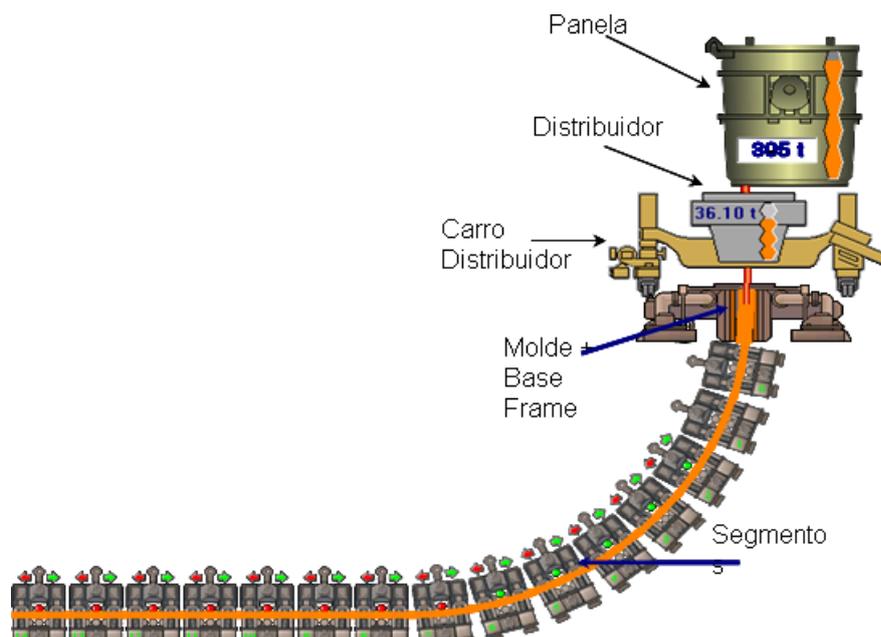


Figura 1.6: Representação da Máquina de Lingotamento Contínuo.

Para uma correta refrigeração no molde e formação da pele sem defeitos é necessário manter um nível constante e predeterminado de aço líquido no seu interior em diferentes condições de processo. Na Figura 1.7 verificam-se os detalhes da passagem de aço do distribuidor ao molde, com o fluxo de aço controlado por uma válvula gaveta, tipo de válvula utilizado na usina da Arcelor Mittal, de forma que o nível no interior do molde seja mantido.

Apesar da aplicação de controlador clássico PID ser ainda encontrado no controle do molde, a sua frágil robustez diante dos efeitos e variações de parâmetros presentes no processo de lingotamento tem impulsionado pesquisadores do mundo inteiro ao desenvolvimento, tanto na siderurgia quanto no meio acadêmico, de téc-

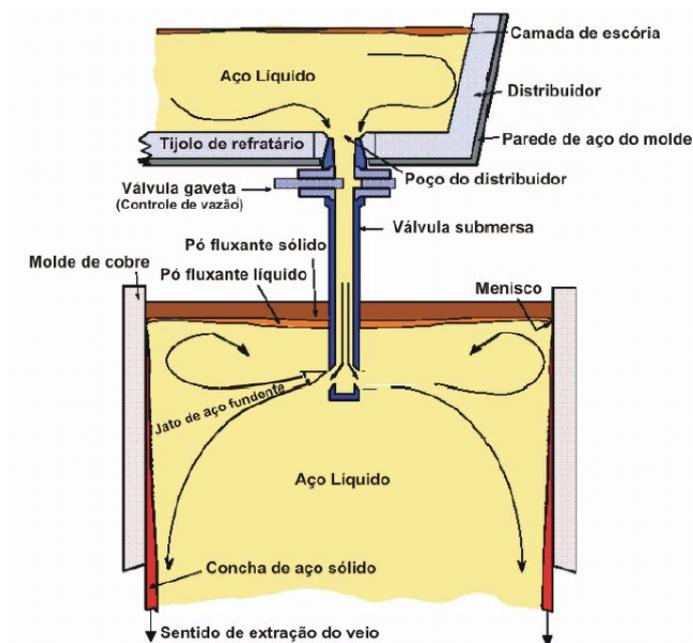


Figura 1.7: Representação da Passagem de Aço do Distribuidor ao Molde.

nicas mais sofisticadas de controle. No trabalho desenvolvido em [120] é mostrado a aplicação de técnica de "controle por modo deslizante". No artigo [69] é desenvolvido um controlador  $H_\infty$  e em [113] é utilizado a Lógica Fuzzy e Redes Neurais para resolver o problema de controle do Nível do Molde. Não se encontram na literatura estudos sobre aplicações de Controle Preditivo Não-linear do nível do molde do lingotamento contínuo. Os artigos existentes sobre este assunto estão relacionados ao controlador GPC para o molde linear, [75] e [105]. Nesta tese será realizado o controle do nível do molde, utilizando técnicas de controle preditivo não-lineares.

## 1.2 Métodos de Sintonia da Literatura

Os MPCs geralmente exibem um bom desempenho e robustez, desde que os parâmetros tenham sido sintonizados corretamente. Entretanto, a seleção destes parâmetros não é uma tarefa fácil, na medida em que não há nenhuma metodologia precisa que assegure a estabilidade [10]. O método mais usado na literatura para o ajuste dos parâmetros dos MPC é o tentativa e erro, exigindo, além de um bom conhecimento do processo por parte do operador, também um certo tempo até que

a resposta do sistema seja satisfatória. Muitas vezes, como nas aplicações estudadas nesta tese, não se consegue obter os parâmetros através de tentativa e erro a fim de obter uma resposta otimizada.

Um segundo método utiliza os Algoritmos Genéticos para sintonizar os parâmetros do MPC. Esta metodologia foi inicialmente testada no Controlador Preditivo Generalizado (GPC) por [40], aplicado a sistemas monovariáveis e irrestritos. Porém, na sintonia feita em [40] foram considerados apenas os horizontes de controle e de previsão, deixando-se de lado os outros parâmetros ( $\alpha$ ,  $\delta$  e  $\lambda$ ). Em [28] mostra-se que a inclusão destes parâmetros na sintonia diminui os horizontes de controle e previsão, o que facilita o esforço computacional realizado na obtenção da ação de controle.

Outro método, apresentado em [78], é restrito a alguns sistemas monovariáveis, sendo os parâmetros determinados de forma heurística, e dependentes também do conhecimento do operador sobre o processo. Este método não é válido para todos os tipos de plantas, principalmente no caso multivariável com restrições, desta forma, é pouco utilizado na literatura.

Na metodologia proposta por [34], a qual fornece explicitamente valores para os parâmetros  $N_u$ ,  $N_1$ ,  $N_2$ , considera-se que o processo seja estável e tenha função de transferência de primeira ordem com atraso. Além disto, este método fornece horizontes de controle grandes, o que afeta o desempenho do controlador, e aumenta a quantidade de restrições, levando à obtenção de soluções não factíveis.

Nesta tese será utilizado Algoritmos Genéticos na sintonia dos parâmetros do Controlador Preditivo, mostrando a potencialidade desta técnica em relação a desenvolvida por [34]. Além disto, este trabalho apresenta um estudo original da aplicação do controle preditivo não-linear no nível do molde do lingotamento contínuo.

### 1.3 Objetivos da Tese

Esta tese consiste de um estudo das estratégias de controle preditivo linear e não-linear monovariável e multivariável, cujos principais objetivos são:

- Usar duas técnicas de Controle Preditivo Neural. Uma delas utiliza Lineari-

zação Instantânea do Modelo Neural do Molde. A outra utiliza o Gradiente da Função Objetivo para determinar a ação de controle. Será também aplicado no Controle de Nível do Molde a técnica de Controle Preditivo Não-linear com Modelo Hammerstein; a contribuição referente a aplicação do Preditivo utilizando Redes Neurais com Linearização Instantânea foi aceito para publicação em um Capítulo do Livro "Computação Evolucionária em Problemas de Engenharia", a ser editado por Omnipax Editora Ltda.

- Comparar os métodos de Controle Preditivo Não-linear aplicado ao controle do nível do molde; quanto a robustez; estabilidade e qualidade de controle. A comparação desses métodos e a aplicação no nível do molde não são encontrados em nenhuma literatura;
- Realizar o Controle Multivariável do Processo de Laminação Linearizado com perturbações através do GPC e compará-lo com a técnica FFF (Feed Forward Forced) que é a técnica aplicada no processo real. Este trabalho foi publicado no anais do IX Simpósio Brasileiro de Automação Inteligente de 2009 [29].
- Comparar técnicas de sintonia do DMC existente na literatura com o Algoritmo Genético. Este trabalho foi publicado no Periódico Internacional Latin American Applied Research em 2009 [31] e no anais do XXVII Congresso Brasileiro de Automação de 2008 [30].

## 1.4 Organização da Tese

Esta tese está dividida em 6 capítulos:

No Capítulo Dois são abordados os conceitos básicos da teoria de identificação usando modelos Hammerstein e Redes Neurais, bem como são introduzidas as definições e a estrutura do Algoritmo Genético necessários ao entendimento desta tese.

O Capítulo Três explana sobre o Controle Preditivo Baseado em Modelos Lineares, mais precisamente o DMC e o GPC mono e multivariável.

No Capítulo Quatro serão apresentadas as técnicas utilizadas para sintonia do MPC linear. Neste capítulo, será mostrada também uma comparação das técnicas de sintonia do DMC por AG apresentado em [31] e [34] para sistemas SISO

*benchmark* e MIMO restrito. Por último, será feita uma aplicação no processo de laminação, cujo modelo é levantado através de dados reais para realizar o controle GPC multivariável.

No Capítulo Cinco é feita uma abordagem das técnicas de controle preditivo não-lineares usando modelos Hammerstein e Redes Neurais. Os algoritmos de controle serão aplicados no simulador do nível do molde do Lingotamento Contínuo da Arcelor Mital, cujo modelo é obtido através de dados reais. Em seguida é feita uma comparação dos controladores não lineares quanto a estabilidade e robustez aos distúrbios presentes no processo.

No Capítulo Seis serão apresentadas as conclusões e trabalhos futuros.

# Capítulo 2

## Revisão Bibliográfica

Neste capítulo são abordados os conceitos e definições da teoria de modelos não lineares e Algoritmo Genético necessários ao entendimento desta tese. Primeiro serão mostrados os modelos não-lineares do tipo Hammerstein e as Redes Neurais, e a forma de realizar a identificação destes modelos. Em seguida é feita uma introdução sobre Algoritmos Genéticos.

### 2.1 Modelo Hammerstein

Este modelo consiste de um elemento estático não-linear seguido por um sistema dinâmico linear como ilustrado na Figura 2.1.

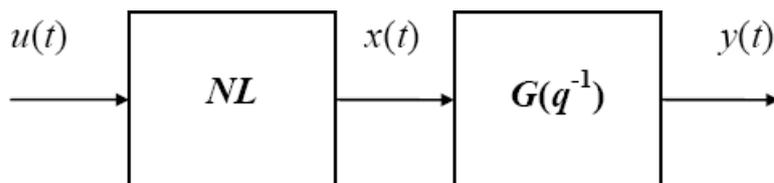


Figura 2.1: Modelo de Hammerstein.

O bloco da não-linearidade estática (NL) pode ser representado por um polinômio, pela equação da não-linearidade ou por modelos semi-paramétricos.

#### Representação da NL por um Polinômio

Este caso é o mais comum quando não se dispõe de informações a respeito da natureza da não-linearidade, aproximado-a por uma expansão polinomial finita

do tipo.

$$x(t) = \gamma_1 u(t) + \gamma_2 u^2(t) + \dots + \gamma_m u^m(t) \quad (2.1)$$

onde  $t$  é o instante de tempo,  $x(t)$  é a pseudo-saída, não-mensurável, do bloco não-linear,  $u(t)$  é a variável de entrada,  $\gamma_i$  ( $i = 1, \dots, m$ ) representam os coeficientes do polinômio e  $m$  é o grau de não-linearidade do modelo ([12]). Normalmente considera-se  $\gamma_1 = 1$  transferindo o ganho estático para a parcela dinâmica linear,  $G(q^{-1})$ , que pode ser representada por qualquer um dos modelos lineares (paramétricos ou não) apresentados em [106]. No caso desta tese, o modelo linear será do tipo CARMA ou CARIMA:

### Modelo CARMA (*Controlled Auto-Regressive Moving Average*)

É representado pela estrutura da Equação 2.2, isto é,

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + C(q^{-1})\xi(t) \quad (2.2)$$

na qual:

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + \dots + a_{na} q^{-na}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + b_2 q^{-2} + \dots + b_{nb} q^{-nb}$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2} + \dots + c_{nc} q^{-nc}$$

$y(t)$  é a saída do sistema,  $u(t)$  é o sinal de controle (entrada),  $\xi(t)$  é uma sequência aleatória que pode representar incertezas de modelagem, erros de medição ou ruídos presentes na saída e  $d$  é o atraso de transporte discreto com  $dT_s \leq L \leq (d+1)T_s$  e  $L$  é o atraso de transporte contínuo [73].

### Modelo CARIMA (*Controlled Auto-Regressive Integrated Moving Average*)

É representado pela seguinte equação a diferenças:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + C(q^{-1})\xi(t)/\Delta \quad (2.3)$$

que pode ser reescrita na forma:

$$A(q^{-1})\Delta y(t) = q^{-d}B(q^{-1})\Delta u(t) + C(q^{-1})\xi(t) \quad (2.4)$$

sendo  $\Delta = (1 - q^{-1})$  de maneira que  $\Delta y(t) = y(t) - y(t - 1)$ .

A popularidade do modelo de Hammerstein deve-se ao fato da maior simplicidade em relação às representações de Volterra e Bilinear aliada a uma capacidade de representação da não-linearidade da maioria dos processos práticos podendo representar processos com atuadores não-lineares e ganhos variantes [107].

A literatura de controle e identificação de sistemas conta com inúmeras aplicações do modelo de Hammerstein na representação de processos de fermentação [102], colunas de destilação [91], trocadores de calor [36]; [46], processos de nível [67]; [22], turbina de gás [16], motor de corrente contínua [65], reatores químicos [66]; [2], controle de pH [123]; [49]; [124], motor diesel [92], além de qualquer processo do Tipo Hammerstein, que possa ser representado por uma não-linearidade seguida de uma parcela dinâmica linear [61]; [21]. Apenas a título de revisão da teoria de sistemas não lineares, no Apêndice B encontra-se uma discussão sobre outros tipos de modelos tais como NCARMA, Volterra, Bilinear e Wiener.

## 2.2 Identificação do Modelo Hammerstein através do Método dos Mínimos Quadrados

Definindo-se o vetor,  $\phi(t)$ , com dimensão  $[(na + nb + 1)m \times 1]$ , para o caso linear:

$$\begin{aligned} \phi^T(t) &= [-y(t-1) \dots -y(t-na) \ u(t-d) \dots \\ &\dots \ u(t-d-nb) \ u^2(t-d) \dots u^2(t-d-nb) \dots u^m(t-d-nb)] \end{aligned} \quad (2.5)$$

e o vetor de parâmetros,  $\theta(t)$ , com dimensão  $[(na + nb + 1)m \times 1]$

$$\hat{\theta} = [a_1 \ a_2 \dots a_{na} \ b_0\gamma_1 \ b_1\gamma_1 \dots b_{nb}\gamma_1 \ b_0\gamma_2 \ b_1\gamma_2 \dots b_{nb}\gamma_2 \dots b_0\gamma_m \ b_1\gamma_m \dots b_{nb}\gamma_m]^T \quad (2.6)$$

tem-se a seguinte expressão:

$$y(t) = \phi^T(t)\theta(t) + \epsilon(t) \quad (2.7)$$

que é denominado modelo de regressão linear [23], [1]. Admitindo-se que a realização de  $N$  medidas são suficiente para determinar os parâmetros  $\theta$ , então tem-se que:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} \phi^T(0) \\ \phi^T(1) \\ \vdots \\ \phi^T(N-1) \end{bmatrix} \theta + \begin{bmatrix} \epsilon(0) \\ \epsilon(1) \\ \vdots \\ \epsilon(N-1) \end{bmatrix} \quad (2.8)$$

A representação matricial de (2.8) é:

$$Y = \Phi\theta + \epsilon \quad (2.9)$$

e a matriz de observação é:

$$\Phi = \begin{bmatrix} -y(-1) & \dots & -y(-na) & u(-d) & \dots \\ -y(0) & \dots & -y(1-na) & u(1-d) & \dots \\ -y(1) & \dots & -y(2-na) & u(2-d) & \dots \\ \vdots & \dots & \vdots & \vdots & \dots \\ -y(N-2) & \dots & -y(N-na-1) & u(N-d-1) & \dots \\ \dots & u(-d-nb) & u^2(-d) & \dots & u^2(-d-nb) & \dots \\ \dots & u(1-d-nb) & u^2(1-d) & \dots & u^2(1-d-nb) & \dots \\ \dots & u(2-d-nb) & u^2(2-d) & \dots & u^2(2-d-nb) & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & u(N-nb-d-1) & u^2(N-d-1) & \dots & u^2(N-nb-d-1) & \dots \end{bmatrix}$$

e o vetor de saída é dado por  $Y^T = [y(0) \ y(1) \ y(2) \ \dots \ y(N-1)]$ .

A estimativa do vetor de parâmetros,  $\hat{\theta}$ , pode ser obtida pelo procedimento dos mínimos quadrados. Utilizando a estimativa  $\hat{\theta}$ , a melhor predição da saída do sistema  $\hat{y}$ , é calculada por:

$$\hat{Y} = \Phi\hat{\theta} \quad (2.10)$$

e o erro de predição,  $E$ , é avaliado de acordo com:

$$E = Y - \hat{Y} = Y - \Phi\hat{\theta} \quad (2.11)$$

O estimador dos mínimos quadrados para o modelo Hammerstein é obtido minimizando o seguinte critério:

$$J = [Y - \Phi\hat{\theta}]^T W [Y - \Phi\hat{\theta}] \quad (2.12)$$

na qual a matriz  $W$  é diagonal, na forma:

$$W = \begin{bmatrix} w(1) & 0 & \dots & 0 \\ 0 & w(2) & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & w(N) \end{bmatrix} \quad (2.13)$$

e  $w(i)$  é a ponderação em cada componente do erro cujo valor é diretamente proporcional à precisão da medida. Minimizando a função custo (2.12) em relação a  $\hat{\theta}$  tem-se que:

$$\frac{\partial J}{\partial \hat{\theta}} = -2(Y^T W \Phi)^T + 2\Phi^T W \Phi \hat{\theta} = 0 \quad (2.14)$$

Assim, o estimador dos mínimos quadrados ponderado para o modelo Hammerstein é calculado por:

$$\hat{\theta} = [\Phi^T W \Phi]^{-1} \Phi^T W Y \quad (2.15)$$

e isto conduz ao mínimo desde que:

$$\frac{\partial^2 J}{\partial \hat{\theta}^2} = 2\Phi^T W \Phi > 0 \quad (2.16)$$

condição esta garantida se a matriz  $(\Phi^T W \Phi)$  é definida positiva, condição de excitação persistente. O estimador dos mínimos quadrados não ponderados para o modelo hammerstein é obtido admitindo-se que  $W = \sigma^2 I_N$ , isto é, a mesma ponderação aplicada em todos os erros de medida, ou seja, considerando a mesma confiança a todas as medidas. Logo, a equação (2.15) torna-se:

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T Y \quad (2.17)$$

sendo denominado estimador dos mínimos quadrados não recursivos.

Para a determinação da não-linearidade estática, deve-se primeiramente considerar  $\gamma_1 = 1$  obtendo-se assim diretamente os parâmetros da parcela linear do

modelo e, em princípio, também os parâmetros da parcela não-linear, pela relação:

$$\gamma_i = \frac{b_0 \gamma_i}{b_0} = \frac{b_1 \gamma_1}{b_1} = \dots = \frac{b_{nb} \gamma_i}{b_{nb}} \quad (2.18)$$

A presença de ruído de medição provoca, no entanto, incoerências nos resultados obrigando a adoção de outras medidas para solucionar o problema de redundância de parâmetros como, por exemplo, uma média aritmética.

$$\gamma_i = \frac{1}{nb + 1} \sum_{j=0}^{nb} \left( \frac{b_j \gamma_i}{b_j} \right) \quad (2.19)$$

Em [106], encontra-se uma dedução de outros métodos utilizados para estimar os parâmetros do modelo Hammerstein, tais como:

- Mínimos Quadrados com Restrições;
- Método do Erro de Predição;
- Método de Narendra-Galman;
- Método de Boutayeb;
- Método de Bai.

## 2.3 Redes Neurais

O conceito fundamental contido nos sistemas classificados atualmente com RNA's (Redes Neurais Artificiais) foi introduzido através do famoso trabalho de McCulloch e Pitts (1943) [82], que partindo do princípio de que o cérebro é constituído de unidades básicas, os neurônios, propuseram um modelo matemático para representar essas unidades: o neurônio artificial. No entanto, a classe denominada RNA's reúne atualmente diversos modelos matemáticos distintos e com domínios de aplicações variados. As redes neurais artificiais são ferramentas computacionais que têm sua origem na inteligência artificial. São formadas por elementos de processamento análogos aos neurônios biológicos, chamados neurônios artificiais.

As redes neurais possuem a habilidade de aproximar funções complexas. O conhecimento inerente da rede é dado por pesos nas conexões entre os neurônios

artificiais. A computação de uma rede neural é executada, numa primeira etapa, em paralelo por todos os neurônios de uma determinada camada e, posteriormente, em série quando as informações são propagadas de uma camada para outra. Esta coletividade e conectividade de operações resulta em um alto grau de processamento paralelo, que habilita a rede à resolver problemas complexos.

As RNs podem ser utilizadas na resolução de uma grande classe de problemas tais como: classificação, identificação, diagnóstico, análise de sinais e de imagens, otimização e controle [56] e [109].

### 2.3.1 Redes Perceptron Multicamadas (MLP)

A rede MLP consiste em um conjunto de unidades sensoriais (nós de fonte) que constituem a camada de entrada, uma ou mais camadas ocultas de nós computacionais intermediárias e uma camada de saída de nós computacionais ([56]). O sinal de entrada se propaga para frente através da rede, camada por camada. A Figura 2.2 mostra uma típica rede MLP formada pelas interconexões entre os neurônios. Esses neurônios são dispostos em camadas altamente interconectadas e elaboram somatórios ponderados de determinados sinais de entrada para gerar uma saída. As camadas da rede podem ser categorizadas como camada de entrada, onde informações disponíveis são apresentadas à rede; camadas intermediárias ou ocultas, no mínimo uma, porém normalmente de uma a três, onde os neurônios interagem entre si; e camada de saída, a qual contém a resposta a uma determinada entrada.

A operação de uma rede é dividida em três fases: a aprendizagem, a representação e a generalização. Na aprendizagem, padrões de entrada e saída são apresentados à rede neural para fins de ajustes em todas as interconexões entre os elementos de processamento, neurônios artificiais. A representação é a fase onde somente padrões de entrada, utilizados na aprendizagem, são apresentados e cabe a rede neural estimar a saída desejada. Na fase de generalização, a rede gera a resposta para padrões novos e similares às entradas.

A definição de uma topologia de uma RN do tipo MLP está intimamente ligada com a definição do número de camadas intermediárias e dos nós de camada intermediária. Este número é, geralmente, definido empiricamente e depende, fortemente, da distribuição dos padrões de treinamento e validação da rede. Uma função

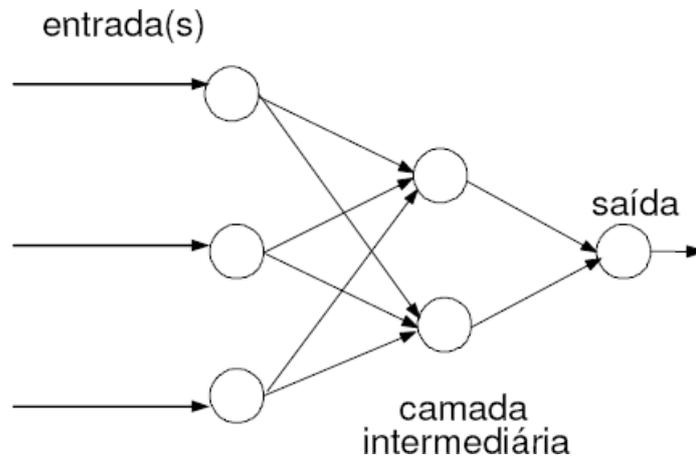


Figura 2.2: Rede MLP com Camada Intermediária.

contínua pode ser aproximada para qualquer grau de precisão usando uma rede neural com três camadas, desde que haja um número suficiente de neurônios ativos na camada oculta ([58]).

### Treinamento Backpropagation de RNs

A aprendizagem ou treinamento é o processo onde os pesos e bias da rede são modificados. O método utilizado nesse trabalho é o supervisionado, onde a rede aprende por experiência, ou seja, exemplos de entradas e saídas esperadas são apresentadas à rede, que ajusta os seus pesos. As redes multicamadas são treinadas pelo método *backpropagation* e os seus sinais são *feedforward* [104]. Tais redes apresentam conexões que são apropriadas para a modelagem dinâmica, onde se deseja mapear uma resposta de saída baseada em um sinal de entrada [8].

O algoritmo de aprendizagem *backpropagation* possui dois momentos distintos: em primeiro lugar, quando um padrão de entrada é apresentado a rede, o fluxo é alimentado para a frente, isto é, propagado adiante, até a camada de saída. Após, a saída obtida é comparada com a saída desejada e, se a saída desejada não corresponder à obtida dentro de uma determinada precisão desejada, é feita uma correção nos pesos das conexões sinápticas, ajustando-se os pesos na direção oposta do gradiente do erro instantâneo. O ajuste é proporcional ao gradiente, segundo um fator de proporcionalidade denominado de taxa de aprendizagem. O ajuste de peso é realizado da última camada em direção à camada de entrada. Isso pode ser visto

na Figura 2.3.

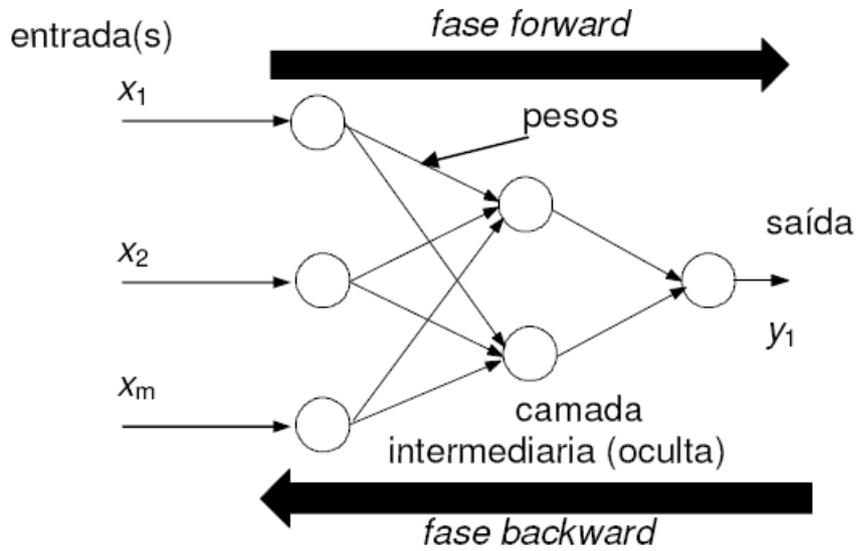


Figura 2.3: Fluxo de Processamento do Algoritmo de Retropropagação do Erro.

O algoritmo da retropropagação do erro (*backpropagation*), que faz uso destas duas fases, é apresentado a seguir:

- (i) inicia os pesos e parâmetros da rede neural;
- (ii) repete até o erro entre a saída produzida pela rede e a saída desejada ser mínimo, ou a realização de um dado número de ciclos:
- (iii) para cada padrão de treinamento  $X$ :
  - definir saída da rede através de fase *forward*;
  - comparar saídas produzidas com as saídas desejadas;
  - atualizar pesos dos nodos através da fase *backward*.

O *backpropagation* é baseado na regra proposta por [118], sendo também denominado regra delta generalizada. Neste caso, a função a ser minimizada é uma função erro, definida pela soma dos erros quadráticos e representada por:

$$E = \frac{1}{2} \sum_p \sum_{i=1}^k (d_i^p - y_i^p)^2 \quad (2.20)$$

onde:

$E$  é a medida de erro total;

$p$  é o número de padrões;

$k$  é o número de unidades de saída;

$d_i$  é a  $i$ -ésima saída desejada;

$y_i$  é a  $i$ -ésima saída gerada pela RN.

A equação (2.20) define o erro total cometido pela rede, ou a quantidade que, para todos os padrões  $p$  de um dado conjunto, as saídas geradas pela rede diferem das saídas desejadas. A regra delta generalizada requer que as funções de ativação utilizadas pelos nós sejam contínuas, diferenciáveis. A equação (2.21) ilustra o cálculo do valor de ativação.

$$y_j^p = f_j(\text{rede}_j^p) \quad (2.21)$$

onde:

$$\text{rede}_j^p = \sum_{i=1}^n x_i^p w_{ij} \quad (2.22)$$

A constante  $n$  representa o número de conexões de entrada do nó  $j$  e  $w_{ij}$  é o peso da conexão entre a entrada  $x_i^p$  e o nó  $j$ . As equações (2.20) a (2.22) mostram como obter o erro global da rede e como encontrar o erro local de um determinado padrão de entrada. Após processamento destas equações, deve-se aplicar a equação (2.23) que ajusta os pesos das diversas camadas:

$$\Delta w_{ji} = \eta \delta_j x_i \quad \text{ou} \quad (2.23)$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j(t) x_i(t) \quad (2.24)$$

onde:

$\delta_j = (d_j - y_j)$  para camada de saída e

$\delta_j = \sum w_{ij}\delta_i$  para camadas intermediárias, sendo  $i$  a unidade em questão e  $j$  a camada em questão.

## 2.4 Identificação em Redes Neurais Artificiais

No processo de identificação em questão, o objetivo é avaliar o comportamento de um processo não linear. Neste processo, os valores deverão ser normalizados antes de serem apresentados à rede, com o objetivo de evitar que valores de ordem de grandeza distinta influenciem nos resultados. A expressão adotada para a normalização resulta num padrão limitado entre 0,10 e 0,90, ao invés de 0,0 e 1,0. A escolha dessa normalização tem como objetivo possibilitar uma pequena extrapolação dos valores e evitar a utilização de valores extremos. A expressão de normalização dos padrões adotada foi:

$$V_{nor} = 0.1 + \left( \frac{V_{real} - V_{min}}{V_{max} - V_{min}} \right) * 0,8 \quad (2.25)$$

para desnormalização dos padrões, foi utilizada a expressão inversa:

$$V_{real} = V_{min} + \frac{(V_{nor} - 0,1)}{0,8} * (V_{max} - V_{min}) \quad (2.26)$$

sendo que:

- $V_{nor}$ : valor normalizado;
- $V_{min}$ : valor mínimo assumido pela variável;
- $V_{max}$ : valor máximo assumido pela variável;
- $V_{real}$ : valor da variável nas unidades do processo.

### 2.4.1 Seleção da Estrutura de Modelos Neurais

A estrutura de um modelo matemático é o esqueleto do modelo que, quando associado aos parâmetros, formam o próprio modelo [1]. Determinar a Estrutura do Modelo significa, em linhas gerais, escolher um modelo candidato dentro de uma

classe de modelos previamente selecionados. De uma maneira geral, o problema relacionado com a seleção da estrutura do modelo pode ser dividido em duas partes:

1. Selecionar uma família de modelos cuja estrutura seja considerada apropriada para descrever o sistema de interesse, por exemplo, modelos lineares (FIR (*Finite Impulsive Response*), ARX (*Autoregressive with Exogenous Inputs*), OE (*Output Error*), ARMAX (*Autoregressive Moving Average with Exogenous Inputs*), etc), modelos neurais (MLP (*Multilayer Perceptron*), RBF (Radial Basis Function), etc), modelos nebulosos (Linguístico, Relacional, Takagi Sugeno, etc), entre outros;
2. Escolher um subconjunto da família de modelos com a estrutura selecionada. Dentro da família dos modelos com estrutura linear, pode-se escolher, por exemplo, uma estrutura do tipo ARX(2,3,1), na qual (2,3,1) significa que a saída atual depende de duas saídas passadas e três entradas passadas, além de considerar um atraso de um período de amostragem.

## Regressores

Na identificação de sistemas lineares, a determinação da estrutura consiste basicamente da determinação da ordem do modelo. No caso de sistemas não lineares, o problema torna-se significativamente mais complexo e considerando que a identificação do processo será feita com redes neurais artificiais, a seleção da estrutura do modelo consiste em escolher não somente as entradas da rede mas também um conjunto adequado de regressores, além da própria arquitetura da rede neural. A estratégia utilizada com mais frequência é selecionar os regressores baseado nos conceitos advindos da teoria de identificação de sistemas lineares e então determinar a melhor arquitetura possível da RNA tomando-se os regressores como entradas da rede neural. A topologia de uma rede neural, por sua vez, é equivalente à estrutura do respectivo modelo matemático linear. Em [87] destacam-se as seguintes vantagens nesta abordagem:

- É uma extensão natural da (bem estabelecida) abordagem linear;
- A arquitetura interna da rede pode ser expandida para possibilitar mapeamentos não lineares mais complexos;

- Nível de complexidade não muito alto associado à determinação da estrutura do modelo;
- Adequado para projeto de controladores.

O equivalente não linear das estruturas de modelos lineares pode ser obtido genericamente por:

$$y(t) = g[\phi(t, \theta), \theta] + \epsilon(t) \quad (2.27)$$

que escrito na forma de preditor fica:

$$\hat{y}(t|\theta) = g[\phi(t, \theta), \theta] \quad (2.28)$$

onde  $\phi(t, \theta)$  é o vetor de regressão,  $\theta$  é o vetor de parâmetros ajustáveis da rede neural, também conhecidos como *pesos*, e  $g$  é a função que realiza o mapeamento não-linear, que aqui é representada por uma rede neural com estrutura *feedforward*.

Várias estruturas de modelos não-lineares podem ser construídas dependendo da escolha dos regressores. Escolhendo-se o vetor de regressão de acordo com a estrutura ARX, a estrutura do modelo não-linear correspondente é denominado NNARX, que é o acrônimo para "*Neural Networks ARX*". Da mesma forma, as estruturas NNFIR, NNARMAX e NNOE são geradas a partir das estruturas lineares correspondentes.

### **Estrutura Neural do Tipo NNARX**

Assim como seu correspondente linear, os preditores associados a esta estrutura neural sempre será estável pelo fato de haver apenas uma relação algébrica entre as previsões e as entradas e saídas passadas [87]. Esta característica é de particular importância em sistemas não-lineares pois as questões relacionadas com a estabilidade assumem uma importância ainda maior neste caso.

A rede neural com estrutura NNARX está representada graficamente na Figura 2.4. A ausência de problemas relacionados com a estabilidade dos modelos faz com que essa estrutura, seja utilizada com mais frequência quando o nível de ruído é baixo.

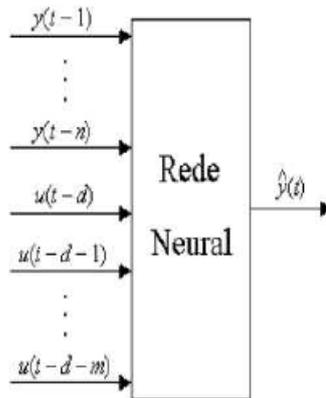


Figura 2.4: Rede Neural com estrutura NNARX.

### 2.4.2 Estimação de Modelos Neurais

O processo de selecionar um modelo a partir de uma estrutura previamente determinada é conhecido na comunidade de redes neurais artificiais como treinamento ou aprendizado [87].

Uma vez que um conjunto de modelos candidatos tenham sido escolhidos, o passo seguinte é escolher um modelo em particular deste conjunto. Normalmente tal escolha é feita baseada em algum tipo de critério de desempenho. Este critério pode ser formulado de várias maneiras, contudo, ele deve estar relacionado com a futura aplicação do modelo. A estratégia mais frequente é escolher o modelo que exhibe o melhor desempenho em predições um passo adiante em termos do menor erro quadrático médio entre as saídas observadas e as predições do modelo. Todavia, este critério pode não ser adequado para aplicações em estratégias de controle avançado, onde o desempenho desejado é não apenas a melhor predição um passo adiante, mas também predições o mais acuradas possível em um horizonte relativamente grande (predições  $k$  passos adiante).

A seguir será apresentada a abordagem utilizada para o treinamento de redes neurais artificiais, dando ênfase no seu emprego para a modelagem de sistemas dinâmicos.

## Modelagem Direta

Seja uma planta dinâmica discreta, causal, invariante no tempo cuja entrada,  $u(\cdot)$ , é uma função do tempo uniformemente limitada e cuja saída é representada por  $y_p(\cdot)$ . A planta é considerada estável, com parametrização conhecida mas com os valores dos parâmetros desconhecidos. O objetivo da identificação é construir um modelo neural adequado de modo que, quando submetido à mesma entrada  $u$  da planta, forneça uma saída  $y_m$  que se aproxime de  $y_p$  segundo um determinado critério. Este processo de identificação está representado graficamente na Figura 2.5.



Figura 2.5: Identificação de uma planta utilizando RNA.

O procedimento de treinamento de uma rede para representar a dinâmica de uma planta é conhecida como modelagem direta (*forward modelling*). O modelo neural é colocado em paralelo com a planta e os erros entre as saídas do sistema e as da rede neural, os erros de predição, são utilizados como sinal para o treinamento da rede. Este procedimento é detalhado na sequência.

Considere que a planta é descrita pela seguinte equação de diferenças não linear:

$$y_p(t+1) = g[y_p(t) \dots y_p(t-n+1); u(t) \dots u(t-m+1)] \quad (2.29)$$

A saída da planta  $y_p$  no tempo  $t+1$  depende dos  $n$  valores passados da própria saída e dos  $m$  valores passados da entrada  $u$ . Neste ponto, somente a parte dinâmica da resposta da planta será considerada; o modelo não representa explicitamente as possíveis perturbações agindo sobre a planta. Em [86] encontra-se uma descrição pormenorizada de métodos de identificação de sistemas utilizando redes neurais, além de analisarem casos especiais do modelo representado por (2.29).

A abordagem mais comum para a modelagem de sistemas é escolher a estrutura de entrada-saída da rede neural idêntica à do sistema que se deseja modelar. Denotando a saída da rede neural por  $y_m$ , obtém-se a seguinte representação:

$$y_m(t+1) = \hat{g}[y_p(t) \dots y_p(t-n+1); u(t) \dots u(t-m+1)] \quad (2.30)$$

Na equação acima,  $\hat{g}(\cdot)$  representa o mapeamento não linear entrada-saída realizado pela rede neural para aproximar o mapeamento  $g(\cdot)$  da planta. Deve-se notar que as entradas da rede neural incluem os valores passados da saída da planta e não os valores passados da saída da rede, ou seja, a rede não apresenta realimentação *feedback*. Neste caso, o algoritmo de aprendizado estático *backpropagation* pode ser utilizado para encontrar os valores ótimos dos pesos da rede neural. A estrutura do modelo representada pela equação (2.30) é chamada *série-paralelo* [86]. A estrutura de identificação resultante é ilustrada na Figura 2.6, onde a sigla LDA é a abreviação para Linha de Derivação de Atrasos.

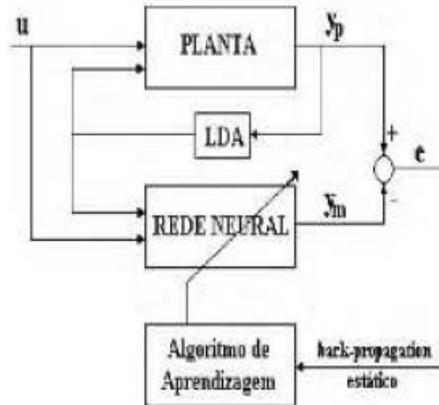


Figura 2.6: Estrutura de identificação série-paralelo.

Se for considerado que após um tempo adequado de treinamento a rede neural é capaz de fornecer uma boa representação da planta, ou seja,  $y_m \approx y_p$ , então a própria saída da rede neural e seus valores atrasados podem ser realimentados e, neste caso, a rede neural pode ser utilizada independentemente da planta. Tal rede neural é descrita por:

$$y_m(t+1) = \hat{g}[y_m(t) \dots y_m(t-n+1); u(t) \dots u(t-m+1)] \quad (2.31)$$

Esta estrutura (2.31) também pode ser utilizada desde o início, ou seja, durante todo o processo de aprendizado e, neste caso, é chamado de estrutura *paralela*[86].

A importância da classe das entradas a serem utilizadas no treinamento de sistemas de aprendizado já é bem conhecida. O conjunto de treinamento tem que ser representativo de toda a classe de entradas a que o sistema pode estar sujeito. Isto garantirá que o sistema responderá da maneira desejada, mesmo quando uma entrada não pertencente ao conjunto da treinamento for aplicada [83], [87] e [56].

## 2.5 Validação de Modelos

Após o modelo ter sido estimado, deve-se proceder a avaliação do mesmo para que se determine se o modelo selecionado atende ou não às necessidades requeridas. A metodologia empregada na validação do modelo identificado deve estar intimamente relacionada com a futura aplicação do modelo. Esta é geralmente a etapa mais imprecisa do procedimento de identificação, uma vez que os critérios de aceitabilidade do modelo nem sempre são muito claros ou fáceis de definir matematicamente.

Existem na literatura, uma série de testes padrões que podem ser realizados com relativa facilidade quando comparados com testes específicos necessários para avaliar o desempenho do modelo para fins de controle. Estes testes permitem obter uma validação preliminar, onde os piores modelos podem ser descartados rapidamente.

A maioria destes testes utilizam um conjunto de dados de validação que, além de serem diferentes do conjunto de dados utilizados durante a estimação, ou treinamento, no caso de redes neurais, deve conter dados representativos de toda a faixa operacional do sistema, além de ser obtido segundo as mesmas condições de operação do primeiro. Este procedimento, conhecido genericamente por validação cruzada [1], é uma medida direta da capacidade de generalização do modelo e, se aplicado simultaneamente ao processo de estimação, pode servir não só para avaliar continuamente o desempenho do modelo que está sendo identificado como também como um critério de parada para o algoritmo de estimação.

Outros testes podem ser encontrados na literatura tais como:

- Análise de Resíduos;
- Somatório do Erro Quadrático;
- Coeficiente de Correlação Múltipla;
- Princípio da Parcimônia;
- Erro de Predição.

Informação adicionais dos testes citados, podem ser vistos em [1], [72] e [20].

## 2.6 Algoritmo Genético

O Algoritmo Genético foi introduzido inicialmente por *Holland*, [60] e é um algoritmo de busca baseado no mecanismo da seleção natural de Darwin e na genética de Mendel.

Uma das principais aplicações do algoritmo genético é a utilização em problemas de otimização combinatória, nos quais existe um conjunto de elementos e deseja-se encontrar aquele que melhor se adapte às condições previamente especificadas ([89], [52] e [53]). O algoritmo genético realiza procedimentos de busca no espaço das soluções viáveis, utilizando regras probabilísticas para combinar soluções a fim de obter melhorias de qualidade.

Muitos problemas de engenharia podem ser modelados como problemas de otimização combinatória e sua resolução é obtida através de métodos algébricos, numéricos ou heurísticos, que fazem uma busca no espaço multidimensional das variáveis do problema. Estes métodos apresentam fraco desempenho quando a natureza do problema envolve não-linearidades, descontinuidades ou espaços de busca grandes. Nestas situações os algoritmos genéticos apresentam utilidade e robustez [52] [74].

O algoritmo genético difere dos procedimentos de busca tradicionais principalmente por não trabalhar com apenas um ponto, mas com um conjunto destes, e por utilizar a função de otimização isoladamente, sem a necessidade de derivadas ou outros cálculos auxiliares [52].

O algoritmo genético avalia um conjunto de soluções e, posteriormente, combina mecanismos de valorização das soluções mais adaptadas ao objetivo em questão, com estruturas de combinação e reprodução. Assim, um novo conjunto de soluções é criado utilizando-se informações das soluções avaliadas anteriormente.

Variações aleatórias são combinadas com seleção polarizada pelos valores de adequação atribuídos a cada solução, desta forma, o algoritmo genético não é considerado uma busca puramente aleatória, pois explora eficientemente a informação histórica para encontrar novas soluções capazes de melhorar a resposta ao critério de desempenho estipulado [15].

A literatura de engenharia de controle apresenta aplicações do algoritmo genético em uma extensa faixa de configurações. São encontradas aplicações em controle clássico e moderno, controle ótimo, controle adaptativo, controle robusto e identificação de sistemas [116], [117], [27] e [63].

### **Terminologia**

Como o algoritmo genético trabalha analogamente aos mecanismos biológicos de seleção natural e princípios da genética, algumas terminologias destas áreas foram adotadas e são freqüentemente encontradas na literatura [52], [84] e [85]:

1. Cromossomo ou Indivíduo => vetores de caracteres que representam as variáveis do problema. Cada indivíduo representa uma possível resposta para o problema considerado;
2. Gene => é a unidade básica do cromossomo, o caractere que descreve uma determinada variável do problema;
3. População => conjunto de indivíduos, inserido no espaço de busca do problema;
4. Geração => número da iteração que o algoritmo genético executa;
5. Operadores Genéticos => operações que o algoritmo genético realiza sobre os indivíduos com o objetivo de garantir a evolução da espécie e explorar novos espaços de busca;

6. Função *Fitness* => é a função que se quer otimizar. Representa as características do problema e é usada para avaliar o nível de aptidão dos indivíduos, ou seja, mede quanto o indivíduo está apto a resolver o problema;
7. Reprodução => consiste em simplesmente copiar o indivíduo de uma geração para a outra;
8. Recombinação ou *Crossover* => consiste na troca, evento aleatório, de informações entre dois indivíduos.
9. Mutação => consiste na troca, evento aleatório da característica de um gene, variável, em determinado indivíduo.

### 2.6.1 Configuração do Algoritmo Genético

A configuração básica de um algoritmo genético pode ser visto na Figura (2.7)

#### Geração da População Inicial

Um dos grandes problemas não só do algoritmo genético mas de toda a computação evolutiva é a escolha do tamanho da população inicial. Esta não pode ser exageradamente grande, desta forma terá um espaço de busca muito grande tornando o algoritmo extremamente lento. Uma população muito pequena corre-se o risco de não se ter a resposta no espaço de busca, ou se ter uma rápida convergência, prejudicando assim a busca por uma solução ótima.

A população inicial deve ser gerada de forma a conter indivíduos suficientemente variados, para que o algoritmo tenha facilidade em percorrer todos os locais do espaço de busca. Para se obter esta variabilidade, a geração da população inicial é normalmente realizada de forma aleatória.

Em algumas ocasiões, nas quais existem alguns conhecimentos sobre a solução do problema, é interessante a inclusão de indivíduos que representam estes conhecimentos heurísticos. Porém esta prática deve ser realizada com cuidado, pois, segundo [37] isto pode impedir o algoritmo de obter soluções melhores que não estejam presentes na população inicial.

## **Início**

1 -  $t = 0$ ;

2 - Criar a população inicial;

3 - Calcular o *fitness* de cada indivíduo dentro da população;

4 - Enquanto os critérios de terminação não forem satisfeitos faça:

{

$t = t+1$

Selecionar os indivíduos da geração atual para realizar as operações genéticas;

Aplicar as operações genéticas de crossover, reprodução e mutação nos indivíduos selecionados;

Inserir os novos indivíduos dentro de uma população na geração seguinte;

}

5 – Volte para o passo 3;

6 – Apresenta o resultado.

## **Fim**

Figura 2.7: Configuração de um Simples Algoritmo Genético.

## Avaliação da Função de Aptidão *Fitness*

A função de aptidão tem por objetivo avaliar cada indivíduo estabelecendo um valor numérico que representa o quão distante cada um deles está da solução ideal. A função de avaliação possui como entrada uma cadeia de *bits* ou um vetor, indivíduo, e como saída um valor real. Portanto, a função de avaliação codifica o conhecimento sobre o problema e permite quantificar cada solução comparativamente. Este é o único critério utilizado para o direcionamento da busca [37].

Na literatura de engenharia de controle, é comum encontrar aplicações que utilizam uma função de avaliação que aplica um dos seguintes índices de desempenho de sistemas de controle em malha fechada, ou composição destes [15]:

- IAE (Integral do valor absoluto do erro)

$$\int_0^T |e(t)| dt \quad (2.32)$$

- ISE (Integral do erro quadrático)

$$\int_0^T e^2(t) dt \quad (2.33)$$

para casos onde a dinâmica apresenta comportamento regulatório, e:

- ITAE (Integral do valor absoluto do erro multiplicado pelo tempo)

$$\int_0^T t |e(t)| dt \quad (2.34)$$

para casos onde a dinâmica apresenta comportamento servo.

com:  $e(t)$  é a referência "Set Point" menos a saída do sistema de controle.

Estes índices de desempenho são comumente utilizados na prática e na literatura para avaliar controladores [55], [6] e [33].

### 2.6.2 Método de Seleção

O processo de seleção ocorre após a aplicação da função de aptidão de cada indivíduo. O processo de seleção normalmente é baseado no princípio da sobrevivência dos melhores indivíduos, assim, os indivíduos com as melhores funções de

aptidão possuem uma maior probabilidade de serem mantidos e selecionados para a próxima geração. Da mesma forma, os indivíduos com baixas funções de aptidão possuem pouca probabilidade de permanecer e, conseqüentemente, podem ser eliminados da população. Os métodos de seleção mais utilizados na literatura de algoritmos genéticos são:

**Seleção por Roleta** O algoritmo genético clássico utiliza um esquema de seleção de indivíduos chamado seleção por roleta [52], [84] e [85]. Neste método cada indivíduo ocupa, em uma roleta, uma área proporcional ao seu nível de aptidão. Assim, aos indivíduos com maior aptidão é associada uma fatia maior da roleta e aos indivíduos com menor aptidão, fatias menores. Em seguida faz-se a simulação do giro da roleta  $n$  vezes, selecionando-se os  $n$  indivíduos que se manterão na etapa seguinte. Os mais aptos possuem uma probabilidade maior de serem selecionados, podendo, inclusive, serem selecionados mais de uma vez, enquanto os menos aptos, com uma probabilidade menor, podem desaparecer logo após as primeiras gerações.

O método de seleção por roleta é o método mais comumente utilizado, porém, apresenta algumas limitações. Este tipo de seleção leva o algoritmo a diminuir a diversidade da população, e isto ocasiona, em determinadas situações, a convergência prematura das soluções [37]. Um outro problema é que, como este tipo de seleção trabalha com probabilidades, é possível que um indivíduo com alto nível de aptidão não seja selecionado, e uma solução interessante se perca nas primeiras gerações.

Devido a estes problemas, outros métodos de seleção foram desenvolvidos, como seleção por torneio, seleção por ordenamento, seleção bi-classista, seleção por sorteio universal, técnica elitista [52] e [11].

**Seleção por Torneio:** Geram-se grupos aleatórios de indivíduos. O indivíduo com o maior nível de aptidão no grupo é selecionado, enquanto que os demais são descartados.

Este método de seleção apresenta a vantagem de não acarretar a convergência prematura do algoritmo e dificultar a estagnação do processo.

**Seleção por Ordenamento:** O método de seleção por ordenamento enumera os indivíduos de acordo com os seus respectivos níveis de aptidão. Para a determinação da probabilidade de seleção podem ser utilizados mapeamentos lineares ou

exponenciais.

Este método também não acarreta a convergência prematura do algoritmo, porém, apresenta como desvantagem o esforço computacional extra necessário para efetuar ordenamentos constantes de toda a população.

**Seleção Bi-classista:** A seleção bi-classista ordena os indivíduos da população e seleciona uma porcentagem pré-determinada de melhores indivíduos, e uma porcentagem de piores indivíduos, objetivando manter a diversidade em todas as gerações.

**Seleção por Sorteio Universal:** Os cromossomos são sorteados com igual probabilidade, independentemente da adequação.

**Técnica Elitista:** Esta técnica é utilizada em conjunto com algum método de seleção, normalmente a roleta, com o objetivo de aumentar a velocidade de convergência do algoritmo e garantir que determinado número de indivíduos com bom nível de aptidão sejam repassados para a geração seguinte.

A técnica elitista envia diretamente certo número de melhores indivíduos da população corrente para a próxima geração, evitando, assim, a possível perda de boas soluções no processo de seleção. Em muitas implementações costuma-se utilizar o elitismo de pelo menos o melhor indivíduo da população.

A desvantagem da utilização da técnica elitista é a possibilidade de forçar a busca, devido à reprodução maior deste indivíduo durante as evoluções, o que pode gerar a convergência prematura do algoritmo genético.

### 2.6.3 Operadores Genéticos

O processo de seleção isoladamente não cria uma nova geração, apenas seleciona os indivíduos mais aptos para fazerem parte do grupo que irá construir a próxima população.

A geração de novos indivíduos, e conseqüentemente, da nova população é feita através da utilização de operadores genéticos. Estes operadores tentam criar novas e melhores soluções, a partir das soluções existentes (selecionadas), encaminhando a busca a pontos cada vez mais próximos do critério de desempenho. A

função dos operadores genéticos é modificar os indivíduos, através de sucessivas gerações, com o objetivo de encontrar o melhor resultado no final do processo. Os operadores diversificam a população, resguardando e tentando aprimorar as características boas dos indivíduos já encontrados.

Os operadores genéticos são controlados por parâmetros que definem a probabilidade de suas aplicações. Estes parâmetros são configurados no momento do projeto e seus ajustes influenciam o desempenho do algoritmo.

Os operadores genéticos utilizados pelo algoritmo são reprodução, *crossover* e mutação, explanados a seguir.

- **Operador de *Crossover* ou Recombinação**

O operador de *crossover* cria novos indivíduos através da combinação de dois ou mais indivíduos selecionados aleatoriamente. O objetivo desta aplicação é a troca de informação entre diferentes soluções candidatas [52], [84] e [85].

Este operador pode ser utilizado de maneiras distintas, entre as quais, as mais comuns são: ***Crossover* ponto único**: um ponto de cruzamento é escolhido e a partir dele as informações genéticas dos indivíduos escolhidos aleatoriamente são trocadas. A figura de *crossover* ponto único pode ser vista em 2.8:

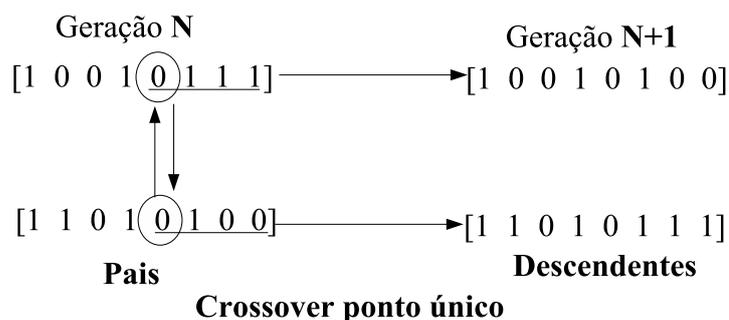


Figura 2.8: Operação de *Crossover* Realizada em um Ponto.

***Crossover* multi-pontos**: é uma generalização da idéia de troca de material genético, na qual muitos pontos de cruzamento podem ser utilizados. A operação de *crossover* multi-pontos pode ser visto na Figura 2.9.

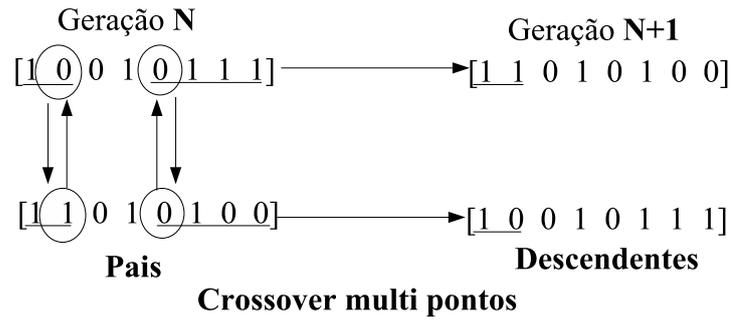


Figura 2.9: Operação de *Crossover* Multi Pontos.

Segundo [125] não há nenhuma comprovação de que alguma maneira de utilização do operador de crossover apresente um desempenho superior às demais. Apenas é possível verificar que cada maneira de se aplicar o operador é particularmente eficiente para uma determinada classe de problemas, e para outras, não.

- **Operador de Mutação**

O operador de mutação tem como objetivo manter e propiciar a diversidade genética da população [52] [84] e [85].

Este operador implica na alteração aleatória de uma ou mais características de um cromossomo escolhido, propiciando assim, a introdução de novos elementos na população. A utilização deste operador visa resgatar material genético perdido e inserir material não explorado, prevenindo desta maneira a convergência prematura do algoritmo para soluções sub-ótimas [23]. O operador de mutação pode ser visto na figura 2.10.



Figura 2.10: Operação de Mutação.

- **Operador de Reprodução**

O operador de reprodução tem como objetivo manter os melhores indivíduos na geração seguinte.

A operação de reprodução consiste em simplesmente copiar o indivíduo de uma geração para a outra, com isso, garantir que o resultado nunca irá piorar. Este tipo de operação so é utilizado quando se trabalha com a técnica elitista. A operação de reprodução pode ser vista na figura 2.11.



Figura 2.11: Operação de Reprodução.

## 2.6.4 Configuração dos Parâmetros

Existem quatro importantes parâmetros de controle que influenciam o desempenho do algoritmo genético: tamanho da população, número de gerações, probabilidade de crossover e probabilidade de mutação [52], [84] e [85]. A configuração adequada destes parâmetros é um dos passos importantes no momento do projeto, pois a eficiência do algoritmo dependente disto. Esta configuração é determinada por tentativa e erro. Apesar de diversas pesquisas na área, não existe uma regra determinística para o projetista estipular estes parâmetros de maneira a obter uma adequada relação quanto aos tópicos de diversidade na população e a capacidade de convergência do algoritmo [23].

Algumas técnicas adaptativas de ajuste de parâmetros vêm sendo sugeridas por pesquisadores, mas ainda nenhuma destas abordagens apresentou uma melhoria significativa no desempenho da busca [70]. Na literatura é possível encontrar algumas recomendações. Srinivas e Patnaik [112] comentam que aplicações típicas de algoritmos genéticos utilizam população entre 30 e 200 indivíduos, probabilidade de recombinação entre 0.5 e 1.0, e probabilidade de mutação entre 0.001 a 0.01.

### Tamanho da População

O tamanho da população é um parâmetro que deve ser definido considerando-se dois aspectos importantes [37]:

- Cobertura do espaço de busca da solução do problema;
- Tempo computacional.

O tamanho da população determina o número de cromossomos na população, afetando diretamente o desempenho global e a eficiência dos AG's. Com uma população pequena o desempenho pode cair, pois deste modo a população fornece uma pequena cobertura do espaço de busca do problema. Uma grande população geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. No entanto, para se trabalhar com grandes populações, são necessários maiores recursos computacionais, ou que o algoritmo trabalhe por um período de tempo muito maior.

### **Número de Gerações**

O número de geração é um parâmetro que irá controlar a evolução dos indivíduos, pois este determina o número de iterações do algoritmo.

### **Probabilidade de Crossover**

A probabilidade de crossover indica com qual taxa o operador de crossover atua sobre os indivíduos selecionados na população. O aumento deste valor possibilita a introdução mais rápida de novos indivíduos nas gerações.

Valores baixos de probabilidade de crossover podem ocasionar uma convergência muito lenta do algoritmo, e valores altos podem acarretar a retirada rápida de indivíduos com boas aptidões, gerando perdas de soluções.

### **Probabilidade de Mutação**

Determina a probabilidade em que uma mutação ocorrerá. Mutação é utilizada para dar nova informação para a população e também para prevenir que a população se sature com cromossomos semelhantes (Convergência Prematura). Uma baixa taxa de mutação previne que uma dada posição fique estagnada em um valor, além de possibilitar que se chegue em qualquer ponto do espaço de busca. Com uma taxa muito alta a busca se torna essencialmente aleatória além de aumentar muito a possibilidade de que uma boa solução seja destruída. A melhor Taxa de Mutação é dependente da aplicação mas, para a maioria dos casos é entre 0,001 e 0,1.

## **2.6.5 Convergência**

O estudo de convergência dos AG pode ser distinguidos em duas grandes áreas de pesquisas. A primeira linha de pesquisa resulta de investigações teóricas por

[60] e [52] e são baseados no teorema dos *schemata* e análise da função de Walsh. Porém, uma busca global de crossover e mutação não é considerada possível através do teorema dos *schemata*. Além do mais, a análise da função Walsh tem sido usado para evoluir o fenômeno da decepção, que é o caso onde não se encontram resultados ótimos, mas não pode analisar o comportamento geral dos AG.

Uma segunda linha de pesquisa trata AG como uma Cadeia de Markov. Em [103] é provado a convergência global do AG com a estratégia Elitista e em [108] é investigado um determinante no qual a taxa de convergência depende dele. Mostrou-se que a principal dependência está ligada ao segundo maior valor dos autovalores da matriz de transição da Cadeia de Markov. Porém, os autores não podem explicar como esta matriz é afetada pelos parâmetros do AG, como taxa de mutação e *crossover*, para melhorar a convergência do AG. Existe uma grande discussão a respeito do ajuste dos parâmetros do AG na literatura. Em [112] utiliza-se a heurística, propondo que o tamanho da população, a taxa de crossover e a taxa de mutação devem estar no intervalo de [30, 200], [0.5, 1.0] e [0.001, 0.05], respectivamente.

### **Crítérios de Parada**

A finalização ideal do algoritmo genético seria a interrupção do procedimento somente no momento em que a solução global fosse encontrada. Porém, a maneira de se identificar este ponto global é desconhecida na maioria dos casos [37]. Devido a isto, é necessário estipular um critério de parada que pode ser feita de diversas formas, como por exemplo, a observação da convergência da população, que ocorre quando praticamente todos os indivíduos são cópias idênticas da mesma seqüência de genes [52] [84] e [85]. A determinação de um número máximo de gerações ou limitação do tempo de processamento, também podem ser utilizados.

## **2.7 Conclusão do Capítulo**

Modelos não-lineares possibilitam um retrato mais fiel do processo quando este se faz necessário. Apesar de apresentar uma complexidade maior, apenas a representação a partir de um modelo não-linear permite a análise de algumas características do sistema como oscilações e bifurcações ([90]).

Neste capítulo foram mostrados dois tipos de modelos não-lineares que são utilizados neste trabalho, sendo o Hammerstein e as Redes Neurais. Foram descritas também as estruturas destes modelos, bem como as técnicas de identificação. No Modelo Hammerstein a estrutura é caracterizada pela ordem do polinômio não-linear e pela ordem do modelo linear CARIMA. Já na rede neural, a estrutura é caracterizada pelo número de regressores e pela quantidade de neurônios na camada oculta. A identificação dos parâmetros do modelo neural é obtida na fase de treinamento da rede, enquanto que a identificação do modelo Hammerstein é feita através do método dos Mínimos Quadrados. No final, é comentado sobre o método de validação do modelo Neural, que se assemelha ao método de validação do modelo Hammerstein.

No final do capítulo foram abordados os conceitos básicos envolvidos nos Algoritmos Genéticos, cujo princípio é baseado nos mecanismos de seleção natural. Portanto, utiliza as mesmas terminologias usadas na Genética como Gene, Cromossomo, População, Geração, Mutação, etc, a fim de elaborar o Algoritmo. Foram discutidas questões importantes como geração da população inicial, métodos de seleção dos indivíduos, configuração dos parâmetros e convergência.

## Capítulo 3

# Controle Preditivo Baseado em Modelos Lineares

A utilização de modelos lineares numa aplicação de controle preditivo é bastante comum pois, além da popularidade deste tipo de modelo, muitas vezes, torna-se necessário o emprego de um modelo simplificado para possibilitar que todos os cálculos envolvidos sejam realizados dentro do intervalo correspondente a um período de amostragem viabilizando, assim, o controle em tempo real. Um modelo linear possibilita, também, uma solução analítica para o problema de minimização da função custo quando não são consideradas restrições. A opção por um modelo linear para a representação da planta deve ser a escolha preferencial sempre que este possibilite que o controlador alcance o desempenho almejado pelo usuário [14].

Nos últimos anos houve um grande crescimento nas aplicações industriais de controle preditivo baseado em modelos lineares. A Tabela 3.1 apresenta algumas destas aplicações presentes no trabalho de Qin e Badgwell [93]. Este estudo vem a confirmar a indústria química e a do petróleo como principais áreas de aplicação das estratégias de controle preditivo baseado em modelo linear (MPC).

### 3.1 Controle por Matriz Dinâmica (DMC)

O controlador DMC (*Dynamic Matrix Control*) desenvolvido por C. R. Cutler e B. L. Ramaker foi um dos primeiros controladores preditivos baseados em modelo a apresentar disponibilidade comercial.

Tabela 3.1: Aplicações Comerciais de MPC.

| Area         | Adersa      | Aspen<br>Technology | Honneywel<br>Hi-Spec | Invensys   | SGS        | Total       |
|--------------|-------------|---------------------|----------------------|------------|------------|-------------|
| refinaria    | 1200        | 480                 | 280                  | 25         | -          | 1985        |
| petroquimica | 450         | 80                  | -                    | 20         | -          | 550         |
| química      | 100         | 20                  | 03                   | 21         | -          | 144         |
| papel        | 18          | 50                  | -                    | -          | -          | 68          |
| ar e gás     | -           | 10                  | -                    | -          | -          | 10          |
| utilidades   | -           | 10                  | -                    | 04         | -          | 14          |
| metalurgia   | 08          | 06                  | 07                   | 16         | -          | 37          |
| alimentos    | -           | -                   | 41                   | 10         | -          | 51          |
| polímeros    | 17          | -                   | -                    | -          | -          | 17          |
| fornos       | -           | -                   | 42                   | 03         | -          | 45          |
| aeroespacial | -           | -                   | 13                   | -          | -          | 13          |
| automotiva   | -           | -                   | 07                   | -          | -          | 07          |
| outras       | 40          | 40                  | 1045                 | 26         | 450        | 1601        |
| <b>Total</b> | <b>1833</b> | <b>696</b>          | <b>1438</b>          | <b>125</b> | <b>450</b> | <b>4542</b> |

### 3.1.1 DMC Monovariável

O DMC para um sistema SISO é baseado no modelo não-paramétrico da resposta ao degrau dado por:

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t - i) \quad (3.1)$$

Os valores previstos ao longo de um horizonte são calculados por:

$$\begin{aligned} \hat{y}(t + k|t) &= \sum_{i=1}^{\infty} g_i \Delta u(t + k + i) + \hat{n}(t + k|t) = \\ &= \sum_{i=1}^k g_i \Delta u(t + k - i) + \sum_{i=k+1}^{\infty} g_i \Delta u(t + k - i) + \hat{n}(t + k|t) \end{aligned} \quad (3.2)$$

onde  $\hat{n}$  é o distúrbio. Considerando os distúrbios constantes, isto é  $\hat{n}(t + k|t) = \hat{n}(t|t) = y_m(t) - \hat{y}(t|t)$ , (3.2) pode ser reescrita como:

$$\begin{aligned}
\hat{y}(t+k|t) &= \sum_{i=1}^k g_i \Delta u(t+k-i) + \sum_{i=k+1}^{\infty} g_i \Delta y(t+k-i) + y_m(t) - \quad (3.3) \\
&- \sum_{i=1}^{\infty} g_i \Delta u(t-i) = \sum_{i=1}^k g_i \Delta u(t+k-i) + f(t+k)
\end{aligned}$$

onde  $f(t+k)$  é a resposta livre do sistema, isto é, parte da resposta que depende de valores passados do sistema e é dada por:

$$f(t+k) = y_m(t) + \sum_{i=1}^{\infty} (g_{k+i} - g_i) \Delta u(t-i) \quad (3.4)$$

Se o processo é estável, o coeficiente  $g_i$  da resposta ao degrau, tende a um valor constante depois de  $N$  períodos de amostragem, assim, pode-se considerar que:

$$g_{k+i} - g_i \approx 0, \quad i > N \quad (3.5)$$

Desta forma, (3.4) pode ser reescrita como:

$$f(t+k) = y_m(t) + \sum_{i=1}^N (g_{k+i} - g_i) \Delta u(t-i) \quad (3.6)$$

Agora, as predições podem ser calculadas ao longo de um horizonte de predição ( $k = 1, \dots, N_2$ ), considerando  $N_u$  ações de controle:

$$\begin{aligned}
\hat{y}(t+1) &= g_1 \Delta u(t) + f(t+1) \\
\hat{y}(t+2) &= g_2 \Delta u(t) + g_1 \Delta u(t+1) + f(t+2) \\
&\vdots \\
\hat{y}(t+N_2) &= \sum_{i=N_2-N_u+1}^{N_2} g_i \Delta u(t+N_2-i) + f(t+N_2)
\end{aligned} \quad (3.7)$$

Definindo uma Matriz Dinâmica  $G$   $N_2 \times N_u$  como:

$$G = \begin{bmatrix} g_1 & 0 & \dots & 0 \\ g_2 & g_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_u} & g_{N_u-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2} & g_{N_2-1} & \dots & g_{N_2-N_u-1} \end{bmatrix} \quad (3.8)$$

então a Equação de Predição (3.7) pode ser escrita como:

$$\hat{y} = G\Delta u + f \quad (3.9)$$

onde  $\hat{y} = [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N_2)]^T$ ;  $\Delta u = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u-1)]^T$ ;  $f = [f(t+1), f(t+2), \dots, f(t+N_2)]^T$ .

### 3.1.2 Solução do DMC Irrestrito

Para solucionar o problema do DMC, uma sequência de sinais de controle  $\Delta u(t)$ ,  $\Delta u(t+1), \dots, \Delta u(t+N_u-1)$  será obtida para minimizar a expressão (1.1), que pode ser escrita na seguinte forma matricial: .

$$J = \delta(G\Delta u + f - w)^T(G\Delta u + f - w) + \lambda\Delta u^T\Delta u \quad (3.10)$$

onde  $w$  é o vetor de referência futura dado por:

$$w = [w(t+d+1) \ w(t+d+2) \ \dots \ w(t+d+N)]^T \quad (3.11)$$

A expressão (3.10) pode ser escrita como:

$$J = \frac{1}{2}\Delta u^T\mathbf{G}\Delta u + \mathbf{b}^T\Delta u + \mathbf{f} \quad (3.12)$$

onde:

$$\mathbf{G} = 2(\delta G^T G + \lambda \mathbf{I})$$

$$\mathbf{b}^T = 2\delta(f - w)^T G$$

$$\mathbf{f} = \delta(f - w)^T(f - w)$$

O mínimo de  $J$ , assumindo que não existe restrições no sinal de controle, é linear e é obtido tomando a derivada de  $J$  em relação a  $\Delta u$  e igualando a zero, obtendo-se:

$$\Delta u = -\mathbf{G}^{-1}\mathbf{b} = (\delta G^T G + \lambda \mathbf{I})^{-1} G^T \delta(w - f) \quad (3.13)$$

O sinal de controle que é enviado para o processo é apenas o primeiro elemento do vetor  $\Delta u$ , que é dado por:

$$\Delta u(t) = \mathbf{K}(w - f) \quad (3.14)$$

onde  $\mathbf{K}$  é dado por:

$$K = (\delta G^T G + \lambda \mathbf{I})^{-1} G^T \delta \quad (3.15)$$

### 3.1.3 Caso Restrito Monovariável

Na prática, porém, todo tipo de processo está sujeito a restrições, que podem ser: limitações nos atuadores, limitações nos sensores, limitações nos sinais de controle, limitações dos sinais de saída, ou seja:

1. Variação do controle:  $\Delta u(t) = u(t) - u(t - 1)$ ,  $\Delta u_{min} \leq \Delta u(t + j) \leq \Delta u_{max}$ ,  $j = 0, 1, \dots, N_u - 1$
2. Sinal de controle:  $u(t)$ ,  $u_{min} \leq u(t + j) \leq u_{max}$ ,  $j = 0, 1, \dots, N_u - 1$
3. Sinal da saída predita:  $\hat{y}(t)$ ,  $y_{min} \leq \hat{y}(t + j) \leq y_{max}$ ,  $j = d + 1, \dots, d + N$

Quando restrições são levadas em consideração, não existe uma solução explícita como foi mostrado anteriormente, desta forma, o problema de programação quadrática definido abaixo deve ser solucionado:

$$\min_{\Delta u} J(\Delta u) \quad \text{sa } \{ \mathbf{S} \Delta u \leq \mathbf{c} \} \quad (3.16)$$

onde:

$$\mathbf{S} = \begin{bmatrix} \Gamma_{N_u} \\ -\Gamma_{N_u} \\ T_{N_u} \\ -T_{N_u} \\ G \\ -G \end{bmatrix}_{(4N_u+2N) \times N_u}, \quad \mathbf{c} = \begin{bmatrix} \Gamma_{N_u} \Delta u_{max} \\ -\Gamma_{N_u} \Delta u_{min} \\ \Gamma_{N_u}(u_{max} - u(t-1)) \\ \Gamma_{N_u}(u(t-1) - u_{min}) \\ \Gamma_N y_{max} - \mathbf{f} \\ \mathbf{f} - \Gamma_N y_{min} \end{bmatrix}_{(4N_u+2N) \times 1} \quad (3.17)$$

$$\Gamma_{N_u} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{N_u \times 1} \quad (3.18)$$

onde:

$$\mathbf{T}_{N_u} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}_{N_u \times N_u} \quad (3.19)$$

Detalhes sobre a formação desta matriz de restrições podem ser vistos em [28]. O problema de programação quadrática é solucionado neste trabalho usando a função *quadprog* do MATLAB. O resultado é um vetor  $\Delta \mathbf{u}$  no qual apenas o primeiro elemento  $\Delta u(t)$  é enviado para o processo para fazer com que a saída se aproxime da referência desejada, ao mesmo tempo minimizando a função(3.12), respeitando as limitações do processo impostas através das restrições.

### 3.1.4 DMC Multivariável

Considera-se um sistema com dinâmica MIMO com  $m$  entradas ( $u_l$ ,  $l = 1, \dots, m$ ) e  $n$  saídas ( $y_j$ ,  $j = 1, \dots, n$ ) que pode ser descrito pela equação 3.20:

$$y_j(t) = \sum_{\ell=1}^m \sum_{q=1}^{\infty} g_{j\ell}(q) \Delta u_{\ell}(t - q) \quad (3.20)$$

onde:  $g_{jl}$  é a saída da resposta ao degrau  $j$  em relação a entrada  $l$ , e  $\Delta u_l(t) = u_l(t) - u_l(t - 1)$ , é a variação do sinal de controle. Denota-se  $\hat{y}_j(t + 1)$  a sequência da saída  $j$  prevista do sistema,  $i$  passos a frente do instante atual  $t$ , e  $w_j(t + i)$  é a predição do *set point* da saída  $j$  de  $i$  passos a frente.

A ideia básica do DMC multivariável é calcular um sinal de controle futuro ao longo de um horizonte de controle  $N_u$ , i.e.,  $u_l(t + i)$ , para  $l = 1, 2, \dots, m$  e  $i = 0, \dots, N_2 - 1$ , de tal maneira que minimize a função custo:

$$\sum_{j=1}^n \sum_{i=1}^{N_2} \delta_j (\hat{y}_j(t+i) - r_j(t+i))^2 + \sum_{\ell=1}^m \sum_{i=0}^{N_u-1} \lambda_j \Delta u_\ell^2(t+i) \quad (3.21)$$

onde:  $\delta_j$ ,  $\lambda_j$  são fatores de ponderação e  $N_2$  é o horizonte de predição. Considerando  $\Delta u_\ell(k+i) = 0$ ,  $i > N_u$ , e assumindo que existe um número inteiro  $N_s$  tal que  $\Delta u_\ell(t) = 0$  para  $t > N_s$ , e  $l = 1, \dots, m$ , a saída prevista a partir do instante atual  $t$  é dada por:

$$\hat{y}_j(t+i) = \sum_{\ell=1}^m \sum_{q=\min\{1, i-N_u+1\}}^{N_s} g_{j\ell}(q) \Delta u_\ell(t+i-q) \quad (3.22)$$

Separando as ações futuras das ações de controle passadas, e definindo:

$$\begin{aligned} G_j(i) &:= [g_{j1}(i) \quad g_{j2}(i) \quad \dots \quad g_{jm}(i)] \\ f_j(t+i) &:= \sum_{\ell=1}^m \sum_{q=i+1}^{N_s} g_{j\ell}(q) \Delta u_\ell(t+i-q) \\ \Delta \bar{u}(t+i) &:= [\Delta u_1(t+i) \quad \Delta u_2(t+i) \\ &\quad \dots \quad \Delta u_m(t+i)]^T \end{aligned} \quad (3.23)$$

pode-se representar a equação (3.22) por:

$$\begin{aligned} \hat{y}_j(t+i) &= \sum_{q=\min\{1, i-h_c+1\}}^{N_s} G_j(q) \Delta u(t+i-q) \\ &\quad + f_j(t+i) \end{aligned} \quad (3.24)$$

onde o primeiro termo de (3.24) é a resposta forçada, que depende das ações de controle presente e futuras e o segundo termo é a resposta livre, que depende das ações de controle passadas.

Sejam os seguintes vetores:

$$\begin{aligned}
\Delta\bar{U} &:= [\Delta\bar{u}(t) \ \Delta\bar{u}(t+1) \ \dots \ \Delta\bar{u}(t+N_u-1)]^T \\
\hat{\bar{y}}(t) &:= [\hat{y}_1(t) \ \hat{y}_2(t) \ \dots \ \hat{y}_n(t)]^T \\
\hat{\bar{Y}} &:= [\hat{\bar{y}}(t+1) \ \hat{\bar{y}}(t+2) \ \dots \ \hat{\bar{y}}(t+N_2)]^T \\
\bar{f}(t) &:= [f_1(t) \ f_2(t) \ \dots \ f_n(t)]^T \\
\bar{w}(t) &:= [w_1(t) \ w_2(t) \ \dots \ w_n(t)]^T \\
\bar{F} &:= [f(t+1) \ f(t+2) \ \dots \ f(t+N_2)]^T \\
\bar{W} &:= [w(t+1) \ w(t+2) \ \dots \ w(t+N_2)]^T \\
G(i) &:= [G_1(i) \ G_2(i) \ \dots \ G_n(i)]^T
\end{aligned} \tag{3.25}$$

Usando as definições (3.25), pode-se representar as  $n$  saídas previstas ao longo de um horizonte de predições  $N_2$  por:

$$\hat{\bar{Y}} = \bar{G}\Delta\bar{U} + \bar{F} \tag{3.26}$$

onde  $\bar{G}$  tem dimensão  $nN_2 \times mN_u$  e é dada por:

$$\bar{G} = \begin{bmatrix} G(1) & 0 & \dots & 0 \\ G(2) & G(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G(N_u) & G(N_u-1) & \dots & G(1) \\ \vdots & \vdots & \ddots & \vdots \\ G(N_2) & G(N_2-1) & \dots & G(N_2-N_u-1) \end{bmatrix}$$

Define-se  $\bar{\lambda}$  como uma matriz diagonal contendo parâmetros de ponderações  $\lambda_j$ ,  $j = 1, \dots, m$ , e  $\bar{\Lambda}$  uma matriz  $mN_u \times mN_u$  dada por:

$$\bar{\Lambda} = \begin{bmatrix} \bar{\lambda} & 0 & \dots & 0 \\ 0 & \bar{\lambda} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{\lambda} \end{bmatrix}$$

De (3.26), pode-se representar a função custo (3.21) por:

$$\frac{1}{2}\Delta\bar{U}^T \bar{\mathcal{H}}\Delta\bar{U} + \bar{\mathcal{B}}^T \Delta\bar{U} + \bar{\mathcal{F}} \tag{3.27}$$

onde  $\bar{\mathcal{H}} = 2(\bar{G}^T \bar{G} + \bar{\Lambda})$ ,  $\bar{\mathcal{B}}^T = 2(\bar{F} - \bar{W})^T \bar{H}$  e  $\bar{\mathcal{F}} = (\bar{F} - \bar{W})^T (\bar{F} - \bar{W})$ .

A solução do problema do Controle Preditivo DMC é dividida em duas partes: na primeira, calcula-se o vetor de controle  $\Delta \bar{U}$  pela minimização da função custo (3.27). Em seguida, aplica-se o sinal  $\Delta \bar{u}(t)$  na entrada do processo e descarta-se os sinais futuros  $\Delta \bar{u}(t + i)$ ,  $i = 1, \dots, N_u - 1$ .

## 3.2 Controle Preditivo Generalizado (GPC)

O método do GPC foi proposto por *Clarke et al* [19] e tem sido largamente utilizado em aplicações industriais como pode ser visto em [18], mostrando bom desempenho e robustez. Esta metodologia de controle abrange uma ampla faixa de processos, desde as plantas com dinâmicas mais simples até aquelas mais complexas com atraso de transporte, fase não-mínima, instabilidade em malha aberta além de restrições nos sinais de controle e na saída do processo. Basicamente o que diferencia o GPC do DMC é o método de obtenção da matriz de previsão, pois a forma de calcular a solução minimizando a função custo é a mesma. Portanto a ênfase do texto que segue é mostrar como determinar a previsão da saída do sistema.

### 3.2.1 GPC Monovariável

O GPC utiliza um modelo interno para gerar as previsões futuras do comportamento do processo, representado pelo modelo CARIMA :

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t-1) + C(q^{-1})\frac{e(t)}{\Delta} \quad (3.28)$$

onde:  $u(t)$  e  $y(t)$  são as sequências de controle e saída da planta respectivamente,  $e(t)$  é um ruído branco de média zero,  $d$  é o atraso de transporte do sistema,  $\Delta = 1 - q^{-1}$  e  $A$ ,  $B$ , e  $C$  são polinômios em  $q^{-1}$  da seguinte forma:

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_naq^{-na}$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q^{-2} + \dots + b_nbq^{-nb}$$

$$C(q^{-1}) = 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_nccq^{-nc}$$

Por motivo de simplificação, o polinômio  $C(q^{-1})$  é considerado igual a 1.

O algoritmo de GPC consiste em obter uma sequência de controle  $\Delta u(t)$ ,  $\Delta u(t+1), \dots$  de tal maneira que a saída prevista do processo  $\hat{y}(t+j)$  acompanhe a referência  $w(t+j)$  com um menor erro possível, além da minimização da função objetivo (1.1).

### Predição Usando o Modelo Monovariável

Será determinado a seguir, o preditor  $j$  passos a frente  $\hat{y}(t+j)$  definido por:

$$\hat{y}(t+j) = \min_q E\{(y(t+j) - q)^2\}^1 \quad (3.29)$$

De acordo com [110]:

$$\hat{y}(t+j) = E\{y(t+j)|y_0, y_1, \dots, y(t)\} \quad (3.30)$$

Observe que (3.28) é equivalente a:

$$\tilde{A}(q^{-1})y(t) = q^{-d}B(q^{-1})\Delta u(t-1) + e(t) \quad (3.31)$$

onde  $\tilde{A} = (1 - q^{-1})A$ ,  $\Delta u = (1 - q^{-1})u$ . Considerando a Equação Diofantina,

$$1 = E_j(q^{-1})\tilde{A}(q^{-1}) + q^{-j}F_j(q^{-1}) \quad (3.32)$$

onde os polinômios  $E_j$  e  $F_j$  são unicamente definidos com graus de  $j-1$  e  $na$  respectivamente. Multiplicando (3.31) por  $E_j q^j$  e usando o fato de que  $\tilde{A}(q^{-1})E_j(q^{-1}) = 1 - q^{-j}F_j(q^{-1})$ , conclui-se que:

$$\hat{y}(t+j) = F_j(q^{-1})y(t) + E_j(q^{-1})B(q^{-1})\Delta u(t+j-d-1) + E_j(q^{-1})e(t+j) \quad (3.33)$$

Como o polinômio  $E_j(q^{-1})$  tem grau  $j-1$ , os termos que multiplicam o ruído  $e(t+j)$  estão no futuro. Então, aplicando  $E[y(t+j)|y_0, y_1, \dots, y(t)]$  em (3.33), obtém-se:

$$\hat{y}(t+j) = F_j(q^{-1})y(t) + E_j(q^{-1})B(q^{-1})\Delta u(t+j-d-1) \quad (3.34)$$

---

<sup>1</sup>Preditor  $j$  passos a frente que minimiza a variância do erro de previsão

Definindo:

$$G_j(q^{-1}) = E_j(q^{-1})B(q^{-1}) = g_0 + g_1q^{-1} + g_2q^{-2} + \dots + g_{nb+j-1}q^{-(nb+j-1)}$$

chega-se a seguinte expressão:

$$\hat{y}(t+j) = F_j(q^{-1})y(t) + G_j(q^{-1})\Delta u(t+j-d-1) \quad (3.35)$$

Observe que  $G_j(q^{-1})\Delta u(t+j-d-1)$ ,  $j \geq d+1$ , possui termos com ações de controle passadas, presentes e futuras. Para separá-los, define-se um polinômio  $H_j$  de ordem  $j-d-1$  dado por:

$$H_j(q^{-1}) = g_0 + g_1q^{-1} + g_2q^{-2} + \dots + g_{j-d-1}q^{-(j-d-1)} \quad (3.36)$$

e também um outro polinômio  $\bar{H}_j(q^{-1})$  por:

$$\bar{H}_j(q^{-1}) = G_j(q^{-1}) - H_j(q^{-1}) = g_{j-d}q^{-(j-d)} + \dots + g_{nb+j-1}q^{-(nb+j-1)} \quad (3.37)$$

Portanto, para  $j \geq d+1$ ,  $H_j(q^{-1})\Delta u(t+j-d-1)$  é um polinômio com ações de controle presente e futuro e  $\bar{H}_j(q^{-1})\Delta u(t+j-d-1)$  é um polinômio com ações de controle passadas.

Usando o **Lema 1** [28] para  $j = d+l$ , a equação de predição (3.35) será:

$$\hat{y}(t+d+l) = H_{d+l}\Delta u(t+l-1) + q^l\bar{H}_{d+l}(q^{-1})\Delta u(t-1) + F_{d+l}(q^{-1})y(t) \quad (3.38)$$

Considerando um horizonte de previsão  $j = d+l$ , e usando (3.38) para  $1 \leq l \leq N$ , obtém-se o seguinte sistema de equações:

$$\begin{aligned} \hat{y}(t+d+1) &= H_{d+1}\Delta u(t) + q\bar{H}_{d+1}\Delta u(t-1) + F_{d+1}y(t) \\ \hat{y}(t+d+2) &= H_{d+2}\Delta u(t+1) + q^2\bar{H}_{d+2}\Delta u(t-1) + F_{d+2}y(t) \\ &\vdots \\ \hat{y}(t+d+N) &= H_{d+N}\Delta u(t+N-1) + q^N\bar{H}_{d+N}\Delta u(t-1) + F_{d+N}y(t) \end{aligned} \quad (3.39)$$

Considerando (3.39) e usando o Lema 2 [28], chega-se a seguinte expressão matricial:

$$\hat{y} = G\Delta u + \bar{\mathbf{H}}(z^{-1})\Delta u(t-1) + \mathbf{F}(z^{-1})y(t) \quad (3.40)$$

onde:

$$\hat{y} = \begin{bmatrix} \hat{y}(t+d+1) \\ \hat{y}(t+d+2) \\ \vdots \\ \hat{y}(t+d+N) \end{bmatrix}_{N \times 1} \quad (3.41)$$

$$\Delta u = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N_u-1) \end{bmatrix}_{N_u \times 1} \quad (3.42)$$

$$G = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_u-1} & g_{N_u-2} & \dots & g_0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \dots & g_{N-N_u} \end{bmatrix}_{N \times N_u} \quad (3.43)$$

$$\bar{\mathbf{H}} = \begin{bmatrix} q\bar{H}_{d+1} \\ q^2\bar{H}_{d+2} \\ \vdots \\ q^N\bar{H}_{d+N} \end{bmatrix}_{N \times 1} \quad (3.44)$$

$$\mathbf{F}(q^{-1}) = \begin{bmatrix} F_{d+1}(q^{-1}) \\ F_{d+2}(q^{-1}) \\ \vdots \\ F_{d+N}(q^{-1}) \end{bmatrix}_{N \times 1} \quad (3.45)$$

Definindo:

$$f = \bar{\mathbf{H}}(q^{-1})\Delta u(t-1) + \mathbf{F}(q^{-1})y(t) \quad (3.46)$$

como resposta livre, que depende das entradas e saídas passadas, chega-se à seguinte expressão:

$$\hat{y} = G\Delta u + f$$

### Predição do Modelo Monovariável com Distúrbio

Neste caso, o GPC também utilizará um modelo interno para gerar as previsões futuras da saída sujeita a distúrbio mensuráveis. Porém, dentro deste novo algoritmo, o modelo CARIMA deve ser modificado para que se possa incluir a ação devido ao distúrbio, o qual é dado por:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t-1) + D(q^{-1})v(t) + \frac{e(t)}{\Delta} \quad (3.47)$$

onde a variável  $v(t)$  é o distúrbio medido no tempo  $t$  e  $D(q^{-1})$  é um polinômio definido como (3.48) e as outras variáveis são as mesmas mencionadas anteriormente.

$$D(q^{-1}) = d_0 + d_1q^{-1} + d_2q^{-2} + \dots + d_{nd}q^{-nd} \quad (3.48)$$

Multiplicando (3.47) por  $\Delta E_j(q^{-1})q^j$ :

$$\begin{aligned} E_j(q^{-1})\tilde{A}(q^{-1})y(t+j) &= E_j(q^{-1})B(q^{-1})\Delta u(t+j-d-1) \\ &+ E_j(q^{-1})D(q^{-1})\Delta v(t+j) + E_j(q^{-1})e(t+j) \end{aligned} \quad (3.49)$$

Usando (3.32) e algumas manipulações algébricas tem-se:

$$\begin{aligned} y(t+j) &= F_j(q^{-1})y(t) + E_j(q^{-1})B(q^{-1})\Delta u(t+j-d-1) \\ &+ E_j(q^{-1})D(q^{-1})\Delta v(t+j) + E_j(q^{-1})e(t+j) \end{aligned} \quad (3.50)$$

Como o polinômio  $E_j(q^{-1})$  tem grau  $j-1$ , os termos que multiplicam o ruído  $e(t+j)$  estão no futuro. Então, aplicando  $E[y(t+j)|y_0, y_1, \dots, y(t)]$  tem-se:

$$\hat{y}(t+j) = G_j(q^{-1})\Delta u(t+j-d-1) + F_j(q^{-1})y(t) + E_j(q^{-1})D(q^{-1})\Delta v(t+j)$$

Definindo o polinômio  $M_j(q^{-1}) = E_j(q^{-1})D(q^{-1})$ , onde:

$$M_j = m_0 + m_1q^{-1} + m_2q^{-2} + \dots + m_{nd+j-1}q^{-(nd+j-1)}$$

Usando os polinômios  $H_j$  e  $\bar{H}_j$  definidos em (3.36) e (3.37), a equação de predição para  $j = d + l$  será dada por:

$$\begin{aligned} \hat{y}(t + d + l) &= H_{d+l}(q^{-1})\Delta u(t + l - 1) + F_{d+l}(q^{-1})y(t) + q^l\bar{H}_{d+l}(q^{-1})\Delta u(t - 1) \\ &+ M_{d+l}(q^{-1})\Delta v(t + d + l) \end{aligned} \quad (3.51)$$

Note que na equação (3.51), os três últimos termos do lado direito dependem apenas dos valores passados da saída do processo, da variável de entrada e do distúrbio medido, e correspondem a resposta livre do processo, enquanto o primeiro termo depende dos valores futuros do sinal de controle e podem ser considerados a resposta forçada. O último termo da equação (3.51) depende de um distúrbio determinístico futuro. Se o distúrbio futuro for supostamente constante e igual ao último valor medido ( $v(t + j) = v(t)$ ), então  $\Delta v(t + j) = 0$  e assim o último termo de (3.51) pode ser desprezado. Deste modo, (3.51) pode ser escrita como:

$$\hat{y}(t + d + l) = H_{d+l}\Delta u(t + l - 1) + f_{d+l}$$

$$\text{com } f_{d+l} = F_{d+l}(q^{-1})y(t) + M_{d+l}(q^{-1})\Delta v(t + d + l) + q^l\bar{H}_{d+l}(q^{-1})\Delta u(t - 1).$$

Considerando agora um conjunto de previsões  $j = d + l, l = 1 \dots N$  passos a frente:

$$\begin{aligned} \hat{y}(t + d + 1) &= H_{d+1}(q^{-1})\Delta u(t) + f_{d+1} \\ \hat{y}(t + d + 2) &= H_{d+2}(q^{-1})\Delta u(t + 1) + f_{d+2} \\ &\vdots \\ \hat{y}(t + d + N) &= H_{d+N}(q^{-1})\Delta u(t + N - 1) + f_{d+N} \end{aligned} \quad (3.52)$$

Desta forma, as expressões podem ser escritas como:

$$\hat{y} = G\Delta u + f \quad (3.53)$$

onde  $G$  está definido em (3.43),  $\Delta u$  em (3.42). Observe que (3.53) tem a mesma forma da equação de previsão geral usada no caso sem distúrbio. O sinal de controle

futuro pode ser encontrado da mesma maneira, o que muda neste caso é a forma na qual encontra-se a resposta livre do processo, onde deve-se incluir os distúrbios medidos e os distúrbios futuros previstos. Portanto, a ação de controle será obtida através de (3.14) e leva em consideração os distúrbios (ação *feedforward*).

### 3.2.2 GPC Multivariável

Seja o sistema Multivariável com  $m$  entradas e  $n$  saídas:

$$\begin{aligned} \tilde{A}_i(q^{-1})y_i(t) &= q^{-d}B_{i,1}(q^{-1})\Delta u_1(t-1) + q^{-d}B_{i,2}(q^{-1})\Delta u_2(t-1) \\ &+ \dots + q^{-d}B_{i,m}\Delta u_m(t-1), \quad i = 1, 2, \dots, m \end{aligned} \quad (3.54)$$

Sejam os polinômios  $E_j^i$  e  $F_j^i$  com graus de  $j-1$  e  $na$  respectivamente, unicamente determinados pela Equação Diofantina,

$$1 = E_j^i(q^{-1})\tilde{A}(q^{-1}) + q^{-j}F_j^i(q^{-1}) \quad (3.55)$$

Da mesma forma que no caso monovariável, define-se os polinômios:

$$G_j^{i,\ell}(q^{-1}) = E_j^i(q^{-1})B_{i,\ell}(q^{-1}) = g_0^{i,\ell} + g_1^{i,\ell}q^{-1} + g_2^{i,\ell}q^{-2} + \dots + g_{nb+j-1}^{i,\ell}q^{-(nb+j-1)}$$

$$H_j^{i,\ell}(q^{-1}) = g_0^{i,\ell} + g_1^{i,\ell}q^{-1} + g_2^{i,\ell}q^{-2} + \dots + g_{j-d-1}^{i,\ell}q^{-(j-d-1)} \quad (3.56)$$

$$\bar{H}_j^{i,\ell}(q^{-1}) = G_j^{i,\ell}(q^{-1}) - H_j^{i,\ell}(q^{-1}) = g_{j-d}^{i,\ell}q^{-(j-d)} + \dots + g_{nb+j-1}^{i,\ell}q^{-(nb+j-1)} \quad (3.57)$$

A predição da saída  $i$ ,  $j$  passos a frente é:

$$\begin{aligned} \hat{y}_i(t+j) &= H_j^{i,1}\Delta u_1(t+j-d-1) + H_j^{i,2}\Delta u_2(t+j-d-1) \\ &+ \dots + H_j^{i,m}\Delta u_m(t+j-d-1) + \bar{H}_j^{i,1}\Delta u_1(t-1) \\ &+ \bar{H}_j^{i,2}\Delta u_2(t-1) + \dots + \bar{H}_j^{i,m}\Delta u_m(t-1) + F_j^i y_i(t) \end{aligned} \quad (3.58)$$

Sejam os vetores:

$$H_j^i = [H_j^{i,1} \quad H_j^{i,2} \quad \dots \quad H_j^{i,m}] \quad (3.59)$$

$$\bar{H}_j^i = [\bar{H}_j^{i,1} \quad \bar{H}_j^{i,2} \quad \dots \quad \bar{H}_j^{i,m}] \quad (3.60)$$

Considerando (3.23) e os vetores (3.59) e (3.60), pode-se escrever:

$$\hat{y}_i(t+j) = H_j^i \Delta \bar{u}(t+j-d-1) + \bar{H}_j^i \Delta \bar{u}(t-1) + F_j^i y_i(t) \quad (3.61)$$

Definindo os vetores:

$$\hat{y}(k) = \begin{bmatrix} \hat{y}_1(t) \\ \hat{y}_2(t) \\ \vdots \\ \hat{y}_n(t) \end{bmatrix}_{n \times 1} \quad H_j = \begin{bmatrix} H_j^1 \\ H_j^2 \\ \vdots \\ H_j^n \end{bmatrix}_{n \times m} \quad \hat{H}_j = \begin{bmatrix} \hat{H}_j^1 \\ \hat{H}_j^2 \\ \vdots \\ \hat{H}_j^n \end{bmatrix}_{n \times m} \quad F_j(t) = \begin{bmatrix} F_j^1 y_1(t) \\ F_j^2 y_2(t) \\ \vdots \\ F_j^n y_n(t) \end{bmatrix}_{n \times m}$$

As previsões das  $n$  saídas  $j$  passos a frente, podem ser escritas na seguinte forma matricial:

$$\hat{y}(t+j) = H_j \Delta \bar{u}(t+j-d-1) + \bar{H}_j \Delta \bar{u}(t-1) + F_j(t) \quad (3.62)$$

Realizando previsões  $d+1$  a  $N_2$  passos a frente, chega-se a seguinte matriz de previsões:

$$\begin{bmatrix} \hat{y}(t+d+1) \\ \hat{y}(t+d+2) \\ \vdots \\ \hat{y}(t+d+N_2) \end{bmatrix}_{(nN) \times 1} = \begin{bmatrix} H_{d+1} \Delta \bar{u}(t) \\ H_{d+2} \Delta \bar{u}(t+1) \\ \vdots \\ H_{d+N_2} \Delta \bar{u}(t+N_u-1) \end{bmatrix}_{(nN) \times m} \quad (3.63)$$

$$+ \begin{bmatrix} \bar{H}_{d+1} \\ \bar{H}_{d+1} \\ \vdots \\ \bar{H}_{d+N_u} \end{bmatrix}_{nN \times m} \times \Delta u(k-1)_{m \times 1} + \begin{bmatrix} F_{d+1} \\ F_{d+2} \\ \vdots \\ F_{d+N_u} \end{bmatrix}_{nN \times m}$$

onde:  $N = N_2 - d$ . Tendo em vista as definições de  $H_j$  e  $\bar{H}_j$ , pode-se escrever:

$$\begin{aligned}
\begin{bmatrix} \hat{y}(t+d+1) \\ \hat{y}(t+d+2) \\ \vdots \\ \hat{y}(t+d+N_u) \\ \vdots \\ \hat{y}(t+d+N_2) \end{bmatrix}_{(nN) \times 1} &= \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_u-d-1} & g_{N_u-d-2} & \dots & g_0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2-d-1} & g_{N_2-d-2} & \dots & g_{N_2-N_u-d-1} \end{bmatrix}_{(nN) \times (mN_u)} \times \quad (3.64) \\
&\times \begin{bmatrix} \Delta \bar{u}(t) \\ \Delta \bar{u}(t+1) \\ \vdots \\ \Delta \bar{u}(t+N_u-1) \end{bmatrix}_{(mN_u) \times 1} + \begin{bmatrix} \bar{H}_{d+1} \\ \bar{H}_{d+2} \\ \vdots \\ \bar{H}_{d+N_u} \\ \vdots \\ \bar{H}_{d+N_2} \end{bmatrix}_{(nN) \times m} \times \Delta \bar{u}(t-1)_{m \times 1} \\
&+ \begin{bmatrix} F_{d+1}(k) \\ F_{d+2}(k) \\ \vdots \\ F_{d+N_u}(k) \\ \vdots \\ F_{d+N_2}(k) \end{bmatrix}_{(nN) \times 1}
\end{aligned}$$

onde:

$$g_t = \begin{bmatrix} g_t^{1,1} & g_t^{1,2} & \dots & g_t^{1,m} \\ g_t^{2,1} & g_t^{2,2} & \dots & g_t^{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ g_t^{n,1} & g_t^{n,1} & \dots & g_t^{n,m} \end{bmatrix} \quad (3.65)$$

Desta maneira, pode-se escrever:

$$\hat{Y} = \bar{G} \Delta \bar{U} + \bar{F} \quad (3.66)$$

onde:

$$\hat{Y} = \begin{bmatrix} \hat{y}(t+d+1) \\ \hat{y}(t+d+2) \\ \vdots \\ \hat{y}(t+d+N_u) \\ \vdots \\ \hat{y}(t+d+N_2) \end{bmatrix}_{(nM) \times 1} \quad G = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_u-d-1} & g_{N_u-d-2} & \dots & g_0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2-d-1} & g_{N_2-d-2} & \dots & g_{N_2-N_u-d-1} \end{bmatrix}_{(nN) \times (mN_u)}$$

$$\Delta\bar{U} = \begin{bmatrix} \Delta\bar{u}(t) \\ \Delta\bar{u}(t+1) \\ \vdots \\ \Delta\bar{u}(t+N_u-1) \end{bmatrix}_{(mN_u) \times 1} \quad \bar{F} = \begin{bmatrix} \bar{H}_{d+1} \\ \bar{H}_{d+2} \\ \vdots \\ \bar{H}_{d+N_u} \\ \vdots \\ \bar{H}_{d+N_2} \end{bmatrix}_{(nN) \times m} \quad \times \Delta\bar{u}(t-1)_{m \times 1} + \begin{bmatrix} F_{d+1}(t) \\ F_{d+2}(t) \\ \vdots \\ F_{d+N_u}(t) \\ \vdots \\ F_{d+N_2}(t) \end{bmatrix}_{(nN) \times 1}$$

### 3.3 Conclusão do Capítulo

Neste capítulo, foi feita uma revisão bibliográfica a respeito dos controladores preditivo lineares mais usados na literatura. Primeiro, foram descritos o DMC monovariável e multivariável, no qual foram mostrados os passos para o desenvolvimento deste algoritmo. Além disto, relatou-se também, que este tipo de algoritmo usa como modelo a entrada ao degrau que é mais simples de implementar, porém só pode ser utilizado em processos estáveis.

Em seguida, descreveu-se o Controlador Preditivo Generalizado (GPC) monovariável e multivariável com e sem distúrbio. Foram mostrados, assim como o DMC, os passos para a execução deste algoritmo. Basicamente o que diferencia o GPC do DMC é o método de obtenção da matriz de previsão, pois a forma de calcular a solução minimizando a função custo é a mesma. No entanto, constatou-se que o modelo utilizado no GPC é o modelo de função de transferência do tipo CARIMA e o mesmo pode ser utilizado para processos estáveis ou não, com atraso de transporte e de fase não mínima. Observou-se que as dimensões das matrizes de previsão e das matrizes de restrições do DMC e do GPC, aumentam proporcionalmente com os horizontes de controle e de previsão, contribuindo para que o tempo de processamento seja mais longo, além de aumentar a possibilidade da obtenção de soluções não factíveis. Daí a necessidade de realizar uma sintonia adequada do controlador preditivo, conforme será visto no capítulo seguinte.

## Capítulo 4

# Sintonia do MPC Linear

O bom desempenho do controlador preditivo é obtido somente se os parâmetros de sintonia, horizonte de predição ( $N_2$ ), horizonte de controle ( $N_u$ ), ponderação do erro de previsão ( $\delta$ ), fator de supressão ( $\lambda$ ), constante da trajetória de referência  $\alpha$ , são adequadamente definidos pelo projetista. Entretanto, a seleção destes parâmetros não é uma tarefa fácil, na medida em que não há nenhum guia preciso para a seleção que assegure a estabilidade [10].

Em geral esta sintonia é feita através de tentativa e erro, tendo em vista que o incremento no parâmetro  $\lambda$  diminui a magnitude (esforço) da variável manipulada, tornando a resposta mais lenta e suave; variações em  $\lambda$  também afetam a robutez do controlador, de maneira que o aumento neste parâmetro proporciona maior robustez; o incremento nos horizontes de controle e de previsão contribui para melhorar a estabilidade do controlador, o sobressinal e o tempo de subida, mas prejudica a robustez do controlador. Observa-se também que sintonizar o parâmetro  $\lambda$ , com os parâmetros  $N_2$  e  $N_u$  fixos, não conduz a bons resultados; mas o caso contrário permite realizar a sintonia do controlador com mais facilidade. Isto mostra que os horizontes de controle e de predição possuem maior prioridade no ajuste em relação ao parâmetro  $\lambda$ .

Este tipo de heurística, baseada em tentativa e erro, é facilmente aplicada na sintonia do controlador preditivo de plantas que possuem um comportamento estável em malha aberta. No entanto, nos casos em que a planta é instável em malha aberta, como é o nível do molde, e/ou plantas Multivariáveis, como o Laminador, este procedimento torna-se entediante e, na maioria das vezes quando o processo é

estabilizado, o desempenho do controlador preditivo fica abaixo das expectativas.

A seguir serão apresentados os métodos que utilizam algum automatismo na definição dos parâmetros de sintonia para o DMC.

## 4.1 Sintonia através do Guia Proposto em *Dougherty e Cooper 2003* [34]

O método apresentado por *Dougherty e Cooper 2003*, [34], apresenta uma expressão analítica para calcular  $N_2$ ,  $N_u$  e o fator energético  $\lambda$  para sistemas SISO e MIMO. O guia proposto em [34] para ajuste dos parâmetros do DMC é descrito nos seguintes passos:

1. Aproximar a dinâmica do processo para um processo de primeira ordem com um atraso de transporte (FOPDT):

$$\frac{y_j(s)}{u_\ell(s)} = \frac{K_{\ell j} e^{-(\theta_{\ell j})s}}{\tau_{\ell j} s + 1}$$

para  $\ell = 1, 2, \dots, m; j = 1, 2, \dots, n$

no qual,  $K$  é o ganho,  $\theta$  é o atraso de tempo e  $\tau$  é a constante de tempo.

2. Selecionar o período de amostragem o mais próximo possível para:

$$T = \min[\max(0.1\tau_{\ell j}, \theta_{\ell j})]$$

3. Calcular o horizonte de predição  $N_2$  e o horizonte de modelo  $N_s$ :

$$N_2 = N_s = \max \left[ \left( \frac{5\tau_{\ell j}}{T} + k_{\ell j} \right) \right]$$

onde  $k_{\ell j} = \text{int} \left( \frac{\theta_{\ell j}}{T} + 1 \right)$

4. Calcular o horizonte de controle  $N_u$

$$N_u = \max \left[ \left( \frac{\tau_{\ell j}}{T} + k_{\ell j} \right) \right]$$

5. Selecionar o fator de peso  $\gamma_j^2$
6. Calcular o fator energético:

$$\lambda_\ell^2 = \frac{N_u}{10} \sum_{j=1}^m \left[ \gamma_\ell^2 K_{\ell j}^2 \left( N_2 - k_{\ell j} - \frac{3}{2} \frac{\tau_{\ell j}}{T} + 2 - \frac{(N_u - 1)}{2} \right) \right] \quad (4.1)$$

7. Implementar o DMC usando os parâmetros calculados nos passos 1-6.

## 4.2 Sintonia do MPC Através do Algoritmo Genético

Uma das principais características do AG é realizar ajuste de todos os parâmetros de maneira automática. A execução do algoritmo é feita da seguinte maneira: Primeiramente, deve-se fornecer o modelo do processo. Em seguida definem-se os parâmetros necessários para a execução do AG tais como: o tamanho da população ( $M$ ), o número de gerações ( $G$ ), a taxa de *crossover* e a taxa de mutação, o tipo de função *fitness* considerada e o critério de seleção. Feito isto,  $M$  indivíduos serão criados aleatoriamente por AG, que terão o formato conforme (4.2).

$$[N_u, N_2, \alpha, \delta, \lambda] \quad (4.2)$$

Depois disto, o AG calculará o *fitness* de cada indivíduo dentro da população inicial, onde os parâmetros sintonizados por AG serão colocados dentro do algoritmo de MPC com o objetivo de separar os melhores e piores indivíduos de acordo com o *fitness* de cada um. A função *fitness* utilizada neste trabalho pode ser vista em (4.3).

$$Fit(N_u, N_2, \alpha, \delta, \lambda) = \frac{1}{\sum abs(y - w)^2} \quad (4.3)$$

onde  $y$  é a saída do processo e  $w$  é a referência.

Logo após, o AG irá fazer a evolução da espécie através da realização das três operações genéticas mencionadas, onde os melhores indivíduos terão uma proba-

bilidade maior de terem descendentes nas gerações futuras. É importante mencionar que neste trabalho, foi utilizada a técnica de elitismo paralelo à seleção por roleta, onde se assegura que os resultados nunca serão piores do que os da geração anterior, pois esta técnica garante os melhores indivíduos na geração seguinte.

O AG irá executar um *loop* de  $G$  gerações, e ao final irá apresentar o melhor indivíduo que será o resultado da simulação. Como neste trabalho não se conhece o valor do ponto ótimo a que se deseja chegar, a validade dos resultados apresentados pelo AG foi comprovada através de inúmeras simulações e percebeu-se que o algoritmo sempre convergia para valores muito próximos, podendo assim ser considerado como um bom resultado.

O algoritmo utilizado pode ser visto a seguir:

- Passo 1:** Determinar o tamanho da população  $M$ , a taxa de *crossover* a taxa de mutação;
- Passo 2:** Criar a população inicial contendo  $M$  indivíduos com o formato de acordo com (4.2);
- Passo 3:** Aplicar os parâmetros calculados por AG dentro do algoritmo do MPC, seja ele restrito ou irrestrito;
- Passo 4:** Determinar a Matriz Dinâmica  $G$  de acordo com (3.8);
- Passo 5:** Determinar a referência futura  $w$  (3.11) e a resposta livre  $f$  (3.6);
- Passo 6:** Determinar o ganho  $K$  (3.15);
- Passo 7:** Determinar o sinal de controle  $\Delta u$  (3.14) e aplicá-lo no processo para determinar a saída do processo;
- Passo 6:** Calcular o *fitness* de cada indivíduo através de (4.3);
- Passo 7:** Distinguir os melhores e piores indivíduos;
- Passo 8:** Criar uma nova população usando as operações genéticas;
- Passo 9:** Ir para o passo 2 até o número máximo de gerações ser alcançado;
- Passo 10:** Apresentar o resultado.

Com este procedimento, entra-se com o modelo da planta que se deseja controlar para que o algoritmo calcule os valores dos parâmetros do MPC de forma automática sem a necessidade de um conhecimento prévio do processo. É importante mencionar que o algoritmo desenvolvido nesta tese pode ser utilizado para qualquer tipo de processo no qual se possa utilizar o MPC, o que o faz ter uma ampla faixa de utilização.

## 4.3 Comparações entre as Técnicas de Sintonia

Nesta seção, serão mostrados os resultados dos algoritmos de MPC sintonizado com AG e de acordo com *Dougherty e Cooper 2003* [34] para processo linear, monovariável e multivariável, restrito.

### 4.3.1 DMC Monovariável Restrito

Para realizar a comparação entre estas metodologias, foram selecionados quatro classes de modelos que, segundo [5], podem ser caracterizados como representativos da maioria dos processos industriais mais comuns. Tais sistemas estão representados pelas seguintes funções de transferências contínuas:

$$G_1(s) = \frac{1}{(\mu s + 1)^2} \quad \text{para } \mu = 0, 1; 3 \text{ e } 10 \quad (4.4)$$

$$G_2(s) = \frac{1}{(s + 1)^4} \quad (4.5)$$

$$G_3(s) = \frac{1}{(s + 1)(s + \mu)(s + \mu^2)(s + \mu^3)} \quad \text{para } \mu = 0, 2 \text{ e } 0,7 \quad (4.6)$$

$$G_4(s) = \frac{1 - 0.1s}{(s + 1)^3} \quad (4.7)$$

Com os diferentes parâmetros para cada classe de função de transferência, tem-se ao todo sete modelos de sistemas. As restrições foram aplicadas no sinal de controle  $u(k)$ , na variação do sinal de controle  $\Delta u(k)$  e na saída do sistema  $y(k)$ . As

restrições são:

$$\begin{aligned}
 -1 &\leq u_i(k) \leq 1 \\
 -0.1 &\leq \Delta u_i(k) \leq 0.1 \\
 0 &\leq y_i(k) \leq 1.0
 \end{aligned} \tag{4.8}$$

A população gerada pelo AG tem 50 indivíduos, o número de gerações é 30 e a taxa de *crossover* e mutação é de 0.8 e 0.01 respectivamente para todas as simulações.

### Função $G_1$

Primeiro, será feito a aproximação da função  $G_1$  para os três valores de  $\mu$  para um modelo de Primeira Ordem com Atraso de Transporte (FOPDT), conforme (4.9), pois a sintonia feita por *Dougherty e Cooper 2003* [34], o modelo necessita deste formato.

$$\frac{Y(s)}{U(s)} = \frac{k_p e^{-\theta s}}{\tau s + 1} \tag{4.9}$$

Usando a aproximação por Degrau apresentada em [20], as funções de transferência para os três casos são:

$$G_1(s) \simeq \frac{1e^{-0.0318s}}{0.2682s + 1} \quad \mu = 0.1 \tag{4.10}$$

$$G_1(s) \simeq \frac{1e^{-0.945s}}{8.055s + 1} \quad \mu = 3 \tag{4.11}$$

$$G_1(s) \simeq \frac{1e^{-3.1s}}{26.9s + 1} \quad \mu = 10 \tag{4.12}$$

A Tabela 4.1 compara os parâmetros de sintonia do DMC, o índice de desempenho ISE e o número total de restrições NR, para os casos  $\mu = 0.1$ ,  $\mu = 3$ ,  $\mu = 10$ , respectivamente, usando AG e o guia proposto por *Dougherty e Cooper 2003* [34].

Tabela 4.1: Parâmetros do DMC para  $G_1$ .

| Método | $N_u$ |    |    | $N_2$ |    |    | $\lambda$ |       |       | ISE  |       |       | NR  |     |     |
|--------|-------|----|----|-------|----|----|-----------|-------|-------|------|-------|-------|-----|-----|-----|
|        | 0,1   | 3  | 10 | 0,1   | 3  | 10 | 0,1       | 3     | 10    | 0,1  | 3     | 10    | 0,1 | 3   | 10  |
| GA     | 1     | 1  | 1  | 4     | 3  | 8  | 5.58      | 0.223 | 0.432 | 2.85 | 7.175 | 16.2  | 12  | 10  | 20  |
| Guia   | 13    | 13 | 13 | 53    | 53 | 53 | 40.3      | 40.3  | 40.3  | 3.09 | 7.496 | 16.53 | 158 | 158 | 158 |

As Figuras 4.1, 4.2 e 4.3 mostram a comparação da saída  $y$ , do sinal de controle  $u$  e da variação de controle  $\Delta u$ .

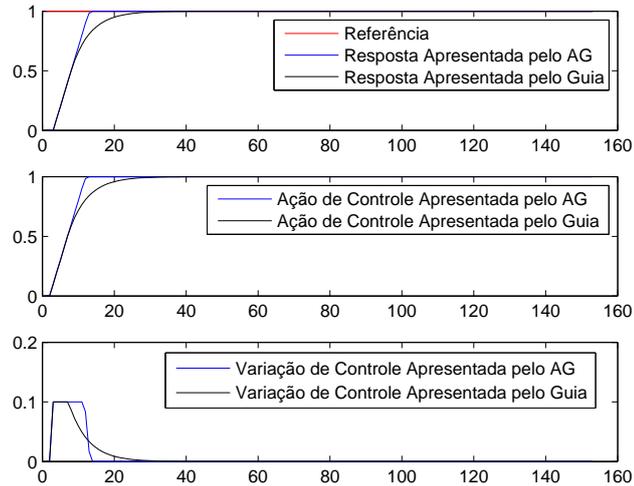


Figura 4.1: Sinal de Resposta  $y$ ,  $u$  e  $\Delta u$  para  $G_1$  com  $\mu = 0, 1$ .

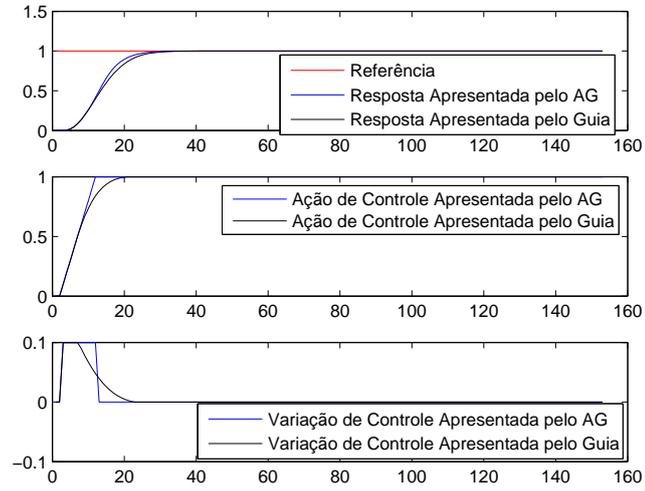


Figura 4.2: Sinal de Resposta  $y, u$  e  $\Delta u$  para  $G_1$  com  $\mu = 3$ .

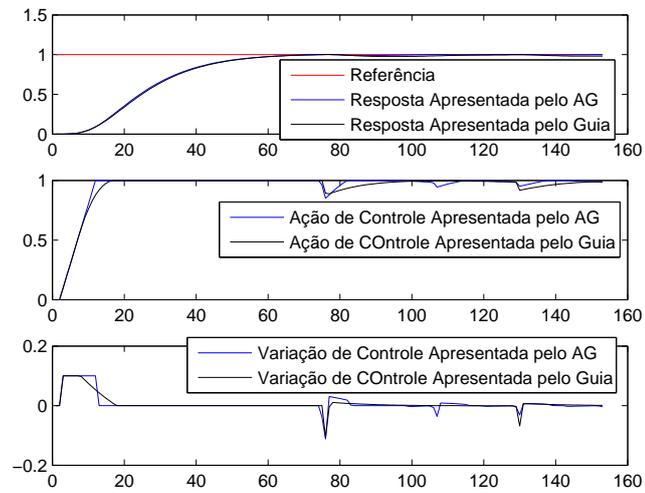


Figura 4.3: Sinal de Resposta  $y, u$  e  $\Delta u$  para  $G_1$  com  $\mu = 10$ .

## Função $G_2$

Da mesma forma que a função anterior, será feito também uma aproximação de  $G_2$  para uma função do tipo FOPDT, usando o método de aproximação ao Degrau apresentado em [20]. A função aproximada pode ser vista na Equação 4.11:

$$G_2 \simeq \frac{1e^{-1.46s}}{4.39s + 1} \quad (4.13)$$

A Tabela 4.2 mostra os parâmetros do DMC sintonizado por AG e por *Dougherty e Cooper* [34].

Tabela 4.2: Parâmetros do DMC para  $G_2$ .

| Método | $N_u$ | $N_2$ | $\lambda$ | $ISE$  | $NR$ |
|--------|-------|-------|-----------|--------|------|
| GA     | 1     | 5     | 1.796     | 8.0068 | 14   |
| Guia   | 10    | 34    | 18.55     | 8.072  | 108  |

A Figura 4.4 mostra a comparação da saída  $y$ , do sinal de controle  $u$  e da variação de controle  $\Delta u$  para os dois casos.

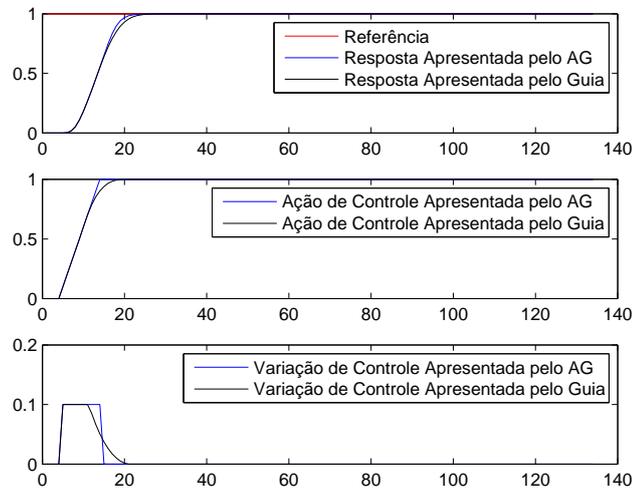


Figura 4.4: Sinal de Resposta  $y$ ,  $u$  e  $\Delta u$  para  $G_2$ .

### Função $G_3$

Da mesma forma que a função anterior, será feito também uma aproximação da função  $G_3$  para os dois valores de  $\mu$  para uma função do tipo FOPDT, usando o método de aproximação ao Degrau apresentado em [20]. As funções aproximadas podem ser vistas em (4.14) e (4.15).

$$G_3(s) \simeq \frac{15622e^{-12.57s}}{50.83s + 1} \quad \mu = 0,2 \quad (4.14)$$

$$G_3(s) \simeq \frac{8.5e^{-3.77s}}{8.66s + 1} \quad \mu = 0,7 \quad (4.15)$$

A Tabela 4.3 compara os parâmetros do DMC sintonizado por AG e por *Dougherty e Cooper* [34] para o caso onde  $\mu = 0,2$  e  $\mu = 0,7$  respectivamente.

Tabela 4.3: Parâmetros do DMC para  $G_3$ .

| Método | $N_u$ |     | $N_2$ |     | $\lambda$     |       | $ISE$ |       | $NR$ |     |
|--------|-------|-----|-------|-----|---------------|-------|-------|-------|------|-----|
|        | 0,2   | 0,7 | 0,2   | 0,7 | 0,2           | 0,7   | 0,2   | 0,7   | 0,2  | 0,7 |
| GA     | 10    | 1   | 40    | 12  | $9,57 * 10^8$ | 46,19 | 54,71 | 6,402 | 120  | 28  |
| Guia   | 12    | 8   | 44    | 26  | $7,26 * 10^9$ | 842,7 | 69,53 | 8,438 | 136  | 84  |

As Figura 4.5 e 4.6 comparam a saída  $y$ , o sinal de controle  $u$  e a variação de controle  $\Delta u$  para os dois casos.

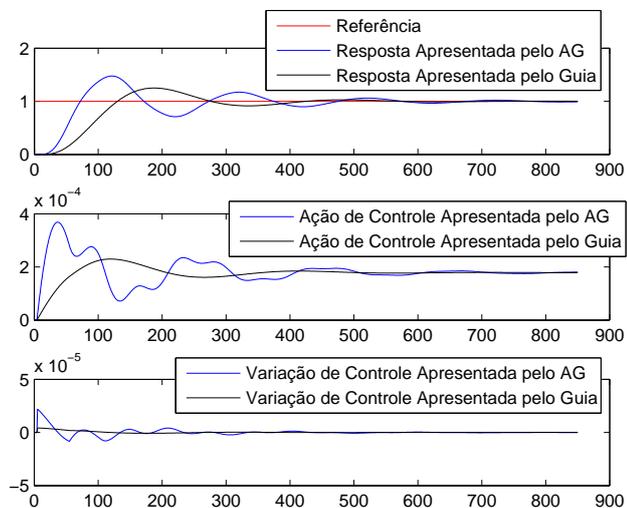


Figura 4.5: Sinal de Resposta  $y$ ,  $u$  e  $\Delta u$  para  $G_3$  com  $\mu = 0, 2$ .

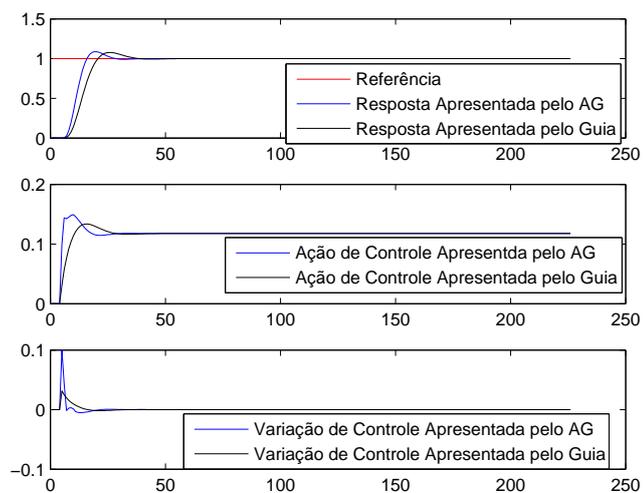


Figura 4.6: Sinal de Resposta  $y$ ,  $u$  e  $\Delta u$  para  $G_3$  com  $\mu = 0, 7$ .

## Função $G_4$

Da mesma forma que a função anterior, será feito também uma aproximação de  $G_4$  para uma função do tipo FOPDT, usando o método de aproximação ao Degrau apresentado em [20]. A função aproximada pode ser vista em (4.16).

$$G_4(s) \simeq \frac{1e^{-0.9s}}{3.71s + 1} \quad (4.16)$$

A Tabela 4.4 compara os parâmetros do DMC sintonizados por AG e por *Dougherty e Cooper* [34].

Tabela 4.4: Parâmetros do DMC para  $G_4$ .

| Método | $N_u$ | $N_2$ | $\lambda$ | $ISE$   | $NR$ |
|--------|-------|-------|-----------|---------|------|
| GA     | 1     | 33    | 1.1511    | 22.9061 | 70   |
| Guia   | 12    | 45    | 30.4267   | 65.974  | 138  |

A Figura 4.7 mostra a saída  $y$ , o sinal de controle  $u$  e a variação de controle  $\Delta u$  para os dois casos.

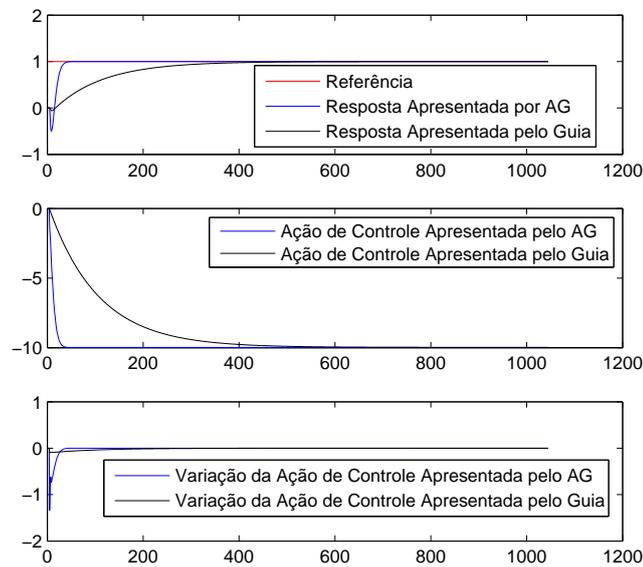


Figura 4.7: Sinal de Resposta  $y$ ,  $u$  e  $\Delta u$  para  $G_4$ .

É importante mencionar que para este caso, nenhuma das duas técnicas conseguiu ajustar os parâmetros para as restrições dada, desta forma, foi feita uma simulação irrestrita, uma vez que estes sistema são meramente ilustrativos.

Percebe-se que em todos os exemplos apresentados os parâmetros sintonizado por AG apresentam um índice de desempenho *ISE* menor do que o sintonizado por *Dougherty e Cooper 2003*, além de se ter horizontes menores, e, conseqüentemente, um número bem menor de restrições, tornando o cálculo menos complexos, uma vez que as dimensões das matrizes de previsão dependem diretamente dos valores destes horizontes.

### 4.3.2 DMC Multivariável Restrito

Outro exemplo a ser avaliado é um sistema multivariável com 2 entradas e 2 saídas, representado pela seguinte matriz função de transferência [34]:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21.0s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \quad (4.17)$$

A Tabela 4.5 compara o ajuste dos parâmetros, o índice de desempenho *ISE* e o número total de restrições (*NR*) para um sistema MIMO.

Tabela 4.5: Parâmetros do DMC para o sistema MIMO.

| Método | $N_u$ | $N_2$ | $\lambda_1$ | $\lambda_2$ | ISE 1  | ISE 2  | <i>NR</i> |
|--------|-------|-------|-------------|-------------|--------|--------|-----------|
| GA     | 1     | 10    | 812.07      | 121.91      | 5.1878 | 5.1059 | 48        |
| Guia   | 17    | 73    | 131.01      | 273.88      | 5.5869 | 5.4303 | 428       |

As Figuras 4.8 e 4.9 mostram o sinal de resposta apresentado pelos dois métodos, considerando uma entrada ao degrau e os parâmetros mostrados na Tabela 4.5.

Pode-se ver através da Tabela 4.5 que a sintonia realizada por AG assim como no caso monovariável, apresenta um  $N_u$  e  $N_2$  menor do que o ajustado por [34]. Conseqüentemente, o algoritmo do DMC ajustado pelo AG tem um número de restrições menor dentro de um problema de programação quadrática.

Pode se observar também através das respostas mostradas nas Figuras 4.8 e 4.9 que o problema de controle ajustado por [34], viola as restrições, ou seja, a

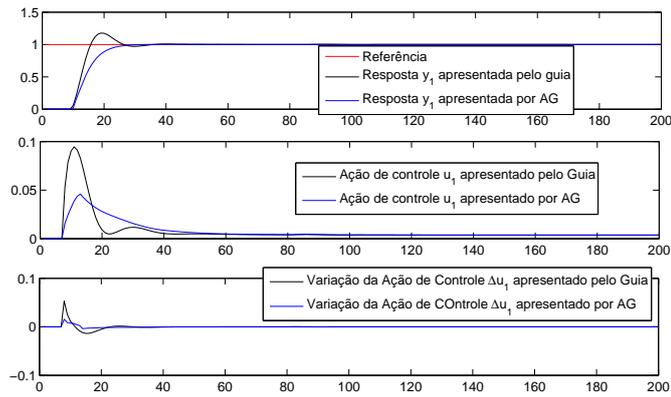


Figura 4.8: Sinal de Resposta  $y_1$ ,  $u_1$  e  $\Delta u_1$ .

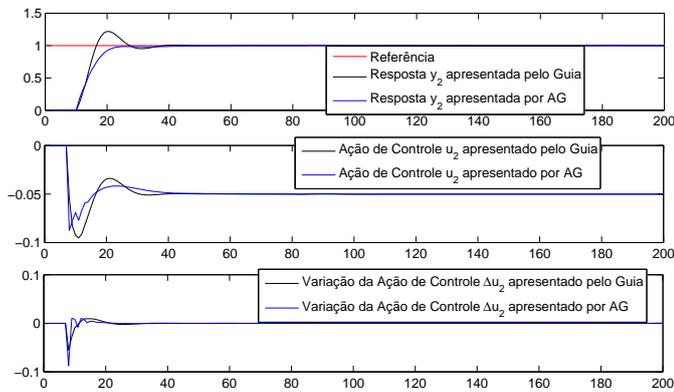


Figura 4.9: Sinal de Resposta  $y_2$ ,  $u_2$  e  $\Delta u_2$ .

saída apresenta um *overshoot*, além de se ter um índice de desempenho maior nos dois casos do que o apresentado pelos parâmetros sintonizados por AG.

Percebe-se através dos resultados, tanto monovariável, quanto multivariável, que os parâmetros sintonizado pelo AG, apresentam em todos os casos além de índice de desempenho menor, os horizontes de controle e de predição também são consideravelmente menor, o que acarreta em um número bem inferior de restrições, fazendo com que as restrições sejam respeitadas na grande maioria dos casos em AG e não respeitadas no guia proposto em [34].

O Algoritmo foi implementado em um Pentium 2.8GHz e 512MB. O soft-

ware MATLAB 7.0 e o problema de otimização foi solucionado pela rotina *quadprog*. O tempo para convergência do algoritmo foi cerca de 101.3 segundos para o caso SISO e 119.12 segundos para o caso MIMO.

## 4.4 GPC do Laminador Sintonizado Através do AG

Nesta seção, o GPC Multivariável é aplicado a um processo de laminação no qual se deseja a redução da variação da espessura de saída em relação aos distúrbios que o processo apresenta tais como: temperatura da tira e a espessura inicial.

Para o sistema considerado na Figura 4.10 as entradas de controle são a variação no *gap* nos *stands* F9,  $g_{in}^{(1)}$  e F10,  $g_{in}^{(2)}$  e a variação da velocidade de laminação no *stand* F9,  $\Delta V_{in}^{(1)}$ . As saídas controladas são as variações das espessuras de saída da chapa,  $\Delta h_2^{(1)}$ ,  $\Delta h_2^{(2)}$ , e a tensão entre os stands ( $\Delta \sigma_1$ ). O principal objetivo de controle é manter a variação da espessura da tira ( $\Delta h_2$ ) próxima de zero, para qualquer variação de temperatura e espessura de entrada do sistema.

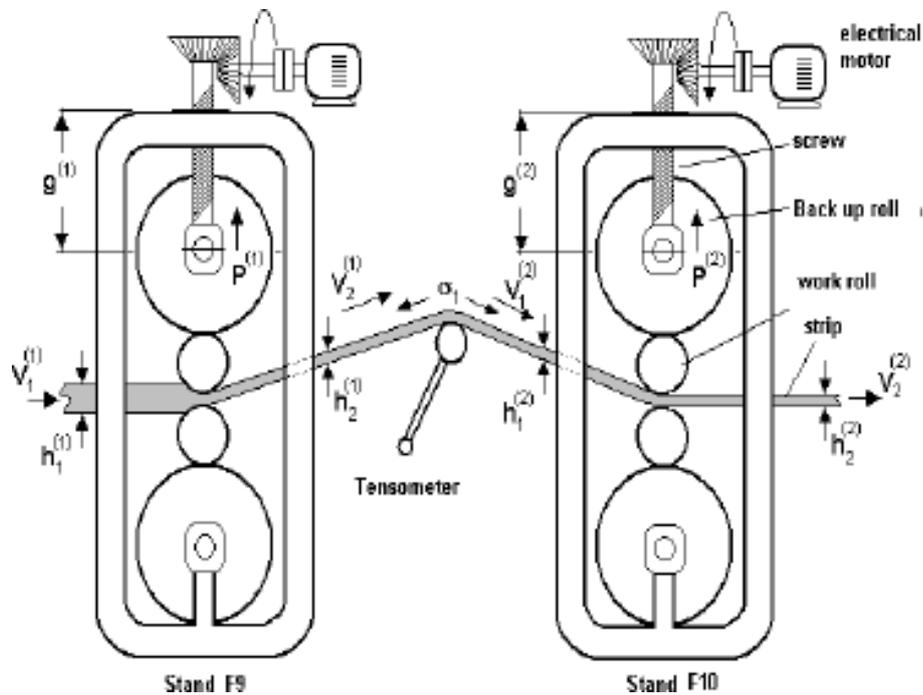


Figura 4.10: Cadeiras de Laminação.

Este problema de controle é solucionado na indústria usando a técnica de controle *Feed Forward Forced(FFF)* a qual é baseada na detecção da dureza das flutuações do material que passa através do primeiro *stand* e envia informações para o *set up*. Em [100] e [99] foi aplicado, respectivamente, um Controle Linear Ótimo e um Controle Neural Adaptativo usando o modelo inverso [98], e foi mostrado que o desempenho dos sistemas de controle é melhor que o controlador FFF. Porém, não foram consideradas as restrições nas entradas, estados e saídas do processo.

De acordo com [93], um controlador para processos industriais, deve manter o sistema o mais próximo possível de sua referência, sem violar as restrições, e esta é a principal razão para a técnica de Controle Preditivo estar crescendo na indústria atualmente.

## Modelo do Processo

O modelo matemático implementado em [100] considera as duas últimas cadeiras de laminação. Nesta parte, será feito um ajuste de espessura com a técnica GPC e será feita uma comparação com o FFF. O modelo matemático usado no GPC, é o modelo CARIMA. A espessura de saída de cada cadeira é uma função do gap dos cilindros e da deformação do sistema, sendo dada por:

$$h_2^{(i)} = gp^{(i)} + \frac{P^{(i)}}{K^{(i)}} \quad (4.18)$$

- $(i)$ : Cadeira de Laminação  $i = 1$  (F9) e  $i = 2$  (F10);
- $h_2^{(i)}$ : Espessura de saída da tira na cadeira  $(i)$ ;
- $gp^{(i)}$ : Gap dos cilindros na cadeira  $(i)$ ;
- $P^{(i)}$ : Carga de laminação na cadeira  $(i)$ ;
- $K^{(i)}$ : Módulo de rigidez da cadeira  $(i)$ .

A derivada da tensão na tira entre as cadeiras é uma função da diferença entre a velocidade de saída da tira e a velocidade de entrada da tira, nas cadeiras F9 e F10:

$$\frac{d\sigma_1}{dt} = \frac{E}{L}(V_2^{(1)} - V_1^{(2)}) \quad (4.19)$$

- $\sigma_1$ : Tensão da tira entre as cadeiras F9 e F10;
- $E$ : Modulo de Young da tira, 21.000N/m;
- $L$ : Distância entre as cadeiras ( $i$ ), 5.486m;
- $V_2^{(1)}$ : Velocidade de saída da tira na cadeira F9;
- $V_1^{(1)}$ : Velocidade de entrada da tira na cadeira F10.

O volume de continuidade da tira dentro do gap dos cilindros, é definido pela expressão:

$$V_1^{(i)} h_1^{(i)} = V_2^{(i)} h_2^{(i)} \quad (4.20)$$

A carga de laminação é representada por:

$$P^{(i)} = P(h_1^{(i)}, h_2^{(i)}, \sigma_1^{(i)}, \sigma_2^{(i)}, S^{(i)}, \mu^{(i)}, T^{(i)}) \quad (4.21)$$

- $\sigma_2^{(i)}$ : Tensão de saída da tira da cadeira ( $i$ );
- $\mu^{(i)}$ : Coeficiente de atrito da cadeira ( $i$ );
- $S^{(i)}$ : Limite de elasticidade da cadeira ( $i$ ).

A função do escorregamento é definida como:

$$f^{(i)} = f(h_1^{(i)}, h_2^{(i)}, \sigma_1^{(i)}) \quad (4.22)$$

### Considerações Básicas sobre o Modelo

- A principal consideração é que o controle será aplicado quando a tira estiver entre as duas cadeiras, isto significa que o momento anterior a entrada da tira na última cadeira não é considerado;
- Será considerada uma variação finita dos valores nominais.

iii) Também será considerado que a carga de laminação é uma função da espessura de entrada, espessura de saída, temperatura da tira e tensão na tira entre as cadeiras

$$\begin{aligned}\Delta P^{(1)} &= \beta_1 \Delta \sigma_1 + \beta_2 \Delta h_2^{(1)} + \beta_3 \Delta h_1^{(1)} + \beta_4 \Delta T^{(1)} \\ \Delta P^{(2)} &= \beta_5 \Delta \sigma_1 + \beta_6 \Delta h_2^{(2)} + \beta_7 \Delta h_1^{(2)} + \beta_8 \Delta T^{(2)}\end{aligned}\quad (4.23)$$

Os coeficientes da função linear são:

$$\begin{aligned}\beta_1 &= \frac{\partial P^{(1)}}{\partial \sigma_1}; \beta_2 = \frac{\partial P^{(1)}}{\partial h_2^{(1)}}; \beta_3 = \frac{\partial P^{(1)}}{\partial h_1^{(1)}}; \beta_4 = \frac{\partial P^{(1)}}{\partial T^{(1)}} \\ \beta_5 &= \frac{\partial P^{(2)}}{\partial \sigma_1}; \beta_6 = \frac{\partial P^{(2)}}{\partial h_2^{(2)}}; \beta_7 = \frac{\partial P^{(2)}}{\partial h_1^{(2)}}; \beta_8 = \frac{\partial P^{(2)}}{\partial T^{(2)}}\end{aligned}\quad (4.24)$$

iv) A variação do escorregamento é uma função da espessura de entrada e saída, e a tensão na tira, Equações (4.26) e (4.27):

$$\begin{aligned}\Delta f^{(1)} &= \alpha_1 \Delta \sigma_1 + \alpha_2 \Delta h_2^{(1)} + \alpha_3 \Delta h_1^{(1)} \\ \Delta f^{(2)} &= \alpha_4 \Delta \sigma_1 + \alpha_5 \Delta h_2^{(2)} + \alpha_6 \Delta h_1^{(2)}\end{aligned}\quad (4.25)$$

Os coeficientes são:

$$\begin{aligned}\alpha_1 &= \frac{\partial f^{(1)}}{\partial \sigma_1}; \alpha_2 = \frac{\partial f^{(1)}}{\partial h_2^{(1)}}; \alpha_3 = \frac{\partial f^{(1)}}{\partial h_1^{(1)}}; \\ \alpha_4 &= \frac{\partial f^{(2)}}{\partial \sigma_1}; \alpha_5 = \frac{\partial f^{(2)}}{\partial h_2^{(2)}}; \alpha_6 = \frac{\partial f^{(2)}}{\partial h_1^{(2)}};\end{aligned}\quad (4.26)$$

v) A variação da velocidade de saída na tira é expressa pela variação dos coeficientes do escorregamento e pela velocidade dos cilindros.

$$\begin{aligned}\Delta V_2^{(1)} &= (1 + f^{*(1)}) \Delta V^{(1)} + V^{*(1)} \Delta f^{(1)} \\ \Delta V_2^{(2)} &= (1 + f^{*(2)}) \Delta V^{(2)} + V^{*(2)} \Delta f^{(2)}\end{aligned}\quad (4.27)$$

vi) As variáveis controladas não são diretamente afetadas pela velocidade da última cadeira, sendo assim, tem-se  $\Delta V_2 = 0$ ;

vii) Cada atuador tem a sua própria dinâmica, que pode ser aproximada por um sistema de primeira ordem expresso como:

$$T_g = \frac{dgp^{(i)}}{dt} = gp_{in}^{(i)} - gp^{(i)} \quad (4.28)$$

$$T_v = \frac{dV^{(i)}}{dt} = V_{in}^{(i)} - V^{(i)} \quad (4.29)$$

viii) As variáveis  $\Delta h_1^{(1)}$ ,  $\Delta T^{(1)}$  e  $\Delta T^{(2)}$ , serão consideradas como distúrbios do processo;

ix) Considerando um atraso de transporte entre F9 e F10, a variação da espessura de saída na tira na cadeira F9 é igual a variação da espessura de entrada na tira na cadeira F10. O atraso de tempo é igual a  $T_d = l/(V + \Delta V)$ , assim,  $\Delta h_2^{(1)} = \Delta h_1^{(2)} e^{(-sT_d)}$ , onde  $s$  é a variável complexa de Laplace.

A matriz de funções de transferência do sistema é dada por:

$$\begin{bmatrix} \Delta h_2^{(1)} \\ \Delta \sigma_1 \\ \Delta h_2^{(2)} \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) & G_{13}(s) \\ G_{21}(s) & G_{22}(s) & G_{23}(s) \\ G_{31}(s) & G_{23}(s) & G_{33}(s) \end{bmatrix} \begin{bmatrix} \Delta gp_{in}^{(1)} \\ \Delta V_{in}^{(1)} \\ \Delta gp_{in}^{(2)} \end{bmatrix} \quad (4.30)$$

onde:

$$G_{11}(s) = \frac{0,00398s^3 - 0,0113s^2 + 0,0107s - 0,00336}{s^4 - 3,833s^3 + 5,503s^2 - 3,506s + 0,8363} \quad (4.31)$$

$$G_{12}(s) = \frac{4,622 * 10^{-3}s^3 - 3,91 * 10^{-5}s^2 - 4,59 * 10^{-5}s + 3,89 * 10^{-5}}{s^4 - 3,833s^3 + 5,503s^2 - 3,506s + 0,8363} \quad (4.32)$$

$$G_{13}(s) = \frac{-1,127 * 10^{-7}s^3 + 9,518 * 10^{-8}s^2 + 1,117 * 10^{-7}s - 9,469 * 10^{-8}}{s^4 - 3,833s^3 + 5,503s^2 - 3,506s + 0,8363} \quad (4.33)$$

$$G_{21}(s) = \frac{7,585 * 10^{-7}s^3 - 7,83 * 10^{-7}s^2 - 6,804 * 10^{-7}s + 7,057 * 10^{-7}}{s^4 - 3,833s^3 + 5,503s^2 - 3,506s + 0,8363} \quad (4.34)$$

$$G_{22}(s) = \frac{-0,000316s^3 + 0,000327s^2 + 0,00028s - 0,00029}{s^4 - 3,833s^3 + 5,503s^2 - 3,506s + 0,8363} \quad (4.35)$$

$$G_{23}(s) = \frac{7,7e^{-7}s^3 - 7,96 * 10^{-7}s^2 - 6,91 * 10^{-7}s + 7,16 * 10^{-7}}{s^4 - 3,833s^3 + 5,503s^2 - 3,506s + 0,8363} \quad (4.36)$$

$$G_{31}(s) = \frac{-3,92 * 10^{-8}s^3 + 3,315 * 10^{-8}s^2 + 3,89 * 10^{-8}s - 3,3 * 10^{-8}}{s^4 - 3,833s^3 + 5,503s^2 - 3,506s + 0,8363} \quad (4.37)$$

$$G_{32}(s) = \frac{1,635 * 10^{-5}s^3 - 1,38 * 10^{-5}s^2 - 1,623 * 10^{-5}s + 1,376 * 10^{-5}}{s^4 - 3,833s^3 + 5,503s^2 - 3,506s + 0,8363} \quad (4.38)$$

$$G_{33}(s) = \frac{0,0031s^3 - 0,0087s^2 + 0,00824s - 0,0026}{s^4 - 3,833s^3 + 5,503s^2 - 3,506s + 0,8363} \quad (4.39)$$

percebe-se que os pólos são  $1,0571 \pm 0,0985i$  e  $0,8594 \pm 0,0582i$  o que caracteriza um sistema instável, pois todos os pólos estão localizados no semi-plano direito do plano  $s$ . Desta forma, não é possível o uso do DMC, uma vez que o mesmo é utilizado em sistemas estáveis, portanto, será usado um GPC Multivariável.

Discretizando as equações para a obtenção do modelo CARIMA Multivariável, tem-se:

$$\begin{aligned} \tilde{A}_i(q^{-1})y_i(t) &= B_{i,1}u_1(t-1) + B_{i,2}u_2(t-1) + B_{i,3}u_3(t-1) \\ &+ D_{i,1}d_1(t-1) + D_{i,2}d_2(t-1) + D_{i,3}d_3(t-1) \\ &+ D_{i,4}d_4(t-1) + \frac{e(t)}{\Delta} \end{aligned} \quad (4.40)$$

no qual:  $\Delta = 1 - q^{-1}$  e os polinômios  $\tilde{A}_i$ ,  $B_{i,k}$  e  $D_{i,l}$ ,  $i = 1, 2, 3$ ,  $k = 1, 2, 3$ , e  $l = 1, 2, 3, 4$  tem ordem  $na_i$  e  $nb_{i,k}$  e  $nd_{i,l}$  respectivamente, e são obtidos a partir de uma discretização da matriz de função de transferência;  $e(t)$  é um ruído branco de média zero; as entradas, saídas e distúrbios ( $u_i$ ,  $y_i$ ,  $i = 1, 2, 3$  and  $d_i$ ,  $i = 1, 2, 3, 4$ ) são

$$\bar{u} = \begin{pmatrix} \Delta gp_{in}^{(1)} \\ \Delta V_{in}^{(1)} \\ \Delta gp_{in}^{(2)} \end{pmatrix} \bar{y} = \begin{pmatrix} \Delta h_2^{(1)} \\ \Delta \sigma_1 \\ \Delta h_2^{(2)} \end{pmatrix} \bar{d} = \begin{pmatrix} \Delta h_1^{(1)} \\ \Delta T^1 \\ \Delta h_1^{(2)} \\ \Delta T^2 \end{pmatrix} \quad (4.41)$$

Para este caso, será levado em conta as restrições mostradas abaixo:

$$\begin{aligned} \Delta u_{imin} &\leq \Delta \hat{u}_i(t+j) \leq \Delta u_{imax} \quad j = 0, \dots, N_u - 1, \\ u_{imin} &\leq \hat{u}_i(t+j) \leq u_{imax} \quad j = 0, \dots, N_u - 1 \\ y_{imin} &\leq \hat{y}_i(t+j) \leq y_{imax} \quad j = 1, \dots, N_2 \end{aligned} \quad (4.42)$$

onde  $\Delta u_{imin} = -0,1$  ,  $\Delta u_{imax} = 0,1$   $u_{imin} = -0,5$ ,  $u_{imax} = 0,5$ ,  $y_{imin} = -0,1$ ,  $y_{imax} = 0,1$

Usando os resultados apresentados do Capítulo 3 para o GPC Multivariável, pode-se representar a previsão da saída pela seguinte equação:

$$\hat{Y} = \bar{G} \Delta \hat{u} + \bar{F} \quad (4.43)$$

onde:  $\bar{G}$  ( $3N_2 \times 3N_u$ ) e  $\bar{F}$  ( $3N_2 \times 1$ ). Define-se  $\bar{I}_{j,j}$  como uma matriz identidade e  $\bar{O}_{j,j}$  uma matriz nula, desta forma, tem-se as matrizes de peso definidas como:

$$\begin{aligned} \bar{\delta} &= \begin{bmatrix} \delta_1 \bar{I}_{N_2, N_2} & \bar{O}_{N_2, N_2} & \bar{O}_{N_2, N_2} \\ \bar{O}_{N_2, N_2} & \delta_2 \bar{I}_{N_2, N_2} & \bar{O}_{N_2, N_2} \\ \bar{O}_{N_2, N_2} & \bar{O}_{N_2, N_2} & \delta_3 \bar{I}_{N_2, N_2} \end{bmatrix} \\ \bar{\lambda} &= \begin{bmatrix} \lambda_1 \bar{I}_{N_u, N_u} & \bar{O}_{N_u, N_u} & \bar{O}_{N_u, N_u} \\ \bar{O}_{N_u, N_u} & \lambda_2 \bar{I}_{N_u, N_u} & \bar{O}_{N_u, N_u} \\ \bar{O}_{N_u, N_u} & \bar{O}_{N_u, N_u} & \lambda_3 \bar{I}_{N_u, N_u} \end{bmatrix} \end{aligned} \quad (4.44)$$

Define-se também o vetor  $\bar{\Gamma}_{j,1}$ , e a matriz triangular inferior  $\bar{T}_{j,j}$ :

$$\bar{\Gamma}_{j,1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad \bar{T}_{j,j} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \quad (4.45)$$

Considerando-se as matrizes  $\bar{\mathcal{T}}_{3j,3j}$  e  $\bar{\Gamma}_{3j,3}$  definidos por:

$$\bar{\mathcal{T}}_{3j,3j} := \begin{bmatrix} \bar{T}_{j,j} & \bar{O}_{j,j} & \bar{O}_{j,j} \\ \bar{O}_{j,j} & \bar{T}_{j,j} & \bar{O}_{j,j} \\ \bar{O}_{j,j} & \bar{O}_{j,j} & \bar{T}_{j,j} \end{bmatrix} \quad \bar{\Gamma}_{3j,3} := \begin{bmatrix} \bar{\Gamma}_{j,1} & \bar{O}_{j,1} & \bar{O}_{j,1} \\ \bar{O}_{j,1} & \bar{\Gamma}_{j,1} & \bar{O}_{j,1} \\ \bar{O}_{j,1} & \bar{O}_{j,1} & \bar{\Gamma}_{j,1} \end{bmatrix} \quad (4.46)$$

pode se trabalhar com restrições conforme (4.42) dentro de uma formulação matricial:

$$\begin{bmatrix} \bar{I}_{3N_u,3N_u} \\ -\bar{I}_{3N_u,3N_u} \\ \bar{\mathcal{T}}_{3N_u,3N_u} \\ -\bar{\mathcal{T}}_{3N_u,3N_u} \\ \bar{G}_{3N_2,3N_u} \\ -\bar{G}_{3N_2,3N_u} \end{bmatrix} \Delta \hat{u} \leq \begin{bmatrix} \bar{\Gamma}_{3N_u,3} \Delta \bar{u}_{max} \\ -\bar{\Gamma}_{3N_u,3} \Delta \bar{u}_{min} \\ \bar{\Gamma}_{3N_u,3} (\bar{u}_{max} - \bar{u}(t-1)) \\ \bar{\Gamma}_{3N_u,3} (\bar{u}(t-1) - \bar{u}_{min}) \\ \bar{\Gamma}_{3N_2,3} \bar{y}_{max} - \bar{F}_{2N_2,1} \\ \bar{F}_{3N_2,1} - \bar{\Gamma}_{3N_2,3} \bar{y}_{min} \end{bmatrix} \quad (4.47)$$

A solução do problema GPC é calcular um vetor  $\Delta \hat{U}$  que minimize a função custo quadrática considerando as restrições. O desempenho da resposta do processo, depende do ajuste dos parâmetros definido nas matrizes  $\bar{\delta}$  e  $\bar{\lambda}$ . A escolha destes parâmetros será feito através de AG.

Para validar esta técnica de controle, foram coletados dados reais de laminação usado na planta da SIDERAR S.A. Argentina. Será feita uma análise regulatória considerando os distúrbios, e uma análise de controle servo não se considerando os distúrbios, uma vez que serão dados diferentes degraus em diferentes instantes para mostrar a capacidade desta técnica para este tipo de processo.

### Caso Regulatório

A Figura 4.11 mostra a variação da espessura de entrada medida e a variação da temperatura nas duas cadeiras de laminação que são usados no modelo proposto.

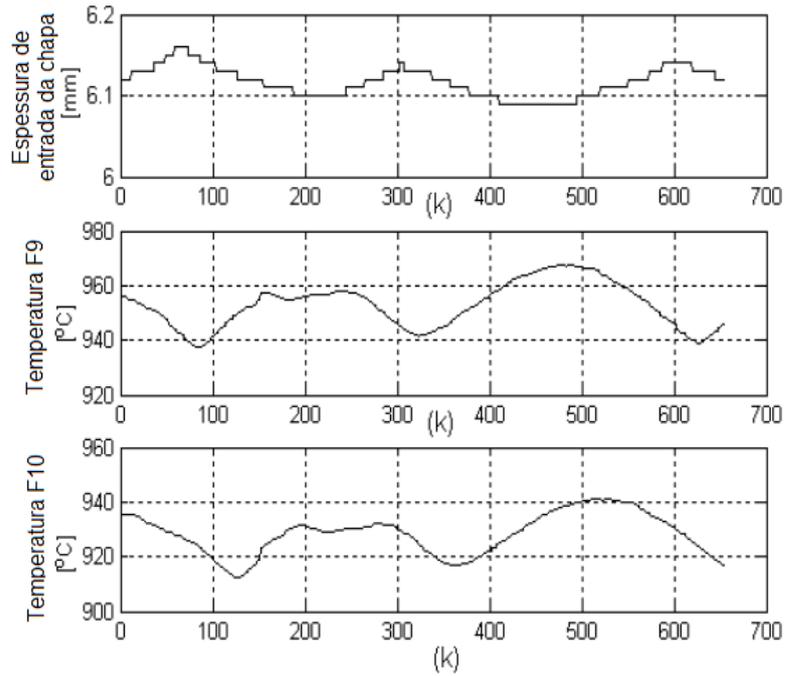


Figura 4.11: Variação da espessura de entrada e temperatura dos *Stands* F9 e F10.

Os parâmetros ajustados por AG foram  $N_u = 1$ ,  $N_2 = 5$ ,  $\alpha = 0,9559$ ,  $\delta_1 = 54,8996$ ,  $\delta_2 = 94,1348$ ,  $\delta_3 = 60,9715$ ,  $\lambda_1 = 73,5311$ ,  $\lambda_2 = 0,4712$  e  $\lambda_3 = 54,5646$ . Percebe-se que a utilização da matriz de  $\bar{\lambda}$ , um parâmetro ponderando cada ação de controle  $\Delta u_i$ , e da matriz  $\bar{\delta}$ , um parâmetro ponderando cada saída prevista  $\hat{y}_i$ , possibilitou o controle do processo de maneira eficaz, pois quando foram feitos testes utilizando apenas um parâmetro  $\delta$  e um  $\lambda$ , o sistema não se estabilizava.

A Figura 4.12 mostra a variação da espessura de saída para as duas cadeiras, considerando que a referência seja igual a zero. Observa-se que o erro apresentado pelo GPC é de cerca de  $50\mu m$  e com o controle convencional utilizado a variação é de  $100\mu m$ . O mesmo resultado para diferentes dados coletados dentro do processo de laminação foram simulados e o erro apresentado pelo GPC sempre foi menor do que o apresentado pelo FFF.

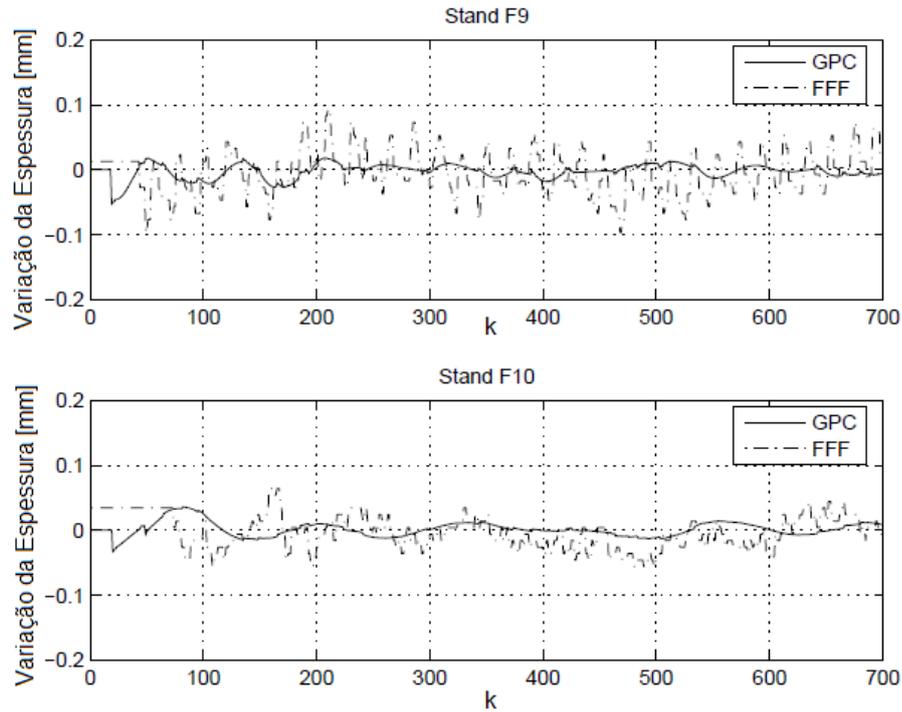


Figura 4.12: Variação da Espessura de saída dos *Stands* F9 e F10 com FFF e GPC.

### Caso Servo

Será mostrado agora o controle multivariável aplicando-se diferentes degraus em diferentes instantes de tempo. Os parâmetros ajustados pelo AG foram  $N_u = 2$ ,  $N_2 = 9$ ,  $\alpha = 0,0641$ ,  $\delta_1 = 72,5775$ ,  $\delta_2 = 2,0536$ ,  $\delta_3 = 89,4942$ ,  $\lambda_1 = 7,1453$ ,  $\lambda_2 = 24,2487$  e  $\lambda_3 = 5,3754$ , e se teve uma boa resposta para as duas saídas do sistema, que pode ser observado na Figura 4.13.

Percebe-se na Figura 4.13, que as saídas não interferem entre si, ou seja, o sistema está desacoplado. Neste caso, assim como no controle regulatório, foi necessário o uso de três parâmetros tanto para  $\delta$  quanto para  $\lambda$ . Em simulações iniciais, percebeu-se que com a sintonia de um parâmetro  $\lambda$  e  $\delta$ , as saídas se apresentam oscilatórias e com elevado *off-set*.

O ISE apresentado pelas espessuras de saídas dos *Stands* F9 e F10 foram respectivamente 0,1738 e 0,4845.

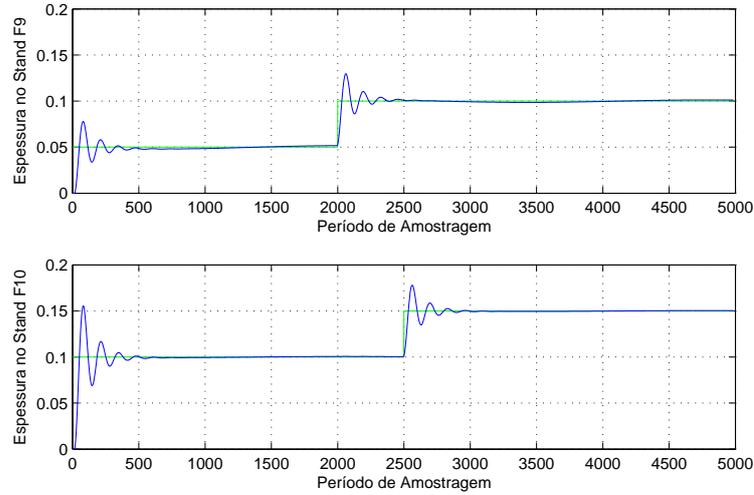


Figura 4.13: Espessura de saída das duas Cadeiras de Laminação.

## 4.5 Conclusão do Capítulo

Neste capítulo, foram apresentadas as técnicas utilizadas para sintonizar os parâmetros do Controlador Preditivo, sendo que a mais utilizada é por tentativa e erro. Embora esta consiga estabilizar alguns processos, principalmente os estáveis em malha aberta e monovariáveis, a sintonia torna-se uma tarefa exaustiva, apresentando resultados não muito satisfatórios.

Em seguida, foi apresentado o algoritmo de sintonia desenvolvido por *Cooper* [34], na qual todo o processo deve ser aproximado por um modelo de primeira ordem com um atraso de transporte, pois o algoritmo necessita da constante de tempo do modelo e do atraso de transporte. Esta técnica é utilizada somente para sistemas estáveis em malha aberta, mono e multivariáveis. Foi mostrado também que os parâmetros de sintonia usando este método apresentam valores elevados, sendo que em casos nos quais as restrições são consideradas, as mesmas não são respeitadas, indicando que a solução obtida pelo controlador preditivo é não factível.

Foi descrita depois a técnica utilizando Algoritmo Genético para a sintonia dos parâmetros do controlador preditivo, que pode ser utilizada para qualquer tipo de modelo de processo. Foi mostrado através de simulações em diferentes modelos, que a sintonia utilizando Algoritmo Genético apresentou parâmetros bem menores

do que o guia proposto por [34], além de ter um índice de desempenho inferior em todas as situações. É importante mencionar também que os processos sintonizados por AG, respeitaram todas as restrições em todos os modelos testados, exceto em um por não se ter feito um estudo detalhado à respeito de suas restrições, o que não era o objetivo desta parte.

Por fim, a sintonia via AG foi utilizada no Controle Preditivo do Processo do Laminador linear, que é multivariável, instável em malha aberta e com distúrbio. Neste processo de laminação, comparou-se a técnica FFF utilizada na indústria SIDERAR da Argentina e o GPC sintonizado por AG, com o objetivo de evidenciar a sintonia com AG. Foram feitas duas análises: uma utilizando o caso regulatório, nos quais os distúrbios de temperatura e de espessura de entrada foram levados em consideração; e uma outra utilizando o caso servo, no qual foram aplicados degraus em diferentes instantes de tempo para se avaliar o acoplamento entre as saídas do processo, bem como a eficácia do GPC para o desacoplamento das malhas.

É importante mencionar o fato de se ter utilizado três parâmetros diferentes para  $\delta$  e  $\lambda$ , pois em casos de testes utilizando apenas um valor em cada parâmetro, o processo ficava instável e oscilatório.

Através dos gráficos das saídas, do caso regulatório, percebe-se que a variação da espessura da chapa no controle realizado pelo GPC, foi menor do que a apresentada pelo FFF.

## Capítulo 5

# MPC Não-Linear Com Modelos Hammerstein e Neural

Quando o processo não-linear atua numa faixa de operação ampla ou a não-linearidade do processo é significativa o bastante para tornar o desempenho do controlador inadequado para atender os requisitos estabelecidos, a utilização de um modelo não-linear deve ser considerada [95].

Nos últimos anos houve um grande crescimento nas aplicações industriais do controle preditivo não-linear NMPC (*Nonlinear Model Predictive Control*) que se apresenta como uma estratégia de controle promissora para diversas áreas da engenharia. A Tabela 5.1 ilustra algumas aplicações industriais conforme mostrado em [93].

Tabela 5.1: Aplicações Comerciais de NMPC.

| Area         | Adersa    | Aspen<br>Technology | Continental<br>Controls | DOT<br>Products | Pavilion<br>Techno | Total     |
|--------------|-----------|---------------------|-------------------------|-----------------|--------------------|-----------|
| ar e gás     | -         | -                   | 18                      | -               | -                  | 18        |
| química      | 02        | -                   | 15                      | -               | 05                 | 22        |
| alimentos    | -         | -                   | -                       | -               | 09                 | 09        |
| polimeros    | -         | 01                  | -                       | 05              | 15                 | 21        |
| papel        | -         | -                   | -                       | -               | 01                 | 01        |
| refinaria    | -         | -                   | -                       | -               | 13                 | 13        |
| utilidades   | -         | 05                  | 02                      | -               | -                  | 07        |
| outras       | 01        | -                   | 01                      | -               | -                  | 02        |
| <b>Total</b> | <b>03</b> | <b>06</b>           | <b>36</b>               | <b>05</b>       | <b>43</b>          | <b>93</b> |

Os principais motivos deste crescimento são o fraco desempenho de contro-

ladores lineares em processos fortemente não-lineares ou em plantas que trabalham numa ampla faixa de operação, a evolução de estratégias de controle baseadas em modelos não-lineares e o desenvolvimento de processadores poderosos o bastante para tornarem possíveis estas implementações.

## 5.1 GPC com Perturbações Mensuráveis para o Modelo Hammerstein (HGPC)

Dentre as aplicações de controle preditivo em plantas não-lineares, o emprego de controladores preditivos baseados no modelo de Hammerstein tem motivado uma série de aplicações bem sucedidas ao longo dos últimos anos ([7]; [66]; [49]; [16]; [64]; [92]; [124]). Isto se deve ao fato de o projeto do controlador preditivo não-linear ser mais simples, possibilitando inclusive uma solução analítica para o problema de minimização da função custo para o caso sem restrições.

Desenvolvido por [106], o controlador HGPC utiliza um modelo de Hammerstein cuja parcela linear é representada por um modelo CARIMA, apresentado no Capítulo 2 através da expressão (2.4) e cuja não linearidade é representada pelo polinômio (2.1).

A lei de controle HGPC é obtida pela minimização do seguinte critério:

$$J_{HGPC} = \sum_{j=N_1}^{N_2} \delta [\hat{y}(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda \Delta x^2(t+j-1) \quad (5.1)$$

na qual  $\delta$  e  $\lambda$  é a ponderação do sinal de erro e de controle respectivamente,  $N_1$  e  $N_2$  são os horizontes de predição da saída inicial e final, respectivamente, e  $N_u$  é o horizonte de controle. Os termos  $\hat{y}(t+j)$  e  $w(t+j)$  representam o sinal da saída e o sinal de referência a  $j$  passos a frente e,  $\Delta x(t+j-1)$  é o incremento do sinal da pseudo-saída  $x$  no instante  $t+j-1$ , ou seja:

$$\Delta x(t) = x(t) - x(t-1) \quad (5.2)$$

A solução do problema em questão, ou seja, o valor de  $\Delta x(t)$  que minimiza  $J_{HGPC}$  é semelhante à apresentada no Capítulo 3, seção 3.2.

A partir do cálculo das raízes do polinômio representado pela Equação (2.1) determina-se a ação de controle que, de fato é aplicada ao processo. O sinal de controle deve ser selecionado dentre as raízes válidas, podendo surgir mais de uma possibilidade conforme o grau da não-linearidade ( $m$ ).

### Multiplicidade de Soluções para a Lei de Controle

A estratégia de controle preditivo HGPC resulta em multiplicidade de soluções para o problema de controle. Isto ocorre porque o controlador encontra o valor ótimo para a pseudo-saída  $x(t)$  a qual pode fornecer diversas soluções  $u(t)$  que satisfazem (2.1). Este problema pode ser resolvido através de uma segunda operação de otimização, por exemplo, através de um método iterativo de busca ([122]; [62]), ou, ainda, empregando algum tipo de aproximação mostrado em [106].

Nesta Tese será usado a Aproximação por Série de Taylor para determinar o sinal de controle.

### Aproximação por Série de Taylor

Considerando que  $f = u^m$ , a aplicação de uma linearização em torno de um ponto  $u_0$  leva a:

$$\bar{f} = f_0 + \left. \frac{\partial f}{\partial u} \right|_{u=u_0} (u - u_0) = u_0^m + m u_0^{m-1} (u - u_0) \quad (5.3)$$

e considerando que o ponto  $u_0 = u(t-1)$ ,

$$\bar{f} = u^m(t-1) + m u^{m-1}(t-1) [u(t) - u(t-1)] = m u^{m-1}(t-1) u(t) - (m-1) u^m(t-1) \quad (5.4)$$

e, desta forma, a não linearidade do sistema

$$x(t) = \gamma_1 u(t) + \gamma_2 u^2(t) + \dots + \gamma_m u^m(t) = \sum_{i=1}^m \gamma_i u^i(t) \quad (5.5)$$

pode ser aproximada por

$$x(t) = \gamma_1 u(t) + \gamma_2 \bar{u}^2(t) + \dots + \gamma_m \bar{u}^m(t) = \gamma_1 u(t) + \sum_{i=2}^m \gamma_i \bar{u}^i(t) \quad (5.6)$$

ou, ainda por,

$$x(t) = \left[ \sum_{i=1}^m i\gamma_i u^{i-1}(t-1) \right] u(t) - \sum_{i=2}^m (i-1)\gamma_i u^i(t-1) \quad (5.7)$$

O sinal de controle torna-se único e determinado pela equação

$$u(t) = \frac{x(t) + \sum_{i=2}^m (i-1)\gamma_i u^i(t-1)}{\left[ \sum_{i=1}^m i\gamma_i u^{i-1}(t-1) \right]} \quad (5.8)$$

Mesmo quando se utilizam aproximações para evitar a aplicação de um método numérico, pode ocorrer que o sinal de controle calculado não atenda às restrições do sistema e, neste caso, é necessário definir um valor de controle a ser aplicado que pode ser o sinal aplicado no instante anterior,  $u(t-1)$ , ou mesmo o valor da entrada em regime permanente, conforme o conhecimento prévio do processo.

## 5.2 Controle Preditivo com Redes Neurais

Nos anos recentes, a habilidade das redes neurais em modelar uma função não-linear para um grau de precisão tem atraído grande atenção da comunidade de controle de processos. *Bhat e McAvoy* [9] estavam entre os primeiros a usar as RN para modelar processos químicos não-linear. *Cybenko* [26] e *Narendra e Parthasarathy* [86] propuseram o uso de redes neurais em identificação e/ou estratégias de controle não-linear de processos.

Neste seção, será mostrado como as redes neurais são aplicadas na estratégia de controle preditivo.

### A Equação Geral da Saída Prevista a 1 passo a frente

A Figura 5.1 representa a estrutura geral da técnica NPC (Controle Preditivo Neural), com a predição a 1 passo a frente.

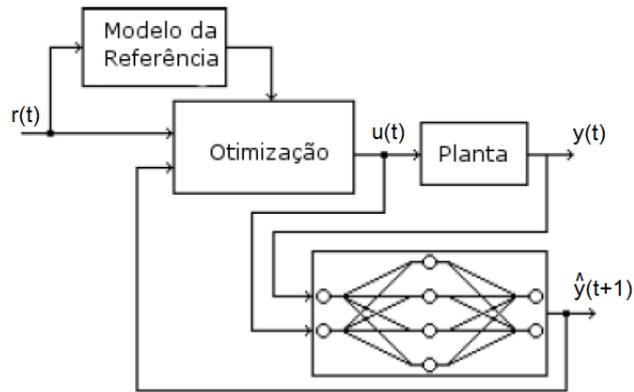


Figura 5.1: Diagrama do Controle Preditivo Neural.

Redes Neurais *feedforward*, com uma camada oculta usando função de ativação linear ou sigmóide, são consideradas estimadores universais [38], isto prova que uma rede neural com somente uma camada oculta é capaz de representar qualquer função de  $R^n \rightarrow R^m$ , sendo limitada apenas pelo número de neurônio na camada oculta.

Considere um modelo não-linear NNARX, cuja saída prevista 1 passo a frente seja:

$$\hat{y}(t+1) = f[y(t), y(t-1), \dots, y(t-n), u(t), u(t-1), \dots, u(t-m)] \quad (5.9)$$

onde  $f$  é um função não-linear da saída do sistema  $[y(t), y(t-1), \dots, y(t-n)]$  e valores da entrada  $[u(t), u(t-1), \dots, u(t-m)]$ . A variável  $n$  é o número de saídas passadas e  $m$  é o número de entradas passadas.

Para representar a dinâmica do sistema, é necessário introduzir o efeito do *feedback* dentro de redes neurais, caracterizado pela aplicação de uma Rede Neural Recursiva.

Assim, como no modelo ARX, os sinais de entrada da rede são associados com os valores de entradas e saída passadas. Esta estrutura caracteriza especificamente uma TDNN (*Time Delay Neural Networks*). Uma rede com três camadas será a base deste estudo, conforme pode-se ver na Figura 5.2, onde  $N$  representa número de neurônios na camada oculta,  $b_i$  são as bias da camada oculta e o bloco  $S$

representa função sigmoide não-linear que será aplicada na saída de cada neurônio da camada oculta,  $b_s$  representa o bias da camada de saída. As matrizes de pesos  $w_1$  e  $w_2$  tem dimensões  $N \times (n + m)$  e  $1 \times N$ , respectivamente, e representam os pesos das conexões entre as camadas 1 – 2 e 2 – 3, respectivamente.

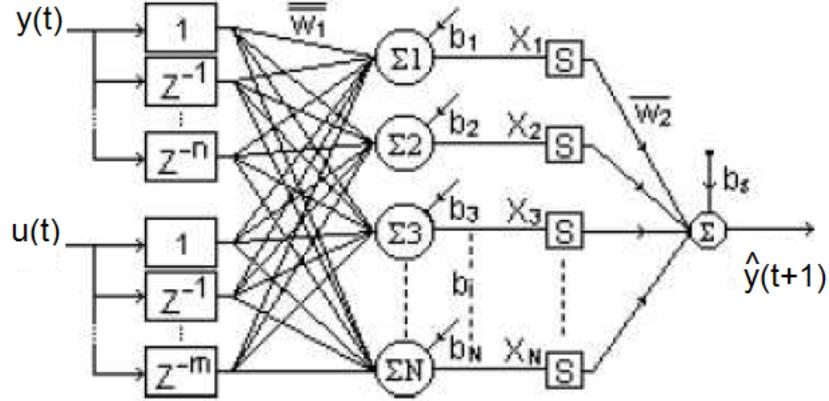


Figura 5.2: A Estrutura Usada em uma TDNN.

A expressão geral do modelo TDNN proposto é:

$$\hat{y}(t + 1) = b_s + \sum_{i=1}^N w_2(1, i)S(X_i) \quad (5.10)$$

onde:

$$X_i = b(i, 1) + \sum_{j=1}^n w_1(i, j)y(t - j + 1) + \sum_{j=1}^m w_1(i, n + j)u(t - j + 1) \quad (5.11)$$

e  $\hat{y}(t + 1)$  representa a saída estimada, que é uma função não-linear das entradas e saídas do sistema.

### Equações Diferenciais do Sistema Neural para Predição 1 passo a frente

Em geral, a aplicação da lei de controle requer o cálculo da expressão diferencial da saída em relação a entrada do processo. Baseado na Equação (5.10) e diferenciando-a em relação a entrada, pode-se generalizar a expressão:

$$\frac{\partial \hat{y}(t+1)}{\partial u(t)} = \frac{\partial}{\partial u(t)} \left[ b_s + \sum_{i=1}^N w_2(1, i) S(X_i) \right] \quad (5.12)$$

e pode ser reescrita como:

$$\frac{\partial \hat{y}(t+1)}{\partial u(t)} = \sum_{i=1}^N w_2(1, i) S'(X_i) \frac{\partial X_i}{\partial u(t)} \quad (5.13)$$

$$S' = \frac{dS}{dX_i} \quad (5.14)$$

Então, a Equação (5.13), representa a expressão diferencial de primeira ordem e  $S'$  pode ser expandida tendo como base (5.11):

$$\frac{\partial X_i}{\partial u(t)} = \frac{\partial}{\partial u(t)} b(i, 1) + \frac{\partial}{\partial u(t)} \left\{ \sum_{j=1}^n w_1(i, j) y(t-j+1) \right\} + \frac{\partial}{\partial u(t)} \left\{ \sum_{j=1}^m w_1(i, n+j) u(t-j+1) \right\} \quad (5.15)$$

Note que os termos  $y(t-1)$ ,  $y(t-2)$ , ...,  $y(t-n)$ , assim como os termos  $u(t-1)$ ,  $u(t-2)$ , ...,  $u(t-m)$ , são valores passados, portanto, não dependem de  $u(t)$ . Desta maneira, o somatório é sempre nulo, exceto para  $j=1$ . O termo  $u(t)$  diferenciado é igual a 1, assim:

$$\frac{\partial X_i}{\partial u(t)} = w_1(i, n+1) \quad (5.16)$$

Substituindo em (5.13):

$$\frac{\partial \hat{y}(t+1)}{\partial u(t)} = \sum_{i=1}^N w_2(1, i) S'(X_i) w_1(1, n+1) \quad (5.17)$$

onde (5.17) representa a equação diferencial generalizada de uma TDNN.

## 5.2.1 MPC Neural Com Regra Baseado no Gradiente (PNRG)

### Predição 1 Passo a Frente

O controle preditivo, por definição, calcula a ação de controle baseado nos valores previsto do sistema. Então, uma rede neural bem treinada, pode representar a dinâmica do modelo não-linear. Em seguida, o algoritmo de controle pode ser implementado baseado no erro entre o sinal de referência e o valor previsto da saída.

Na Figura 5.3, observe que a ação de controle é baseada nos valores previsto e não na saída real. O modelo considerando uma TDNN bem treinada é capaz de representar a dinâmica da planta.

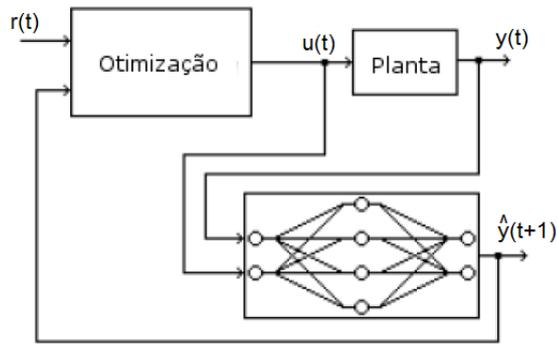


Figura 5.3: Diagrama de Bloco do Controle Preditivo Neural.

$t$  = tempo de amostragem;

$r(t)$  = sinal de referência;

$u(t)$  = ação de controle;

$y(t)$  = saída da planta;

$e(t)$  = erro entre  $r(t)$  e a saída da planta;

$\hat{y}(t + 1)$  = saída prevista da planta.

Por definição, o controle preditivo busca minimizar um índice de desempenho,  $J$  considerando aspectos como: erro, esforço de controle, etc. Nesta aplicação, o índice  $J$ , será considerado como:

$$J = \frac{1}{2}e^2(t+1) \quad (5.18)$$

$$e(t+1) = r(t+1) - \hat{y}(t+1) \quad (5.19)$$

A regra que é usada para atualizar a ação de controle, é baseada no método do gradiente:

$$u^i(t) = u^{i-1}(t) - \mu \frac{\partial J}{\partial u(t)} \quad (5.20)$$

onde:  $\mu$  especifica o tamanho do passo e  $\frac{\partial J}{\partial u(t)}$  a direção de busca.

ou seja, a ação de controle é atualizada na direção contrária do gradiente  $\frac{\partial J}{\partial u(t)}$ , caminhando para o mínimo  $J$  com passo  $\mu$ .

Através de (5.18) e (5.19) conclui-se que:

$$\frac{\partial J}{\partial u(t)} = \frac{1}{2}[-2e(t+1)\frac{\partial \hat{y}(t+1)}{\partial u(t)}] = -e(t+1)\frac{\partial \hat{y}(t+1)}{\partial u(t)} \quad (5.21)$$

De acordo com a TDNN proposta, a expressão diferencial já é conhecida. Então usando (5.17):

$$\frac{\partial J}{\partial u(t)} = -e(t+1) \sum_{i=1}^N w_2(1, i) S'(Xi) w_1(i, n+1) \quad (5.22)$$

Com os resultados obtidos, voltando na equação (5.20), a lei de controle pode ser escrita como:

$$u(t+1) = u(t) + \mu e(t+1) \left[ \sum_{i=1}^N w_2(1, i) S'(Xi) w_1(i, n+1) \right] \quad (5.23)$$

O algoritmo de controle é implementado pelos seguintes passos:

1. selecione  $\mu$ ;
2. Use (5.10) e (5.11) para calcular  $\hat{y}(t+1)$ ;
3. Use (5.19) para calcular  $e(t)$ ;

4. Use (5.23) para calcular o novo sinal de controle;
5. Aplicar o novo sinal de controle na entrada do sistema;
6. Retorne ao passo 2.

### Predição $N_2$ Passos a Frente

O algoritmo anterior, pode ser utilizado como base para fazer predições  $N_2$  passos a frente, bastando para isto, que os valores escalares possam ser agora representados por vetores de predição, porém, será mostrado agora um caso mais geral, no qual a função custo  $J$  será composta tanto pelo erro, quanto pelo esforço de controle. Desta forma, tem-se:

$$J(u) = \delta[W - \hat{Y}]^T [W - \hat{Y}] + \lambda \Delta U^T \Delta U = \delta E^T E + \lambda \Delta U^T \Delta U \quad (5.24)$$

no qual

$$W = [w(t+d), w(t+d+1), \dots, w(t+d+N_2)]^T \quad (5.25)$$

$$\hat{Y} = [\hat{y}(t+d), \hat{y}(t+d+1), \dots, \hat{y}(t+d+N_2)]^T \quad (5.26)$$

$$U = [u(t), u(t+1), \dots, u(t+N_u-1)]^T \quad (5.27)$$

$$\Delta U = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u-1)]^T \quad (5.28)$$

$$E = [e(t+1), e(t+2), \dots, e(t+N_2)] \quad (5.29)$$

A saída predita  $k$  passos a frente pode ser escrita como:

$$\begin{aligned} \hat{y}(t+k) &= f[\hat{y}(t+k-1), \dots, \hat{y}(t+k-\min[k, n]), y(t-1), \\ &\dots, y(t-\max[n-k, 0]), u(t+k-d), \dots, u(t+k-d-m)]. \end{aligned} \quad (5.30)$$

sendo:  $n$  o número de regressores da saída,  $m$  o número de regressores da entrada,  $f$  uma rede perceptron de duas camadas.

A minimização da função  $J$  quando a predição é uma função não-linear, torna o problema de otimização complexo. Existem na literatura vários métodos para a resolução deste problema. Neste trabalho, foi utilizado o método *Broyden-Fletcher-Goldfarb-Shanno*(BFGS) apresentado em [32], pois apresenta rápida convergência podendo ser aplicado em tempo real, o que não acontece nos outros métodos mostrados em [87].

$$U^{(i+1)}(t) = U^{(i)}(t) - \mu \frac{\partial J}{\partial U} \quad (5.31)$$

onde:  $\mu$  especifica o tamanho do passo e  $\frac{\partial J}{\partial U}$  a direção de busca.

Usando a Equação (5.31), calcula-se o valor do passo  $\mu$  e a direção de busca, para achar o valor do sinal de controle a ser aplicado no processo. O algoritmo que calcula o tamanho do passo  $\mu$  pode ser visto em [87]. Para o cálculo da direção de busca, será usado o método BFGS mostrado em [87] sendo:

$$\frac{\partial J}{\partial U} = -B^{(i)}(t) * Grad[U^{(i)}(t)] \quad (5.32)$$

tal que:  $B^{(i)}(t)$  especifica uma aproximação da inversa da Hessiana de  $U$ , a qual é denominada pela letra  $H$  e Grad é o gradiente do vetor  $U$ .

Dessa forma, calculam-se as derivadas parciais mostrada acima para que se possa determinar a direção de busca e em seguida determinar o vetor de controle.

A derivada parcial  $\frac{\partial \Delta U}{\partial U}$ , desde que  $\Delta u(t) = u(t) - u(t - 1)$  é:

$$\frac{\partial \Delta U}{\partial U} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}_{N_u \times N_u} \quad (5.33)$$

A derivada parcial  $\frac{\partial \hat{y}}{\partial U}$  é:

$$\frac{\partial \hat{Y}}{\partial U} = \begin{bmatrix} \frac{\partial \hat{y}(t+N_1)}{\partial u(t)} & \cdots & \frac{\partial \hat{y}(t+N_1)}{\partial u(t+N_u-1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}(t+N_2)}{\partial u(t)} & \cdots & \frac{\partial \hat{y}(t+N_2)}{\partial u(t+N_u-1)} \end{bmatrix}_{N_u \times (N_2 - N_1 + 1)} \quad (5.34)$$

O algoritmo de controle do NPC pode ser implementado pelos seguintes passos:

1. Inicializar uma sequência de controle futuros  $U^0$ , determinar  $grad[U^0]$ ,  $B^0 = I$  e  $i = 0$ ;
2. Selecionar o passo  $\mu^i$  conforme [87];
3. Obter  $\hat{y}$ ,  $\Delta U^i$ ,  $E$  definidos em (5.26) e (5.28) (5.29);
4. Calcular o gradiente e a aproximação da inversa da Hessiana utilizando o método BFGS mostrado em [87];
5. Calcular o novo sinal de controle,  $U^{i+1}(t) = U^i(t) - \mu B^i \cdot Grad(U^i)$  conforme (5.31);
6. Verificar se  $|U^{i+1} - U^i| < \eta$  ou  $i > i_{max}$ . Caso alguma das duas condições for verdadeira, aplicar o novo sinal de controle  $u^i(t)$  na planta, fazer  $t = t + 1$ ,  $i = 0$  e retornar para o passo 1. Caso contrário, faça  $i = i + 1$  e retorne ao passo 2.

### 5.2.2 Controle Preditivo Neural com Linearização Instantânea (PNLI)

A aplicação do princípio da linearização instantânea em controle preditivo, gera uma ampla vantagem ao ser comparada com a técnica de controle preditivo não-linear mostrada acima. Quando o modelo é linear, tem-se apenas um mínimo que é encontrado diretamente. Assim, usando um modelo linearizado, fica-se livre de vários problemas que são apresentados nas otimizações não-lineares.

No princípio da linearização instantânea, obtém-se modelos lineares que serão válidos apenas próximo do seu ponto de operação.

## Linearização Instantânea

A ideia da linearização instantânea é extrair o modelo linear de uma rede neural não-linear em cada período de amostragem. Este princípio será demonstrado para modelos determinísticos. Mais detalhes pode ser encontrado em [87].

Assume-se que o modelo de entrada e saída de uma rede neural determinística do sistema a ser controlado é avaliado como:

$$y(t) = g[\varphi(t)] \quad (5.35)$$

e o vetor de regressão é dados por:

$$\varphi(t) = [y(t-1), \dots, y(t-n), u(t-d), \dots, u(t-d-m)] \quad (5.36)$$

O princípio da linearização instantânea é mostrado a seguir:

- Interpretar o vetor de regressão como um vetor definindo os estados do sistema e o tempo  $t = \tau$ . Linearize  $f$  em torno dos estados corrente  $\varphi(t)$  para obter o modelo aproximado.

$$\tilde{y}(t) = -a_1 \tilde{y}(t-1) - \dots - a_n \tilde{y}(t-n) + b_0 \tilde{u}(t-d) + \dots + b_m \tilde{u}(t-d-m) \quad (5.37)$$

onde:

$$a_i = -\frac{\partial f[\varphi(t)]}{\partial y(t-i)} \Big|_{\varphi(t)=\varphi(\tau)} \quad b_i = -\frac{\partial f[\varphi(t)]}{\partial u(t-d-i)} \Big|_{\varphi(t)=\varphi(\tau)} \quad (5.38)$$

e:

$$\tilde{y}(t-i) = y(t-i) - y(\tau-i) \quad \tilde{u}(t-i) = u(t-1) - u(\tau-1) \quad (5.39)$$

- Separar a parte da expressão que contém os componentes do vetor de regressão corrente,  $\varphi(t)$ , o modelo aproximado pode ser representado como:

$$y(t) = [1 - A(q^{-1})]y(t) + q^{-d}B(q^{-1})u(t) + \varsigma(\tau) \quad (5.40)$$

sendo o termo  $\zeta(\tau)$  dado por:

$$\zeta(\tau) = y(\tau) + a_1 y(\tau-1) + \dots + a_n y(\tau-n) - b_0 u(\tau-d) - \dots - b_m u(\tau-d-m) \quad (5.41)$$

Os coeficientes  $a_i$  e  $b_i$  são os coeficientes dos polinômios  $A(q^{-1})$  e  $B(q^{-1})$  que será aplicado no modelo CARIMA.

- Realizar o controle GPC, cujo algoritmo foi apresentado no Capítulo 3.

### 5.3 MPC Não-Linear do Nível do Molde do Lingotamento Contínuo

O controle do nível de aço no molde é uma das malhas mais importantes em processo de lingotamento contínuo, influenciando diretamente na qualidade do aço produzido e conseqüentemente em seu valor comercial. O sistema utilizado atualmente na usina da Arcelor Mittal Tubarão é constituído de três partes principais ilustradas na Figura 5.4, a saber; unidade de medição, unidade de controle, unidade de atuação hidráulica.

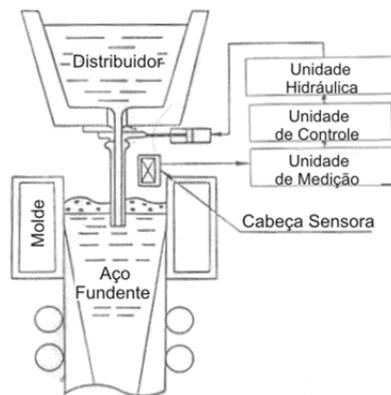


Figura 5.4: Configuração Básica do Sistema de Controle.

O funcionamento consiste na colocação de um sensor acima da superfície do aço líquido recoberto por pó fluxante. O sinal elétrico captado pelo sensor é tratado na unidade amplificadora, que envia o sinal para a unidade de controle. Nesta última unidade estão contidas as rotinas de controle e alarmes, através de

um algoritmo específico para controle do nível. A unidade de controle avalia o comportamento do nível medido e envia as devidas correções ao sistema de atuação hidráulica, que posiciona um cilindro acoplado a uma válvula gaveta, utilizada para regular a vazão de aço do distribuidor para o molde. Com a posição da válvula gaveta, mais precisamente, sua placa central, estará definida a vazão de entrada de aço no molde, vindo do distribuidor, que em uma situação de regime deve contrapor ao volume de aço retirado do molde para continuidade do processo e formação das placas na saída da máquina.

No processo real existem diversos fatores externos que alteram o funcionamento da malha de controle, dentre distúrbios e comportamento do aço, relacionados aos diferentes materiais produzidos pela máquina. Nesta seção propõe-se estratégias de controle para reduzir os efeitos destes distúrbios através das técnicas de controle preditivo não linear apresentadas no início deste capítulo, ou seja, o controle preditivo neural com a regra do gradiente (PNRG), o controle preditivo neural com linearização instantânea (PNLI) e o controle preditivo usando o modelo Hammerstein (HGPC).

### 5.3.1 Equacionamento Básico do Nível do Molde

A Válvula gaveta é um dispositivo formado por três placas sobrepostas que possuem um orifício idêntico em cada uma destas placas, onde a placa central é provida de movimento horizontal ( $X_{sg}$ ), fazendo com que a área de passagem de aço, consista na área de intersecção dos orifícios, conforme é ilustrado pela Figura 5.5. Obtém-se a área efetiva ( $A_s$ ) de passagem de aço, através de um cálculo considerando a intersecção de orifícios concêntricos de raio  $R$ . De acordo com [120] a área efetiva é dada por:

$$A_s = 2 \left[ R^2 \cos^{-1} \left( \frac{R - \frac{X_{sg}(t)}{2}}{R} \right) - \left( R - \frac{X_{sg}(t)}{2} \right) \sqrt{RX_{sg} - \left( \frac{X_{sg}(t)}{2} \right)^2} \right] \quad (5.42)$$

A válvula Gaveta está localizada logo após a saída distribuidor e o transporte do fluxo do aço da válvula gaveta para o molde é feita através da válvula submersa, conforme mostra a Figura 1.7. Considerando o comprimento da válvula submersa

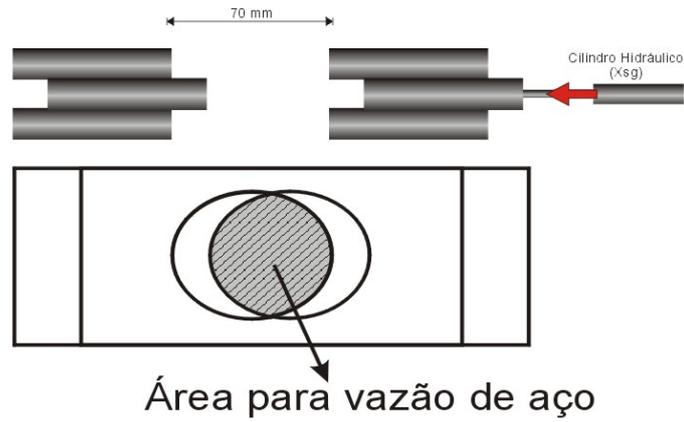


Figura 5.5: Movimento e Formato da Válvula Gaveta do Distribuidor.

de  $L = 600mm$  e a velocidade do fluxo de aço na saída da válvula gaveta de  $V_n = \sqrt{2.g.h}$ , o atraso de transporte será dado por  $T_v = 0,11seg..$  Assim, de acordo com [120] a equação dinâmica do nível do molde, é dada por:

$$\frac{dh(t)}{dt} = \frac{1}{A_m} [A_s(t - T_v) \cdot V_n(t) - A_m \cdot V_{cs}(t)] \quad (5.43)$$

onde:

- $H$  é a altura do nível do molde  $m$ ;
- $A_m$  é a área do molde  $m^2$ ;
- $g$  é a gravidade  $m/s^2$ ;
- $h$  é a altura do fluido entre os níveis de aço do distribuidor e do molde  $m$ ;
- $V_{cs}$  Velocidade de Lingotamento ou extração da placam  $m/s$ ;

A partir de (5.43) chega-se à seguinte função de transferência:

$$H(s) = \frac{1}{A_m s} [A_s(s) \sqrt{2gh} - A_m V_{cs}(s)] \quad (5.44)$$

caracterizando-se, portanto, como uma malha integradora. Para a representação matemática dos demais blocos que compõem a malha de controle, foram realizados

experimentos afim de identificar o modelo conforme mostrado em [105]. A função de transferência (5.45) representa o controlador da malha de posição da válvula gaveta, malha escrava, e o sistema hidráulico responsável por sua movimentação física.

$$G_p(s) = \frac{0.25}{0.5s + 1} \quad (5.45)$$

A função de transferência (5.46) representa o bloco referente ao sensor de nível do molde:

$$G_S(s) = \frac{e^{-0.02s}}{0.4s + 1} \quad (5.46)$$

Desta maneira, verifica-se na Figura 5.6 o sistema não-linear completo simulado no Matlab para representação do processo real de controle do nível do molde.



Figura 5.6: Diagrama de Blocos da Malha Completa de Controle de Nível do Molde.

### 5.3.2 Representação dos Distúrbios de *Bulging*, *Clogging* e de Argônio

Em uma malha real de controle de nível de molde, verifica-se a ocorrência de uma série de efeitos inerentes ou não ao processo metalúrgico de produção de placas que afetam de forma direta e indireta o regime estável do nível de aço na superfície do molde, podendo então ser interpretados como distúrbios a serem controlados/reduzidos pelo sistema de controle. Os principais deles são:

1. Vibrações provocadas pelo movimento oscilatório do molde;

2. Agarramento ou desgastes das placas da válvula gaveta, provocado por solidificações de aço entre as placas, incrustação de alumina;
3. Efeitos provocados pelo tracionamento dos rolos abaixo do molde, *Bulging*;
4. *Clogging*: obstrução parcial da válvula submersa, prejudicando a vazão de aço para o interior do molde;
5. Influência do Argônio na medição de nível;
6. Alterações nas condições de processo e ambiente, como velocidade de lingotamento, tipo de aço lingotado e sua respectiva viscosidade, mudança de largura da placa em processo, etc.;

O aparecimento destes efeitos, ou distúrbios, como são chamados neste texto, e a conseqüente variação demasiada do nível de aço no molde, têm como conseqüência final o surgimento de defeitos nas placas produzidas pela máquina de Lingotamento, além dos riscos de transbordamento de aço para fora do molde e *break-outs*.

A seguir são descritos os efeitos que mais interferem na estabilidade de uma malha de controle de nível de molde, seus comportamentos e as respectivas propostas de representação destes, objetivando retratar com fidelidade o processo a ser controlado.

### **Distúrbio de *Clogging***

O acúmulo progressivo de deposição de alumina ou outros produtos nas paredes da válvula submersa e/ou da própria válvula gaveta é conhecido como efeito *Clogging*. Como esse acúmulo é progressivo, há a redução também progressiva da área efetiva de passagem de aço, fazendo-se necessária a compensação pelo controlador da abertura da válvula gaveta. Em adição a esse efeito, há também o desprendimento abrupto da alumina acumulada (*Unclogging*), liberando o fluxo e provocando a necessidade de ação rápida do controlador para retornar o processo às condições anteriores. Pode-se ver o comportamento dos efeitos "*Clogging*" e "*Unclogging*" na Figura 5.7.

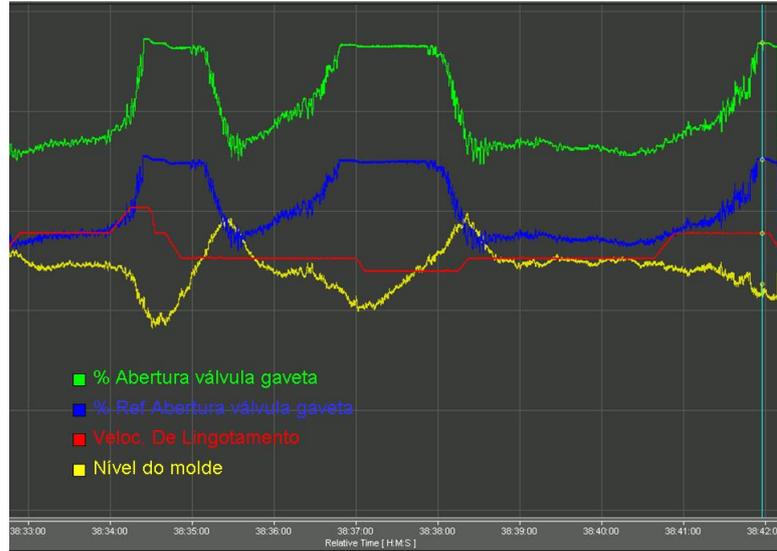


Figura 5.7: Comportamento do efeito *Clogging* na malha de controle.

O comportamento do efeito de *clogging* pode ser perfeitamente simulado em Matlab, incluindo este na malha da válvula gaveta. Observando a posição da válvula gaveta durante o efeito *Clogging*, pode-se representar o movimento de abertura da válvula por uma função aleatória crescente no tempo à medida que aumenta a obstrução da válvula. Após terminar o efeito do entupimento, a válvula atinge a abertura máxima, permanecendo até a ocorrência de *Unclogging* (desentupimento) alguns segundos depois. Assim, a simulação da diminuição da área de passagem do aço pela válvula Gaveta e/ou Submersa, pode ser realizada multiplicando a área da válvula gaveta ( $A_s$ ) sem obstrução pela função de redução da área ( $A_{clg}$ ), a qual é dada pela expressão:

$$A_{clg}(t) = \begin{cases} 1 - \alpha(t - T_1) & , T_1 \leq t < T_2 \\ f & , T_2 \leq t < T_3 \\ f + \beta(t - T_3) & , T_3 \leq t < T_4 \end{cases}$$

onde  $T_1$  é o instante em que inicia o entupimento da válvula gaveta,  $T_2$  é o instante em que ocorre o entupimento máximo,  $T_3$  e  $T_4$  são os instantes em que inicia e termina o processo do desentupimento respectivamente,  $f$  é o fator de entupimento máximo tal que  $0 < f < 1$  e  $f = 1 - \alpha(T_2 - T_1) = 1 - \beta(T_4 - T_3)$ .

No processo de *Clogging* real, por observação, é prudente afirmar que os

instantes de  $T_1 < T_2 < T_3 < T_4$  são aleatórios e encontram-se entre os intervalos:  $20s < T_2 - T_1 < 120s$  e  $10s < T_3 - T_2 < 60s$ ,  $0 < T_4 - T_3 < 5s$ .

Com o objetivo de simular o entupimento da válvula submersa, ou seja, simular a diminuição da área de passagem de aço pela válvula submersa, identifica-se que na Figura 5.6 deve-se incluir um ganho variável dado por  $K = A_{clg} * A_s$  para representar o efeito do *Clogging* no ponto onde está representada a área de passagem do aço, conforme ilustra a Figura 5.8.

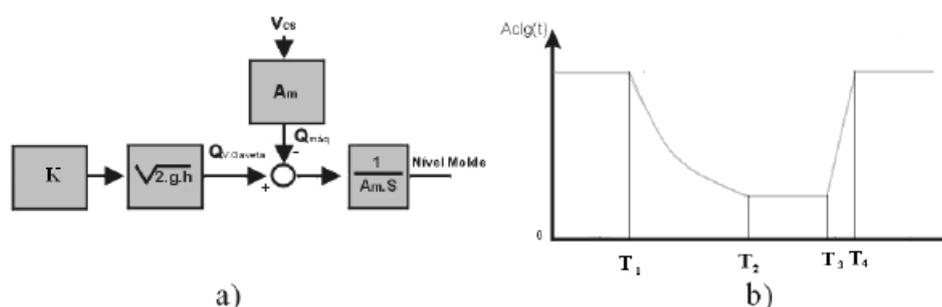


Figura 5.8: a) Diagrama de bloco com a inclusão do efeito de clogging; b) Representação gráfica da função de redução da área  $A_{clg}$ .

### Efeito *Bulging*

Conforme foi comentado, após a passagem pelo molde, a placa de aço no interior do veio, ou seja, no interior da máquina de lingotamento, possui uma camada de pele e em seu interior ainda tem aço líquido que será gradativamente resfriado pela adição de água na superfície até que esteja totalmente solidificado em seu interior, antes de sair da máquina.

Esta placa de aço, ainda no estado líquido, exerce, portanto uma força ferrostática que tende a empurrar o aço líquido para fora do veio, movimento impedido pelos rolos que compõem os segmentos. Como estes rolos possuem um espaçamento  $D$  entre eles, a pele de aço formada sede um pouco, de forma que o aço ocupe parte deste espaçamento. Ao passar pelos rolos, haverá diminuição do espaço e conseqüente laminação deste aço sólido/líquido, provocando o rearranjo de aço líquido no interior da máquina.

Após passar por entre os rolos, o aço expande-se novamente até o próximo conjunto, caso ainda não tenha sido formada pele espessa o suficiente para contrapor a pressão ferrostática. Este fenômeno de retração e expansão ocorre simultaneamente por toda a máquina onde há presença de aço líquido no interior, chegando até o molde. Como a máquina de Lingotamento possui uma velocidade de extração da placa em seu interior, verifica-se como efeito decorrente da expansão e retração interna da placa no interior da máquina o surgimento de ondas senoidais na superfície do molde. Estas ondas são consequência, portando, da variação de volume de aço no interior do veio, função do espaçamento entre os respectivos rolos de onde ocorre o fenômeno e da velocidade de extração. A frequência de oscilação pode ser determinada com boa aproximação por [105]:

$$f_{oscilao}(Hz) = \frac{V_c(m/min)}{D(m) * 60} \quad (5.47)$$

O aparecimento do fenômeno de Bulging ocorre principalmente na região de desdobramento do aço no interior da máquina, próximo ao molde, onde há maior volume de aço líquido no interior da placa e obviamente, menor espessura de pele de aço solidificado. A expressão (5.47) indica uma relação direta do fenômeno com aspectos construtivos da máquina: capacidade de refrigeração e de velocidade, distanciamento de rolos definidos em projeto e curvatura total de lingotamento.

Particularmente, dependendo da região onde há a ocorrência de Bulging e do espaçamento dos rolos que compõem esta região, pode ocorrer o aparecimento de frequências de oscilação diferentes e simultâneas na superfície do molde. A amplitude deste fenômeno está diretamente relacionada com a capacidade de refrigeração e tipo de aço lingotado, quanto menor a quantidade de carbono em sua composição em conjunto com a adição de ligas, maior é a maleabilidade do material e amplitude do fenômeno, dentre outros fatores de menor contribuição.

Pode-se verificar o comportamento do nível do molde perante o aparecimento do fenômeno, com uma ou duas frequências, conforme mostra o exemplo real da Figura 5.9 correspondente ao lingotamento de uma planta real, máquina de lingotamento N° 3 da Arcelor Mittal, na presença do fenômeno.

Para a simulação do efeito Bulging no processo identificado, procede-se da mesma forma que o realizado no item anterior. Será então necessário definir o ponto



Figura 5.9: Efeito Bulging na MLC03.

correto de inserção do fenômeno no diagrama de blocos do processo identificado, na Figura 5.6. Conforme descrito anteriormente, entende-se o efeito Bulging como uma variação de volume no molde. A figura 5.10 representa um possível ponto de inserção do efeito no modelo obtido.

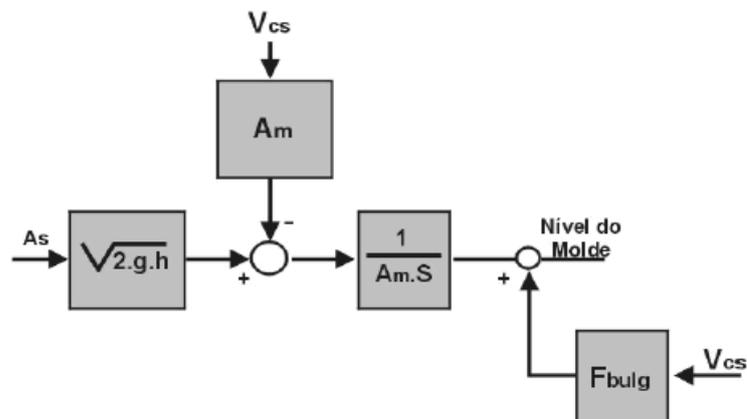


Figura 5.10: Ponto de Inclusão do Efeito Bulging no Processo

### Efeito do Argônio

A injeção de Argônio na válvula submersa é extremamente importante no processo do lingotamento contínuo: é utilizado para isolamento interno das paredes da

válvula gaveta e da válvula submersa quanto à incrustação de alumina, natural neste tipo de processo, nestes equipamentos [114].

A utilização do Argônio também é importante para a vedação do processo, impedindo o contato do aço líquido com o Oxigênio, provocando a oxidação do material lingotado.

O Argônio, após sua injeção, é arrastado pelo fluxo de aço no interior da válvula submersa e provoca um borbulhamento na superfície do molde, incluindo a região onde é efetuada a medição do nível de aço, refletindo em interferências na medição. Na Figura 5.11 verifica-se, através de simulação do lingotamento em um tanque de água, o comportamento do efeito provocado pelo borbulhamento de argônio na saída de dois modelos de válvula submersa comumente utilizados. Na Figura 5.11 também estão representados em vermelho os pontos onde são localizados os sensores de nível no molde na planta real.

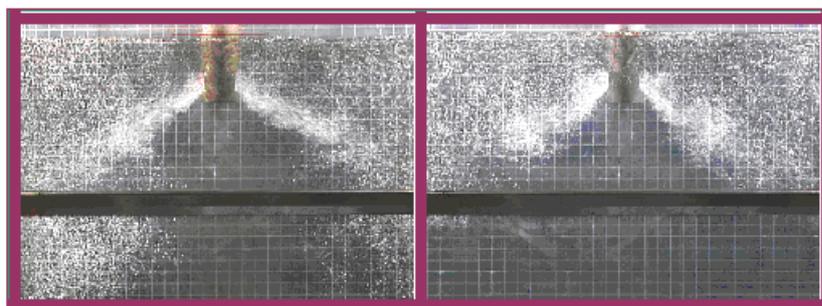


Figura 5.11: Simulação do borbulhamento de Argônio em um molde com água.

A amplitude das interferências por vazão de Argônio na medição de nível é proporcional a vazão do gás inserida no interior da válvula submersa e o efeito possui um comportamento estocástico quanto a sua frequência de influência. Na Figura 5.12 é apresentado um espectro do sinal do nível medido no molde em uma planta real de Lingotamento Contínuo, onde nota-se um exemplo da faixa de frequência atingida por este fenômeno, historicamente, de 0.4 a 0.8Hz.

Em virtude deste comportamento e da definição da amplitude também ser relacionada a uma dada situação real, insere-se diretamente na entrada do sensor um sinal de borbulhamento de argônio, previamente captado do processo real da máquina de lingotamento contínuo nº 3 da Arcelor Mittal.

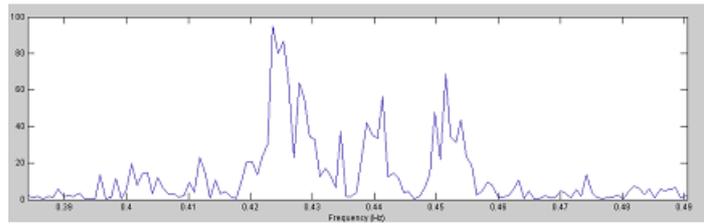


Figura 5.12: Espectro de frequência de borbulhamento de argônio num molde real.

### 5.3.3 O Modelo Neural do Nível do Molde

Com o objetivo de desenvolver um modelo do processo para fins de implementação da estratégia de controle avançada, realiza-se nesta seção a identificação do sistema utilizando-se redes neurais artificiais. A Figura 5.13 mostra o conjunto de entrada - saída ("input - output") utilizado na identificação. Estes dados foram gerados a partir de uma perturbação aleatória imposta à variável de entrada, dentro de certo limite (limite este factível do ponto de vista operacional), onde a abertura máxima da válvula gaveta é de 70 milímetros e seu curso pode ser visto na Figura 5.14. Os dados foram divididos em dois subconjuntos a saber: i) treinamento; ii) validação.

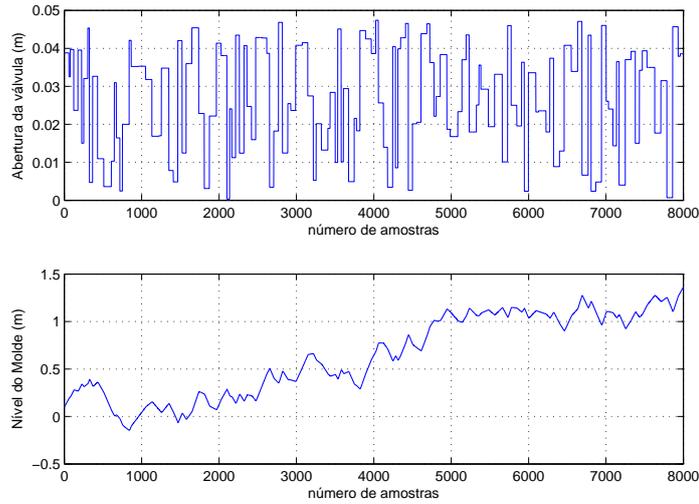


Figura 5.13: Dados de entrada e saída utilizados na identificação do modelo.

Em seguida os dados (treinamento e teste) foram escalonados de tal modo a apresentarem média zero e variância 1, procedimento este recomendável quando se tem em mente a utilização das redes neurais na correlação de tais dados.

A escolha de um regressor apropriado para descrever a dinâmica de um sistema não é uma tarefa fácil e pressupõe certo empirismo. Uma variável importante na escolha do regressor é a ordem do sistema. Tal ordem pode ser determinada a partir do conhecimento físico do sistema em estudo ou pode ser inferida a partir do conjunto de dados de entrada - saída. Neste último caso, pode-se empregar a metodologia proposta na literatura [57] para se determinar a estrutura do atraso ( $m$  e  $n$ ) antes da identificação do modelo dinâmico do sistema:

$$y(t) = f(y(t-1), \dots, y(t-n), u(t-1), \dots, u(t-m)) \quad (5.48)$$

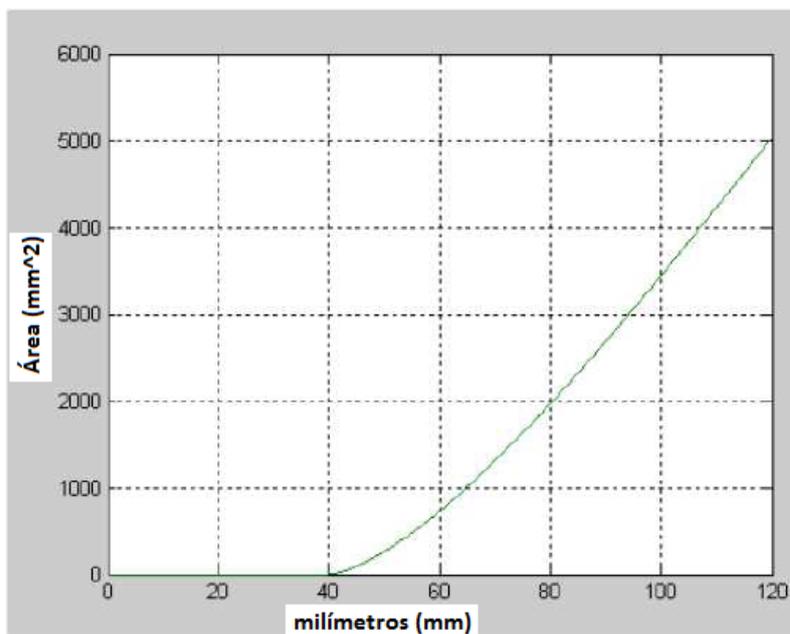


Figura 5.14: Curso x Área Efetiva da Válvula Gaveta.

Em [57], foi utilizado a função *lipschitz* para determinar a ordem do sistema. Isto pode ser visualizado na Figura 5.15, na qual mostra o índice em função do número de entradas e saídas passadas. Neste gráfico, o eixo das abscissas corresponde ao número de regressores que serão aplicados a entrada e a saída do modelo. A análise do gráfico é feito olhando os valores do eixo das ordenadas no ponto 1, 2, 3, 4, 5. Se a variação entre os pontos 1 e 2 for muito pequena, adota-se o valor 1, sendo assim,  $m = 0$  e  $n = 1$ , se o valor obtido for 2,  $m = 1$  e  $n = 1$  e assim sucessivamente. Pela Figura 5.15, percebe-se que a variação entre os pontos 4 e 5 é muito pequena, sendo assim, adotar-se-á um modelo com  $m = 2$ ,  $n = 2$  (ordem 2). A vantagem deste procedimento é evitar uma determinação iterativa e tediosa da quantidade de entradas a serem utilizadas no regressor conforme pode ser visto em [57].

O atraso do sistema foi escolhido por tentativa e erro, onde se adotou os valores  $m = 2$  e  $n = 2$ , e criaram-se vários modelos com diferentes valores para o atraso, onde o melhor modelo foi selecionado usando o critério de informação de Akaike (*AIC Akaike's Information Criterion*) e o critério de erros de predição final (*FPE Final Prediction Criterion*).

Através desta seleção, tem-se um modelo com  $m = 2$ ,  $n = 2$  e  $k = 1$ .

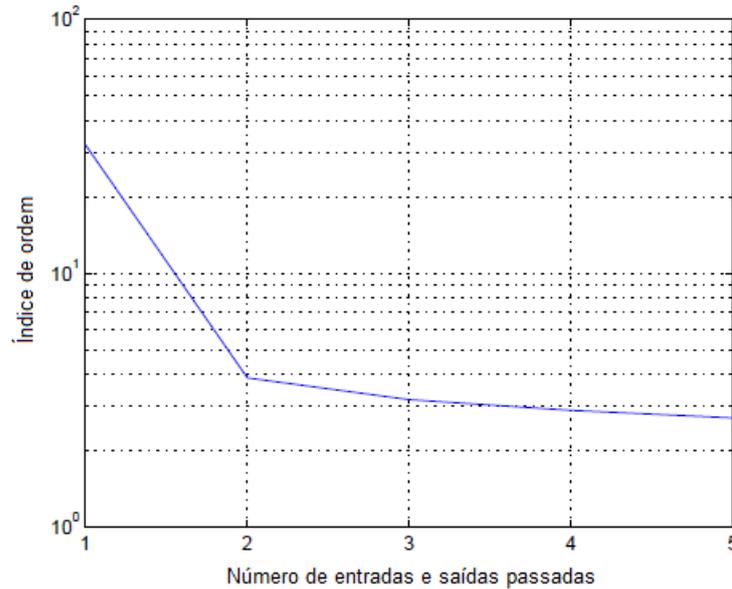


Figura 5.15: Efeito da ordem dos regressores no índice de avaliação proposto por [57].

Será identificada agora uma rede neural na forma do modelo ARX, rede que é chamada NNARX. Inicialmente, investiga-se o efeito do número de neurônios na camada interna, com função de ativação tangente hiperbólica na camada interna. A rede neural utilizada terá arquitetura totalmente conectada e função linear na camada de saída. Na Figura 5.16 mostra-se a saída real do processo e a saída identificada por redes neurais, assim como o erro de predição um passo a frente. A arquitetura da rede utilizada, possui uma camada oculta com 10 neurônios. Como se trata do modelo NNARX, este não possui característica recorrente, sendo que as entradas da rede serão definidos pelas saídas do processo  $y$  em dois intervalos de tempo anteriores  $[y(t - 1), y(t - 2)]$ , juntamente com o sinal de entrada  $u$  em dois instantes anteriores  $[u(t - 1), u(t - 2)]$ , caracterizando uma estrutura na forma  $[2, 2, 1]$ , ou,  $[n_a, n_b, d]$ .

Percebe-se através desta Figura 5.16, que a identificação realizada por redes neurais apresenta uma precisão satisfatória para o caso de predição um passo a frente, porém, quando a utilização da rede for em controle, o modelo não pode ser julgado por apenas este item.

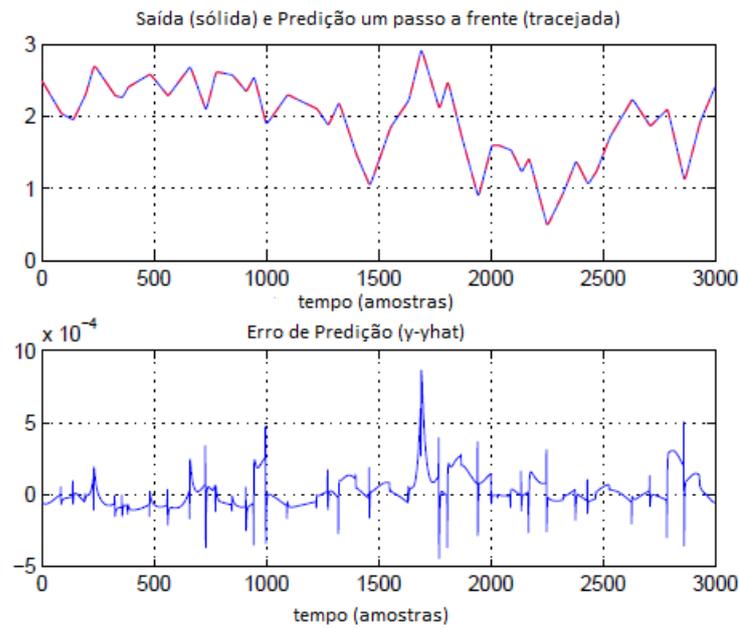


Figura 5.16: Validação do Modelo Neural.

Uma vez que o modelo obtido por redes neurais será utilizado como modelo de um processo, o qual será usado no cálculo do controlador preditivo, uma característica muito importante que deve ser analisada é a capacidade de predição do modelo. Desta forma, a Figura 5.17 mostra a saída real do processo juntamente com a saída da rede com predição de 30 passos a frente.

Através da Figura 5.17, percebe-se a capacidade que o modelo criado por redes neurais tem de mapear o sistema real, desta forma, conclui-se que o modelo apresentado por redes neurais será muito vantajoso para realizar o controle.

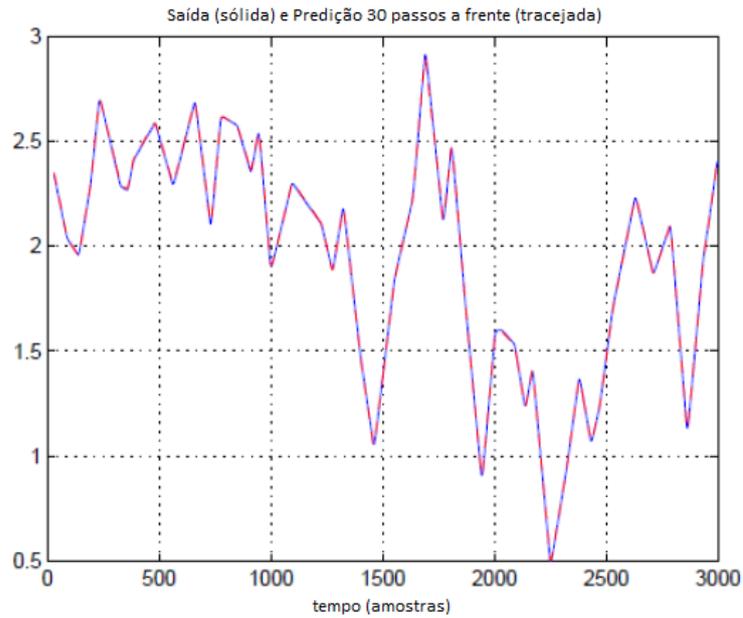


Figura 5.17: Saída real e saída prevista pela rede neural.

Primeiro, será mostrado o controle do nível do molde com e sem os distúrbios mencionados usando a técnica de Linearização Instantânea mostrado anteriormente, em seguida, o mesmo será feito utilizando a Regra do Gradiente para a obtenção do sinal de controle.

### 5.3.4 Controle Preditivo Neural

Uma vez que o modelo neural já está bem definido, será feito então o controle preditivo neural do nível do molde de lingotamento contínuo sem levar em conta a presença dos distúrbios relacionados tanto para o caso em que se utiliza a linearização instantânea (PNLI), quanto para o caso em que se utiliza a regra do gradiente (PNRG).

O algoritmo proposto por [87] para o controle preditivo neural será usado aqui tanto para o caso com linearização instantânea, quanto para o caso utilizando a regra do gradiente. É importante mencionar que foram feitas algumas modificações, tais como, a sintonia foi feita através de algoritmo genético, o que torna o algoritmo mais robusto, pois sem esta técnica de sintonia, este processo seria feito na tentativa e erro, o que iria tornar uma pesquisa exaustiva, além de não se ter a garantia de se

obter bons parâmetros, como é o caso do algoritmo genético.

Os parâmetros ajustados por AG para a sintonia do controle preditivo neural com linearização instantânea foram:  $N_u = 10$ ,  $N_2 = 10$ ,  $\alpha = 0.8843$ ,  $\delta = 91.3287$  e  $\lambda = 0.0796$ . Já os parâmetros ajustados por AG para a sintonia do controle preditivo neural com a regra do gradiente foram:  $N_u = 1$ ,  $N_2 = 1$ ,  $\alpha = 0,5755$ ,  $\delta = 747,6628$  e  $\lambda = 645,5345$ . Desta forma, a resposta apresentada pelo sistema de nível do lingotamento contínuo para os dois casos sem a presença de distúrbios pode ser visto na Figura 5.18 para o controlador com linearização instantânea e na Figura 5.19 para o controlador com a regra do gradiente.

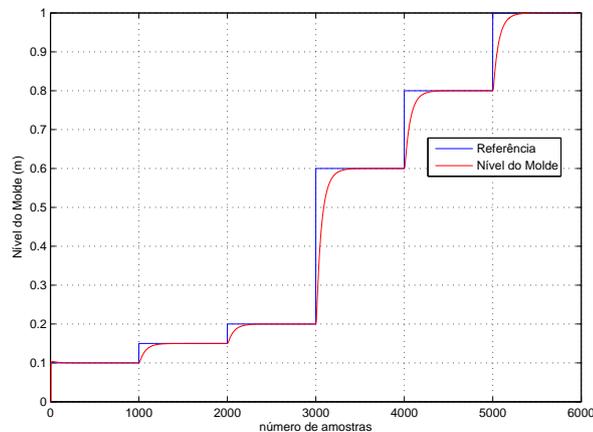


Figura 5.18: Resposta ao Degrau com o Controlador PNLI.

Percebe-se através das Figuras 5.18 e 5.19, que os controladores projetados, conseguem fazer com que o nível siga a referência em todo o intervalo da altura do molde, tendo apenas um pequeno erro em regime no caso do PNRG, mostrando assim, a eficiência do modelo e conseqüentemente dos controladores. Demonstra-se assim que os controladores projetados podem trabalhar em diversas regiões do molde, obtendo um controle muito satisfatório.

É importante mencionar também que o índice  $ISE$  apresentado pelos dois controladores foram muito próximos, sendo de 7,7477 para o PNLI e de 7,754 para o PNRG.

O tempo de simulação do algoritmo apenas com os parâmetros ótimos, ou seja, depois de selecionados os parâmetros, foi feita uma simulação com os parâme-

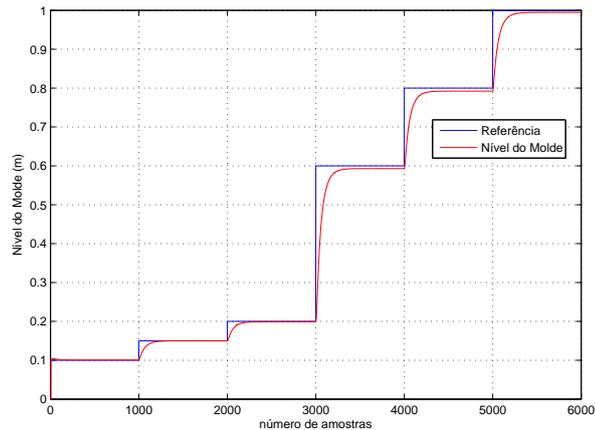


Figura 5.19: Resposta ao Degrau com o Controlador PNRG.

fosse escolhido para se comparar o tempo de processamento de cada controlador, foi de 32,67seg. para o PNLI e de 48,44seg. para o PNRG, o que já se tinha em mente, uma vez que o PNRG utiliza apenas rotinas não lineares, deixando-o mais pesado para simulação.

### Controle Preditivo Neural com Distúrbio de Bulging

Primeiramente será acrescentado ao processo o distúrbio de *bulging* mencionado anteriormente, com o objetivo de medir a eficácia dos controladores a este tipo de distúrbio. O distúrbio de bulging pode ser representado por uma senóide com uma amplitude de 20mm e uma frequência  $w = (2\pi)(0,077)$  rad/seg. Desta forma, a resposta apresentada pelo sistema de nível do lingotamento contínuo com a presença do distúrbio de *bulging* utilizando o PNLI pode ser visto na Figura 5.20, e na Figura 5.21, pode-se ver o nível do molde com a presença de *bulging* utilizando o PNRG.

Nas Figuras 5.20 e 5.21, percebe-se que a amplitude do *bulging* aplicado, tem sua amplitude reduzida de 20mm para cerca de 5mm, o que mostra a eficácia destes controladores para este tipo de caso, sendo observado um erro máximo de 5,7mm para o controlador PNLI e de 6,3mm para o controlador PNRG.

O tempo de simulação foi de 16,73seg. para o PNLI e de 25,49seg. para o PNRG. O tempo foi menor, pois o número de amostras utilizadas nesta simulação

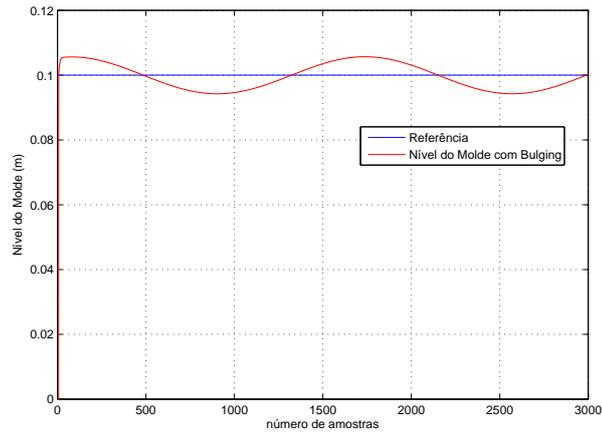


Figura 5.20: Simulação com Controlador PNLI e Distúrbio de *Bulging*.

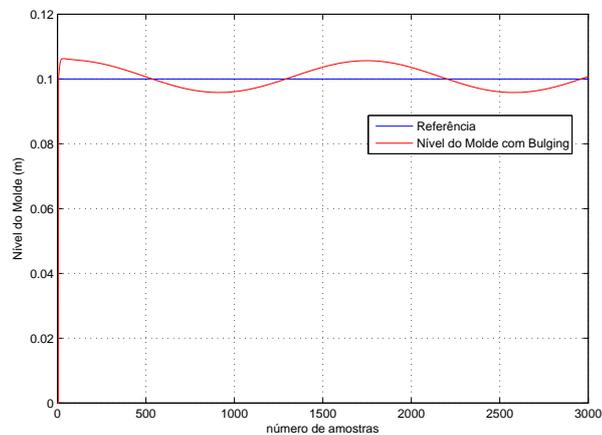


Figura 5.21: Simulação com Controlador PNRG e Distúrbio de *Bulging*.

foi menor, pois se queria saber apenas como os controladores iriam se comportar com a presença do distúrbio de *Bulging*.

### Controle Preditivo Neural com Distúrbio de Clogging

Neste parte, será mostrado o controle de nível do molde com o distúrbio de *clogging*, onde será considerado o pior caso do distúrbio, pois desta forma, tem-se um controlador preditivo robusto capaz de controlar o processo em todo seu estágio. Para este caso, o processo de entupimento da válvula ocorre num pequeno intervalo de tempo ( $T_2 = T_1 = 20\text{seg.}$ ), permanecendo obstruída com um determinado fator

de entupimento  $f$  durante um intervalo de tempo maior ( $T_3 - T_2 = 60\text{seg.}$ ) e em seguida, tem-se um desentupimento abrupto ( $T_4 - T_3 = 0$ ). O fator de entupimento  $f$  foi incrementado gradativamente de 10% em 10% para se obter o entupimento máximo que o controlador consegue controlar o nível.

As Figuras 5.22 e 5.23, mostram o controle do nível do molde com a presença do distúrbio de *clogging*, onde este entupimento chega até um máximo de 80% para o PNLI e de 75% para o PNRG, bem como os seus respectivos sinais de controle e variação do sinal de controle.

Para este caso, é importante mencionar que foi preciso fazer uma resintonia do processo para os dois controladores, tanto o PNLI, quanto para o PNRG, pois os parâmetros utilizados na simulação anterior, não foram capazes de estabilizar o sistema com o entupimento citado. Estes parâmetros foram  $N_u = 2$ ,  $N_2 = 4$ ,  $\alpha = 0.1342$ ,  $\delta = 14.2027$  e  $\lambda = 1.5825$  para o PNLI e  $N_u = 2$ ,  $N_2 = 4$ ,  $\alpha = 0.1522$ ,  $\delta = 825.8170$  e  $\lambda = 538.3424$  para o PNRG.

Percebe-se nas Figuras 5.22 e 5.23 que no momento em que se inicia o processo de entupimento, o nível do molde começa a cair abruptamente, sendo necessário então, aplicar um degrau no sistema, fazendo com que haja uma elevação do nível do molde, pois, caso contrário, o nível ficaria negativo durante um instante de tempo, o que não poderia acontecer na prática.

É importante mencionar também que mesmo com um entupimento maior que 80% para o PNLI e 75% para o PNRG, o controlador sempre mantém o sistema estável e que no caso PNLI, o degrau aplicado no momento de entupimento máximo para que o nível não venha a ter valores negativos é menor que no PNRG.

O algoritmo do controlador PNRG precisa de ser reiniciado no momento em que acontece o desentupimento, devido aos resíduos que se tem nos cálculos de rotinas deste controlador não linear, causando assim erros numéricos no procedimento de otimização.

O tempo de simulação foi de 4,86seg. para o PNLI e de 17,48seg. para o PNRG. Da mesma forma que no caso anterior, a simulação foi feita com um menor número de amostras, mas percebe-se nos três casos que o PNLI apresenta resultados melhores que o PNRG, além de se ter um menor tempo de simulação.

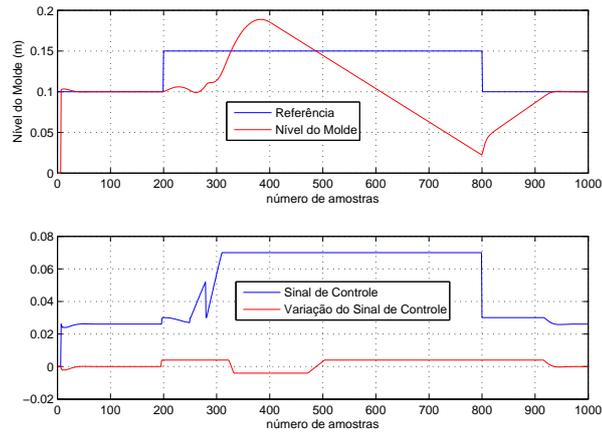


Figura 5.22: Simulação com Controlador PNL e Clogging com  $f=0,8$ .

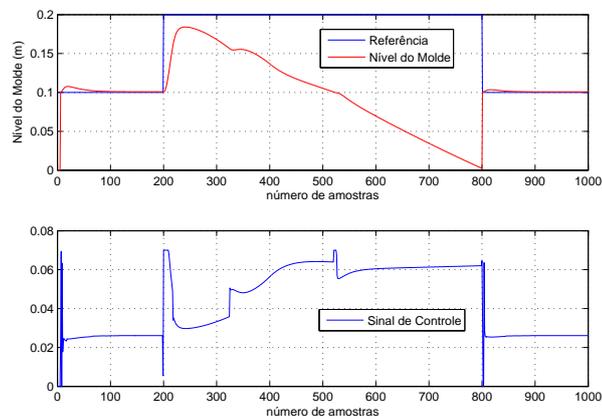


Figura 5.23: Simulação com Controlador PNRG e Clogging com  $f=0,75$ .

### 5.3.5 GPC para o Modelo Hammerstein (HGPC)

Nesta seção, será mostrado o controle preditivo utilizando o modelo Hammerstein. Os dados de entrada e saída do processo mostrado na Figura 5.13 para o treinamento das redes neurais, serão os mesmos utilizados para identificar o modelo Hammerstein.

Da mesma forma, os dados foram divididos em dois conjuntos, sendo um de identificação e outro para a validação. Com estes dados, foram criados vários modelos, com diferentes ordens tanto no grau da não-linearidade estática, quanto na dinâmica do modelo linear. Desta forma, foi feito a identificação do modelo

hammerstein utilizando o método dos mínimos quadrados conforme apresentado no Capítulo 2, seção 1.

Logo após, foi utilizado o critério de informação de Akaike (*AIC Akaike's Information Criterion*) e o critério de erros de predição final (*FPE Final Prediction Criterion*) com o objetivo de selecionar o melhor modelo.

O modelo selecionado possui uma estrutura do modelo linear de  $[2 \ 2 \ 0]$  e o grau da não-linearidade estática  $m$  igual a 2, sendo representado pela seguinte expressão:

$$y(t) = 1.365y(t - 1) - 0.365y(t - 2) - 0.701x(t - 1) + 1x(t - 2) \quad (5.49)$$

$$x(t) = 0.234u(t) + 1.0884u^2(t)$$

A Figura 5.24 mostra que a identificação do modelo Hammerstein apresenta uma precisão satisfatória para o caso de predição um passo a frente, desta forma, da mesma maneira que as redes neurais, o modelo não pode ser julgado somente por este item, sendo necessário testar a capacidade de predição do modelo apresentado.

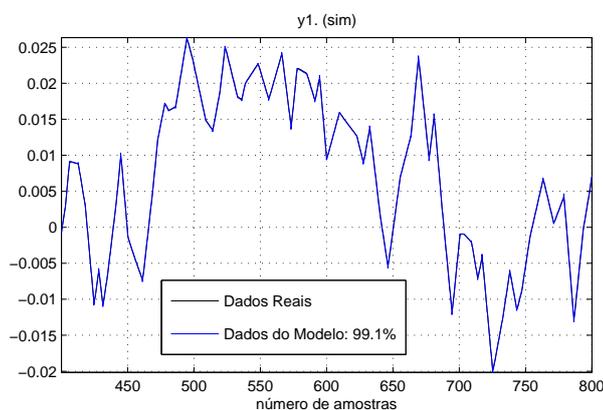


Figura 5.24: Saída real e saída identificada do Modelo Hammerstein.

A Figura 5.25 mostra a capacidade de predição do modelo identificado a 30 passos a frente, sendo assim, pode-se dizer que o modelo identificado tem uma grande probabilidade de ser um bom modelo para o controle preditivo.

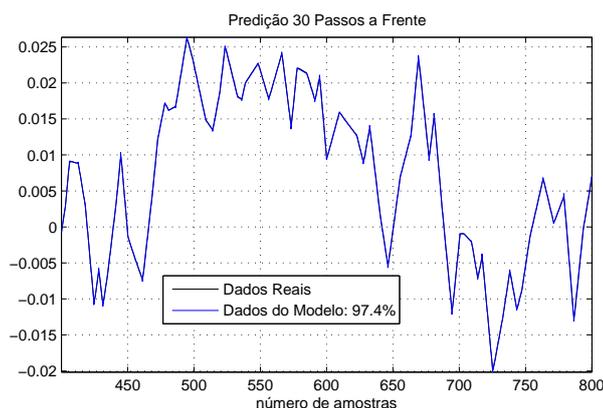


Figura 5.25: Saída real e saída prevista com o modelo Hammerstein.

### HGPC sem Distúrbio

Assim como na utilização de redes neurais, primeiramente realiza-se o controle do nível do molde sem a presença de distúrbio, para em seguida, ser acrescentado os distúrbios mencionados anteriormente.

Para as variações da entrada aplicada nos controladores PNLI e PNRG sem distúrbio, mostrado nas Figuras 5.18 e 5.19 não foi possível controlar o nível do molde utilizando o HGPC, pois, através de simulações, percebe-se que o HGPC não consegue controlar o nível do molde em toda a sua extensão, ou seja, o HGPC, controla o nível até  $70\text{cm}$  acima do ponto de operação. Foi visto também que o HGPC não consegue controlar o nível quando se aplica a entrada degrau com valor maior ou igual a  $30\text{cm}$ .

Desta forma, mostra-se aqui, uma simulação com uma outra faixa de entrada, a qual pode ser vista na Figura 5.26. Os parâmetros ajustados por AG para a sintonia do controle preditivo HGPC foram:  $N_u = 2$ ,  $N_2 = 7$ ,  $\alpha = 0,5836$ ,  $\delta = 37,3579$  e  $\lambda = 22,1695$ ;

A mesma entrada degrau foi simulada utilizando o PNLI e o PNRG para efeito de comparação com o HGPC. Os parâmetros do PNLI e do PNRG foram os mesmos utilizados anteriormente. Os índices de desempenho apresentados pelos controladores foram :  $0,9024$  para o HGPC,  $3,319$  para o PNLI e  $3,3046$  para o PNRG; o tempo de simulação gastos por cada um deles foi de  $18,178\text{seg.}$  para o HGPC,  $26,48\text{seg.}$  para o PNLI e  $40,89\text{seg.}$  para o PNRG.

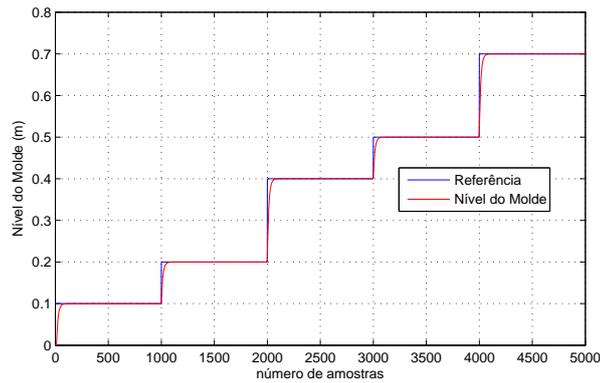


Figura 5.26: Resposta ao Degrau com Controlador HGPC.

Percebe-se que o HGPC, embora não trabalhe em toda extensão do molde, possui um índice ISE menor do que os outros dois controladores quando comparados numa faixa na qual o sistema é estável para todos os controladores.

### HGPC com Distúrbio de *Bulging*

Nesta seção, analisa-se o distúrbio de *bulging* no processo. Da mesma forma que os casos anteriores, este distúrbio apresenta uma oscilação constante e uma amplitude de  $20mm$ . Desta forma, a resposta apresentada pelo sistema de nível do lingotamento contínuo com a presença de *bulging* pode ser visto na Figura 5.27.

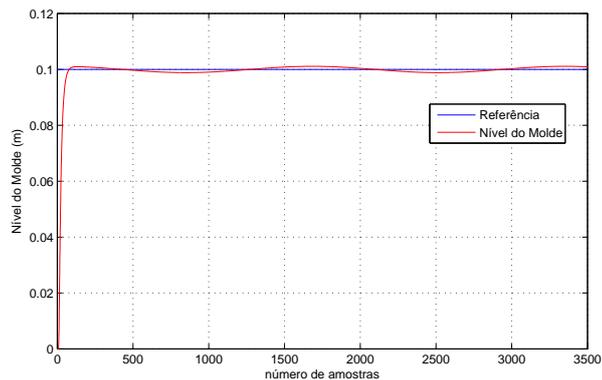


Figura 5.27: Resposta do Sistema com o Distúrbio de Bulging.

Através da Figura 5.27, observa-se que o distúrbio de *bulging* sofre uma re-

dução significativa, pois este distúrbio possui inicialmente uma amplitude de  $20mm$  que é reduzida para  $1mm$  após a compensação realizada pelo HGPC. Portanto este controlador tem uma eficiência melhor em tratar este tipo de caso do que o controlador usando redes neurais, pois a redução do efeito do distúrbio na rede foi de  $20mm$  para  $5mm$ . O tempo de simulação para este caso foi de  $12,89seg$ .

### HGPC com Distúrbio de *Clogging*

Agora será avaliado o controle do molde com o distúrbio de *clogging*, onde se consideram as mesmas condições dos outros tipos de controladores apresentados. Realizou-se também entupimentos gradativos de 10 em 10% para se obter o entupimento máximo que o controlador consegue se manter estável e controlar o nível dentro das restrições apresentadas.

A Figura 5.28, mostra o controle do molde com a presença do distúrbio de *clogging*, onde este entupimento chega a 90%, que é o máximo ocorrido na prática.

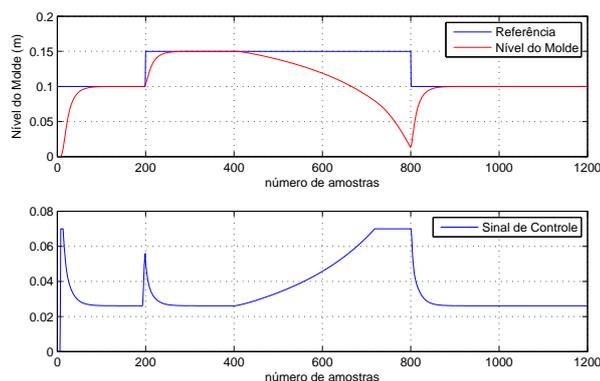


Figura 5.28: Resposta do Sistema com o Distúrbio de Clogging.

Da mesma forma que os casos anteriores, a medida em que a válvula vai entupindo, há a necessidade de se aumentar a referência para que o nível do tanque não fique negativo. Percebe-se aqui também que o controlador com o modelo Hammerstein, possui uma robustez maior para tratar este tipo de distúrbio, pois, enquanto o entupimento no controle realizado pelas redes neurais chegou a 80% para o PNLI e de 75% para PNRG, no controle usando o modelo Hammerstein, este entupimento chegou a 90% que é o entupimento máximo permitido, pois, após este

entupimento, o efeito de *unclogging* é realizado. O tempo computacional para este caso foi de 4,713seg..

### 5.3.6 HGPC com Ruído de Argônio

Nesta parte do trabalho, será mostrado a utilização do HGPC no controle do nível do molde do lingotamento contínuo com a presença do ruído de argônio. O HGPC foi escolhido, pois, em comparação com os outros algoritmos não-lineares usando redes neurais, foi o que apresentou maior robustez quando o processo apresenta distúrbios, além de sua implementação ser a mais simples para se projetar um Filtro de Kalman Estendido (veja apêndice B para maiores detalhes sobre este projeto).

Utilizando expressão linear do modelo de Hammerstein definido em (5.49) chega-se à seguinte função de transferência:

$$\frac{Y(z^{-1})}{X(z^{-1})} = \frac{-0,701z^{-1} + z^{-2}}{1 - 1,365z^{-1} + 0,365z^{-2}} \quad (5.50)$$

Transformando a função de transferência acima em uma equação de estados na forma canônica observável, tem-se que:

$$\begin{cases} xe_{t+1} = \Phi_t xe_t + \Gamma x_t + Y_k w_t \\ y_{t+1} = H_{t+1} xe_{t+1} + v_{t+1} \end{cases}$$

onde:

$$\Phi_t = \begin{bmatrix} 0 & -0,365 \\ 1 & 1,365 \end{bmatrix} \quad (5.51)$$

$$\Gamma_t = \begin{bmatrix} 1 \\ -0,701 \end{bmatrix} \quad (5.52)$$

$$Y_t = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5.53)$$

onde  $x_t$  expressa uma função não-linear em (5.49), fazendo com que o sistema, na verdade, tenha a seguinte forma:

$$\begin{cases} xe_{t+1} = f(xe_t, u_t, w_t) \\ y_{t+1} = h(xe_{t+1}) + v_{t+1} \end{cases}$$

Para este modelo identificado considera-se, através de tentativa e erro, que as autocovariâncias sejam:  $E[w_t w_t^T] = Q_t = 0$  e  $E[v_t v_t^T] = 0,9$ . De posse das matrizes mostradas acima, realiza-se expansões em séries de Taylor de modo a linearizar as equações em questão. Calculando as derivadas parciais das funções  $f$  e  $h$  (matrizes jacobianas) obtém-se:

$$Df(xe) = \begin{bmatrix} \frac{\partial f_1}{\partial xe_1} & \frac{\partial f_1}{\partial xe_2} \\ \frac{\partial f_2}{\partial xe_1} & \frac{\partial f_2}{\partial xe_2} \end{bmatrix} = \quad (5.54)$$

$$\begin{bmatrix} \frac{\partial(-0,365xe_2(t)+x(t))}{\partial xe_1} & \frac{\partial(-0,365xe_2(t)+x(t))}{\partial xe_2} \\ \frac{\partial(xe_1(t)+1,365xe_2(t)-0,701x(t))}{\partial xe_1} & \frac{\partial(xe_1(t)+1,365xe_2(t)-0,701x(t))}{\partial xe_2} \end{bmatrix} = \begin{bmatrix} 0 & -0,365 \\ 1 & 1,365 \end{bmatrix} \quad (5.55)$$

$$Df(w) = \begin{bmatrix} \frac{\partial f_1}{\partial w} \\ \frac{\partial f_2}{\partial w} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5.56)$$

$$Dh(x_e) = \begin{bmatrix} \frac{\partial h}{\partial y_1} & \frac{\partial h}{\partial y_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial(y_2(t))}{\partial y_1} & \frac{\partial(y_2(t))}{\partial y_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (5.57)$$

Sendo assim, de posse das matrizes jacobianas calculadas acima, inclui-se no simulador do sistema de controle do nível do molde do lingotamento contínuo o Filtro de Kalman Estendido (FKE). O objetivo é que o FKE incorpore toda informação obtida a partir das medições ruidosas, devido ao ruído ocasionado pela injeção de argônio, e estime de forma adequada as variáveis de estado.

Nas Figuras 5.29 e 5.30, é mostrado o controle do nível do molde com o HGPC após a injeção de argônio com e sem a utilização do Filtro de Kalman Estendido.

Percebe-se que o nível de ruído na válvula sem a utilização do FKE, é muito intenso, o que pode ocasionar problemas futuros na válvula, o que não acontece quando se utiliza o FKE.

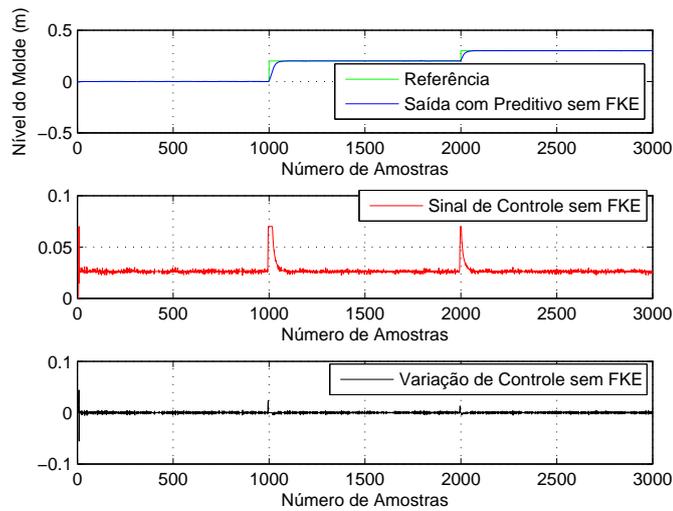


Figura 5.29: Resposta do Sistema com o Distúrbio de Argônio sem o FKE.

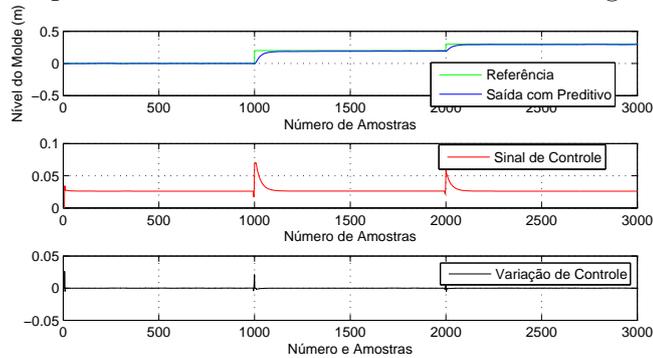


Figura 5.30: Resposta do Sistema com o Distúrbio de Argônio com o FKE.

## 5.4 Conclusão do Capítulo

Na Tabela 5.2, mostra-se uma síntese de todos os resultados obtidos com os 3 controladores não-lineares sujeitos aos distúrbios *clogging* e *bulging* para efeito de comparação. Uma discussão sobre esta tabela é feita no último capítulo desta tese.

Tabela 5.2: Comparação do Desempenho dos Controladores Não Lineares.

|   | PNLI                          | PNRG                            | HGPC                          |
|---|-------------------------------|---------------------------------|-------------------------------|
| $(N_u; N_2; \alpha; \delta; \lambda)$<br>com controle servo           | (10; 10; 0, 88; 91, 3; 0, 07) | (1; 1; 0, 57; 747, 6; 645, 5)   | (2; 7; 0, 58; 37, 35; 22, 17) |
| $(N_u; N_2; \alpha; \delta; \lambda)$<br>com <i>bulging</i>           | idem servo                    | idem servo                      | idem servo                    |
| $(N_u; N_2; \alpha; \delta; \lambda)$<br>com <i>clogging</i> e degrau | (2; 4; 0, 13; 14, 20; 1, 58)  | (2; 4; 0, 15; 825, 81; 538, 34) | idem servo                    |
| TP<br>para 5.000 amostras   | 26,48 seg.                    | 40,89 seg.                      | 18,178 seg.                   |
| ISE<br>no Controle Servo  | 3,319                         | 3,3046                          | 0,9024                        |
| $H_{max}$   | 1,2m                          | 1,2m                            | 0,7m                          |
| $R_{max}$   | 0,99m                         | 1,1m                            | 0,29m                         |
| $\Delta H_{max}$<br>com <i>bulging</i>                                | 5mm                           | 5mm                             | 1mm                           |
| $f_{max}$<br>com degrau   | 0,8                           | 0,75                            | 0,9                           |
| $f_{max}$<br>sem degrau   | 0,3                           | 0,58                            | 0,85                          |

Vale a pena comentar que as condições operacionais em que a simulação foi realizada no Matlab são:  $h = 1,2m$  (Altura do Distribuidor);  $A_m = 0,25m^2$  (área do molde);  $\bar{X}_{sg} = 0,35m$  (Abertura da válvula gaveta para uma velocidade constante de  $0,02m/seg$ ). O período de amostragem foi de  $0,1seg$ .

As restrições no processo são: abertura da Válvula Gaveta ( $u(t) \leq 0,7m$ ) e na sua velocidade de abertura ( $\Delta u(t) \leq 0,0041m$  por período de amostragem) e nos limites físicos do sensor do nível do molde ( $y(t) \leq 0,2m$ ).

É importante mencionar que para o controle do nível do molde do lingotamento contínuo, foi necessário trabalhar com todas as medidas em metro, para evitar saturação das redes neurais.

Outra observação importante é que nos controladores PNLI e HGPC, que utilizam o algoritmo GPC (ver Capítulo 3), e no PNRG, o cálculo da ação de controle foi obtido através da ação proporcional da forma  $\Delta u(t) = u(t) - u(t - 1)$ . Isto é devido ao fato do nível do molde apresentar uma elevada instabilidade à ação de controle integrativa dada por  $u(t) = u(t - 1) + \Delta u(t)$ . Isto pode ser visto através da análise de estabilidade do controlador PI feita em [105].

# Capítulo 6

## Conclusões Gerais

Neste Capítulo, são discutidas as principais contribuições desta tese, considerando os objetivos especificados no Capítulo 1.

Primeiramente, foi apresentado o algoritmo de sintonia do DMC desenvolvido por *Cooper* [34], na qual o processo deve ser aproximado por um modelo de primeira ordem com um atraso de transporte, pois neste algoritmo é necessário determinar a constante de tempo do modelo e do atraso de transporte. Esta técnica é utilizada somente para sistemas estáveis em malha aberta, mono e multivariáveis. Foi mostrado também que os parâmetros de sintonia usando este método apresentam valores elevados, sendo que em casos nos quais as restrições são consideradas, as mesmas não são respeitadas, indicando que a solução obtida pelo controlador preditivo é não factível.

Uma das contribuições originais deste trabalho é mostrar que o Algoritmo Genético é fundamental para a sintonia otimizada dos parâmetros do controlador preditivo, o qual pode ser aplicado para qualquer tipo de modelo de processo. Foi mostrado através de simulações em diferentes modelos, que a sintonia utilizando Algoritmos Genéticos apresentou parâmetros bem melhores do que o guia proposto por [34], além de se ter um índice de desempenho superior em todas as situações. É importante mencionar também que os processos sintonizados por AG, respeitaram as restrições na maioria dos modelos avaliados.

Outra contribuição desta tese foi a aplicação do GPC multivariável num sistema de laminação com modelo linearizado na presença de distúrbios. Devido ao grande número de parâmetros envolvidos na sintonia do controlador preditivo

a fim de estabilizar este processo, que é multivariável, instável em malha aberta e com distúrbios, pode-se concluir que esta aplicação só é viável na prática se a sintonia for realizada via AG. Neste processo de laminação, comparou-se a técnica FFF utilizada na indústria SIDERAR da Argentina e o GPC sintonizado por AG. Foram feitas duas análises: uma utilizando o caso regulatório, nos quais os distúrbios de temperatura e de espessura de entrada foram levados em consideração; e outra utilizando o caso servo, no qual foram aplicados degraus em diferentes instantes de tempo para se avaliar o acoplamento entre as saídas do processo, bem como a eficácia do GPC para o desacoplamento das malhas.

Outra contribuição relevante desta tese foi a solução do problema de controle de nível do molde do lingotamento contínuo, cujo modelo foi obtido a partir de dados reais levando-se em conta diferentes distúrbios. Foram utilizados três algoritmos de controle preditivo não-linear sintonizados pelo algoritmo genético, sendo eles PNLI, PNRG e HGPC.

Através da Tabela 5.2 mostrada no Capítulo 5, verifica-se que os controladores PNLI e PNRG são estáveis às entradas degraus em toda extensão do molde. Além disso percebe-se que a amplitude do degrau aplicado pode ser maior no PNLI e PNRG do que o HGPC. Constata-se também neste mesmo caso que, na faixa de aplicação do HGPC, este possui uma resposta melhor e um tempo computacional menor que todos os outros controladores.

Quanto ao distúrbio de *bulging*, observa-se que a resposta apresentada pelo HGPC foi melhor do que os outros dois tipos, uma vez que a redução do distúrbio de *bulging* para o PNLI e PNRG foi 20mm para 5mm e no HGPC esta redução foi de 20mm para 1mm.

Em relação ao distúrbio de *clogging*, a melhor resposta foi apresentada pelo HGPC, pois o mesmo conseguiu controlar o processo mesmo com a válvula estando com 90% de entupimento, já o PNLI foi de 80% e o PNRG foi de 75%. Estes resultados foram obtidos aplicando uma entrada degrau com o objetivo de manter o nível do molde acima de zero. Se o degrau não for aplicado o HGPC suportaria um *clogging* de até 85%, o PNLI suportaria um *clogging* de até 30% e o PNRG suportaria um *clogging* de até 58%. É importante mencionar também que os mesmos parâmetros utilizados no HGPC para o caso sem distúrbios também foram utilizados para o caso onde foi levado em conta os distúrbios de *clogging* e

de *bulgging*, o que não foi possível para o PNLI e o PNRG, sendo necessária uma resintonia dos parâmetros quanto o distúrbio de *clogging* foi aplicado.

Por último, o controlador preditivo HGPC foi utilizado para controlar o nível do lingotamento contínuo com a presença do ruído de argônio. Percebe-se que neste caso, foi necessário a implementação de um FKE, pois, devido ao nível de ruído injetado no processo, a válvula teria sua atuação prejudicado, fazendo com que o tempo de vida útil de operação fosse reduzido.

É importante mencionar também que o HGPC foi escolhido pelo fato de o mesmo ter se comportado melhor quando os distúrbios de *Clogging* e *Bulging* foram levados em conta e também por ser o mais compatível com a utilização do FKE.

Com o desenvolvimento deste trabalho, criou-se algumas linhas de pesquisas para serem desenvolvidas em trabalhos futuros, entre as quais:

1. Análise Matemática da estabilidade e robustez dos algoritmos de controle preditivo não-lineares usando modelos Hammerstein e Redes Neurais para controlar o nível do molde;
2. Implementação de um controle preditivo multivariável e não-linear aplicado ao processo de laminação;
3. Desenvolvimento de um controlador preditivo fuzzy para o controle do nível do molde do lingotamento contínuo;
4. Estudo de filtros usando redes neurais que possam ser utilizados nos algoritmos do Controlador Preditivo Neural PNLI e PNRG.

# Referências Bibliográficas

- [1] AGUIRRE, L. A. *Introdução à Identificação de Sistemas - Técnicas Lineares e Não-Lineares Aplicados a Sistemas Reais*, 3 ed ed. Editora UFMG, Belo Horizonte, MG, 2007.
- [2] AGUIRRE, L. A., COELHO, M. C. S., AND CORRÊA, M. V. On the interpretation and practice of dynamic differences between hammerstein and wiener models. *IEE Proceedings- Control Theory Applications* 152, 4 (2005), 349–356.
- [3] AL-DUWAISH, H., AND NAEEM, W. Nonlinear model predictive control of hammerstein and wiener models using genetic algorithm. *Proc. IEEE Conference on Control Applications* (2001), 465–469.
- [4] AL-SEYAB, R. K., AND CAO, Y. Nonlinear model predictive for the alstom gasifier. *Journal of Process Control* 16, 8 (2006), 795–808.
- [5] ASTROM, K. J., AND HAGGLUND, T. *PID Controllers: Theory, Design and Tuning*, vol. 2. Instr. Society of America, 1998.
- [6] ASTROM, K. J., AND WITTENMARK, B. *Adaptive Control*. Addison - Wesley, USA, 1995.
- [7] BARS, R., AND HABER, R. Weighted one-step-ahead adaptive predictive control of nonlinear processes. *IMACS Symposium Modelling and Control of Technological Processes 1* (1991), 16–21.
- [8] BAUGHMAN, D. R., AND LIU, Y. A. *Neural networks in bioprocessing and chemical engineering*. Academic Press, 1995.

- [9] BHAT, N., AND MCAVOY, T. Use of neural nets for dynamic modeling and control of chemical process systems. *Computer Chemical Engineering* 14, 4/5 (1990), 573–582.
- [10] BITMEAD, R. R., GEVERS, M., AND WERTZ, V. *Adaptative optimal control. The thinking man's GPC*. System and Control Engineering Englewood Cliffs, New Jersey: Prentice-Hall, 1995.
- [11] BLICKLE, T., AND THIELE, L. A comparison of selection schemes used in genetic algorithms. *Computer Engeneering and Communication Networks Lab* (July 1995), 173–176.
- [12] BOUTAYEB, M., RAFARALAHY, H., AND DAROUACH, M. A robust an recursive identification method for the hammerstein model. *Proc. 13th IFAC Triennial World Congress* (1996), 447–452.
- [13] BULUT, B., KATEBI, M. R., AND GRIMBLE, M. J. Predictive control of hot rolling processes. *Proceedings of the American Control Conference* (2000).
- [14] CAMACHO, E. F., AND BORDONS, C. *Model Predictive Control*. New York: Springer, 2004.
- [15] CAMPOS, M. M., AND SAITO, K. *Sistemas inteligentes em controle e automação de processos*. Ciência Moderna, Rio de Janeiro, 2004.
- [16] CHIRAS, N. *Linear and Nonlinear Modelling of Gas Turbine Engines*. PhD thesis, 2002.
- [17] CHOI, I. S., ROSSITER, J. A., AND FLEMING, P. J. Robust constrained predictive controllers for hot rolling mills: disturbance uncertainty case. *Systems and Control Engeneering* 222 (2007).
- [18] CLARKE, D. W. Application of generalized predictive control to industrial processes. *IEEE Control Systems Magazine* (1988).
- [19] CLARKE, D. W., MOHTADI, C., AND TUFFS, P. S. Generalized predictive control. part i. the basic algorithm. *Automatica* (1987), 137–148.
- [20] COELHO, A. A. R., AND COELHO, L. S. *Sistemas Dinâmicos Lineares*. 2004.

- [21] COELHO, A. A. R., AND SANTOS, J. E. S. Identificação e controle preditivo para o modelo de hammerstein: Abordagem não paramétrica. *XIV COngresso Brasileiro de Automática* (2002), 2810–2815.
- [22] COELHO, A. A. R., SANTOS, J. E. S., ALMEIDA, O. M., SUMAR, R. R., AND CALLAI, T. C. Identification and predictive control for the hammerstein model: Nonparametric approach. *V INDUSCON - Conferência de Aplicações Industriais* (2002), 370–374.
- [23] COELHO, L. S., AND COELHO, A. A. R. Algoritmos evolutivos em identificação e controle de processo: Uma visão integrada e perspectivas. *SBA Controle e Automação* 10, 1 (1999).
- [24] CORRÊA, L. G. C. Inferência de qualidade de produtos de destilação utilizando redes neurais artificiais e filtro de kalman estendido. Master's thesis, Dissertação de Mestrado, PUC - Rio, 2005.
- [25] CUTLER, C. R., AND RAMAKER, B. C. Dynamic Matrix Control- A Computer Control Algorithm. *In Automatic Controle Conference* (1980).
- [26] CYBENKO, G. V. Approxomation by superpositions of a sigmoid function. *Mathematics of Control, Signals, and Systems* 2 (1989), 36–42.
- [27] DAS, R., AND GOLDBERG, D. E. Discrete-time parameters estimation with genetic algorithm. *Proceedings of the 19th Annual Pittsburg Conference on Modeling and Simulation* (1988).
- [28] DE ALMEIDA, G. M. Sintonia de controle preditivo generalizado através de algoritmo genético. Master's thesis, Dissertação de Mestrado, PPGEE/UFES, 2006.
- [29] DE ALMEIDA, G. M., SALLES, J. L. F., AND FILHO, J. Using genetic algorithm to optimize the tuning parameters of dynamic matrix control. *I Simpósio Brasileiro de Inteligência Computacional* (2007).
- [30] DE ALMEIDA, G. M., SALLES, J. L. F., AND FILHO, J. D. Optimal tuning parameters of the dynamic matrix predictive controller with constraints. *XII Congresso Brasileiro de Automática* (2008).

- [31] DE ALMEIDA, G. M., SALLES, J. L. F., AND FILHO, J. D. Optimal tuning parameters of the dynamic matrix predictive controller with constraints. *Latin American Research Applied* 39, 1 (2009).
- [32] DENNIS, J. E., AND SCHNABEL, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equation*. Prentice-Hall, 1983.
- [33] DORF, R. C., AND BISHOP, R. H. *Sistemas de controle Modernos*. TC Publishing Company, 1995.
- [34] DOUGHERTY, D., AND COOPER, D. Tuning guidelines of a dynamic matrix controller for integrating (non-self-regulating). *Processes, Ind. Eng. Chem. Res.* 42 (2003), 1739–1752.
- [35] DOYLE, F. J., PEARSON, R. K., AND OGUNNAIKE, B. A. *Identification and Control Using Volterra Models*. Springer-Verlag, 2001.
- [36] ESKINAT, E., JOHNSON, S. H., AND LUYBEN, W. L. Use of hammerstein models in identification of nonlinear systems. *AIChE Journal* 37, 2 (1991), 255–268.
- [37] FABRO, J. A. Uma abordagem neuro-nebulosa para controle preditivo de processos multi-estágios. Tech. rep., Centro Federal de Educação Tecnológica do Paraná, Curitiba, 2003.
- [38] FARAGO, A., AND LUGOSI, G. Strong universal consistency of neural networks classifiers. *IEEE Trans. On Information Theory* 39 (1993), 1146–1151.
- [39] FAVIER, G., KIBANGOU, A. Y., AND KHOUAJA, A. Nonlinear system modelling by means of volterra models. approaches for parametric complexity reduction. *Symposium Techniques Avancées et Stratégies Innovantes en Modélisation et Commande Robuste de Processus Industriels* (2004).
- [40] FILALIT, S., AND WERTZ, V. Using genetic algorithms to optimize the design parameters of generalized predictive controllers. *International Journal of Systems Science* 32, 4 (2001), 503–512.

- [41] FILHO, J. D., BATISTA, E., AND HELMAN, H. Análise de sensibilidade na laminação de chapas - uma ferramenta para a monitoração de parâmetros operacionais específicos. *X CBA/VICLCA* (1994), 476–481.
- [42] FILHO, J. D., AND HELMAN, H. Um modelo teórico para o controle automático da espessura de chapas laminadas a frio. *I Metal Forming Week* (1993), 123–134.
- [43] FILHO, J. D., AND HELMAN, H. Monitoração de parâmetros operacionais no processo de laminação de tiras por análise de sensibilidades. *49 Congresso Anual da ABM* (1994), 69–77.
- [44] FILHO, J. D., AND HELMAN, H. Uma técnica para o controle da espessura de saída na laminação de tiras baseadas nas curvas de operação do processo com avaliação da curva de operação desviada. *49 Congresso Anual da ABM* (1994), 61–68.
- [45] FILHO, J. D., AND LIMA, L. E. M. A new theoretical exit thickness control model for rolling process. *IASTED - International Association of Science and Technology for Development , International Conference on Modelling - Identification and Control* (1997).
- [46] FINK, A., AND NELLES, O. Nonlinear internal model control based on local linear neural networks. *Proc. International Conference on Systems, Man and Cybernetics 1* (2001), 117–122.
- [47] FONTES, A. B., MAITELLI, A. L., AND SALAZAR, A. O. A new bilinear generalized predictive control approach: Algorithm and results. *Proc. 15th IFAC Triennial World Congress* (2002), 1390–1395.
- [48] FRANK, H., PICHLER, R., SCHOISSWOHL, M., AND STAUFER, W. Improved rolling-mill automation by means of advanced control techniques and process models. *Proceedings of automation in mining, mineral and metal processing* (1998), 315–323.
- [49] FRUZZETI, K. P., PALAZOGLU, A., AND MCDONALD, K. A. Nonlinear model predictive control using hammerstein models. *Journal of Process Control* 7, 1 (1997), 31–41.

- [50] GERKSIC, S., JURIDIC, D., STRMCNIK, S., AND MATKO, D. Wiener model based nonlinear predictive control. *International Journal of Systems Science* 31, 2 (2000), 189–202.
- [51] GIANNAKIS, G. B., AND SERPEDIN, E. A bibliography on nonlinear system identification. *IEEE Transactions Signal Processing* 81, 3 (2001), 533–580.
- [52] GOLDGERB, D. E. *Genetic algorithms in search, optimization, and machine learning*. Adison-Wesley. Ann Arbor, 1989.
- [53] GREFENSTETTE, J. J., GOPAL, R., ROSMAITA, B. J., AND GUCHT, D. V. Genetic Algorithms for the Traveling Sales man Problem. *Proceedings of an International Conference on Genetic Algorithms and their Applications* (1985), 160–168.
- [54] HABER, R. Predictive control of nonlinear dynamic processes. *Applied Mathematics and Computation* 70, 2-3 (1995), 169–184.
- [55] HARRIS, C. J., MOORE, C. G., AND BROWN, M. Intelligent control: Aspects of fuzzy logic and neural nets. *World Scientific Co.* (1993).
- [56] HAYKIN, S. *Redes Neurais: Princípios e Práticas. Trad. Paulo Martins Engel*, 2 ed. Bookman, Porto Alegre, 2001.
- [57] HE, X., AND ASADA, H. A new method for identifying orders of input-output models for nonlinear dynamic systems. *Proc. of the American Control* (1993).
- [58] HECHT-NIELSEN, R. Theory of the backpropagation neural network. *IEEE Int. Conf. On Neural Networks* 1 (1989), 593–605.
- [59] HELMAN, H., AND CETLIN, P. R. *Fundamentos da Conformação Mecânica dos Metais*. Guanabara Dois, 1983.
- [60] HOLLAND, J. H. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, Berlim, 1975.
- [61] HWANG, C. L., AND HSU, J. C. Nonlinear control design for a hammerstein model system. *IEE Proceedings - Control Theory Applications* 142, 4 (1995), 277–285.

- [62] ISERMANN, R., LACHMANN, K. H., AND MATKO, D. *Adaptive Control Systems*. Prentice-Hall, Upper Saddle, NJ, 1992.
- [63] JEON, Y. S., LEE, C. O., AND HONG, Y. S. Optimization of the control parameters of a pneumatic servo cylinder drive using genetic algorithms. *Control Engineering Practice* 6 (1998), 847–853.
- [64] JURADO, F. Predictive control of solid oxide fuel cells using fuzzy hammerstein models. *Journal of Power Sources* 158, 1 (2006), 245–253.
- [65] KARA, T., AND EKER, I. Nonlinear modeling and identification of a dc motor for bidirectional operation with real time experiments. *International Journal of Energy Conversion and Management* 45, 7-8 (2004), 1087–1106.
- [66] KATENDE, E., AND JUTAN, A. Nonlinear predictive control of complex processes. *Industrial Engineering Chemical Research* 35, 10 (1996), 3539–3546.
- [67] KATENDE, E., JUTAN, A., AND CORLESS, R. Quadratic nonlinear predictive control. *Industrial Engineering Chemical Research* 37, 7 (1998), 2721–2728.
- [68] KEYSER, R. M. C. D., AND CUAWENBERGHE, A. R. V. Extended Prediction Self-Adaptive Control. In *IFAC Symp. on Identification and System Parameter Estimation* (1985), 1317–1322.
- [69] KONDO, T. K. Development of mold level control in continuous casting by h infinity control theory. *2nd IEEE Conference on Control Application* (1993).
- [70] LAST, M., AND EYAL, S. A fuzzy-based lifetime extension of genetic algorithms. *Fuzzy Sets and Systems* 149 (2005), 131–147.
- [71] LELIC, M. A., AND WELLSTEAD, P. E. Generalized pole placement self tuning controller. part 1. basic algorithm. *Int. J. Control* (1987).
- [72] LJUNG, L. *System Identification: Theory for the User*, 2nd ed. Prentice Hall, 1999.
- [73] LJUNG, L., AND GLAD, T. *Modelling of Dynamic Systems*. Prentice-Hall, Englewood Cliffs NJ, 1994.

- [74] LOPES, H. S. Algoritmos genéticos em projetos de engenharia: aplicações e perspectivas futuras. *Anais do 4 Simpósio Brasileiro de Automação Inteligente* (1999), 64–74.
- [75] LOUREIRO, R. A. M. Controle do nível do aço de uma máquina de lingotamento contínuo. Master's thesis, Dissertação de Mestrado, FEE - UNICAMP, 1994.
- [76] LUINENBURG, A. W., ONINK, M., AND RITMAN, V. A. New physical on-line model for coiling temperature control. *Proceedings of the seventh international conference on steel rolling*. (1998), 87–92.
- [77] MACALISTER, A. F., REEVE, P. J., AND SMITH, P. Predictive control of temperature and width for hot strip mills. *Proceedings of automation in mining, mineral and metal processing* (1998), 243–249.
- [78] MACIEJOWSKI, J. *Predictive Control with Constraints*. Englewood Cliffs, NJ: Prentice Hall, 2002.
- [79] MAHFOUF, M., AND LINKENS, D. A. *Generalized Predictive Control and Bioengineering*. Taylor & Francis, Londres, 1998.
- [80] MANER, B. R., III, F. J. D., OGUNNAIKE, B. A., AND PEARSON, R. K. A nonlinear model predictive control scheme using second order volterra models. *Proc. American Control Conference* (1994), 3253–3257.
- [81] MARCHI, P. A. Abordagem não linear para modelagem e controle preditivo: Estudo de casos. Master's thesis, PGEEL/UFSC, Florianópolis, SC, 1999.
- [82] MCCULLOCH, W. S., AND PITTS, W. A logical calculus of the ideas immanent in neurons activity. *Bull Math. Biophys* 5 (1943).
- [83] MELEIRO, L. A. C. *Projeto e Aplicações de Controladores Baseados em Modelos Lineares, Neurais e Nebulosos*. PhD thesis, 2002.
- [84] MICHALEWICZ, Z. *Genetic Algorithms + data structures = evolution programs*. Springer-Verlag, New York, 1996.
- [85] MITCHELL, M. *An Introduction to Genetic Algorithms*. A Bradford Book. The MIT Press, London, 1997.

- [86] NARENDA, K., AND PARTHASARATHY, K. Identification and control of dynamic systems using neural networks. *IEEE Trans. On Neural Networks* 1, 1 (1990), 4–27.
- [87] NORGAARD, M., RAVN, O., POULSEN, N. K., AND HANSEN, L. K. *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag, 2000.
- [88] NORQUAY, S. J., PALAZOGLU, A., AND ROMAGNOLI, J. A. Model predictive control based on wiener models. *Chemical Engineering Science* 53, 1 (1998), 75–84.
- [89] OSMAN, I. H., AND KELLY, J. P. *Meta-Heuristics: Theory and Applications*. Kluwer Academic Publishers Norwell, 1996.
- [90] PEARSON, R. K. Selecting nonlinear model structures for computer control. *Journal of Process Control* 13, 1 (2003), 1–26.
- [91] PEARSON, R. K., AND POTTMANN, M. Gray-box identification of block-oriented nonlinear models. *Journal of Process Control* 10, 4 (2000), 301–315.
- [92] PÉREZ, E., BLASCO, X., GARCIA-NIETO, S., AND SANCHIS, J. Diesel engine identification and predictive control using wiener and hammerstein models. *Proc. IEEE International Conference on Control Applications* (2006), 2417–2423.
- [93] QIN, S., AND BADGWELL, T. A survey of industrial model predictive control technology. *Control Engineering Practice* 11, 7 (2003), 733–764.
- [94] QIN, S. J., AND BADGWELL, T. A. An overview of industrial model predictive control technology. *Fifth International Conference of Chemical Process Control* (1997), 232–256.
- [95] RAWLINGS, J. B. Tutorial overview of model predictive control. *IEEE Control Systems Magazine* 20, 3 (2000), 38–52.
- [96] RICHALET, J., EL ATA-DOSS, S. A., ARBER, C., KUNTZE, H. B., JACUBASH, A., AND SCHILL, W. Predictive Functional Control. Application to fast and accurate robots. *In Proc. 10th IFAC Congress* (1987).

- [97] ROSSITER, J. A. *Model-Based Predictive Control: A Practical Approach*. Boca Raton, FL: CRC Press, 2003.
- [98] ROSSOMANDO, F. G. *Estabelecimento de Estratégias de Controle Inteligente na Laminação de Produtos Planos*. PhD thesis, 2006.
- [99] ROSSOMANDO, F. G., FERREIRA, E. P., AND FILHO, J. Controle neural adaptativo de um trem de laminação a quente. *VII Simpósio Brasileiro de Automação Inteligente* (2005).
- [100] ROSSOMANDO, F. G., AND FILHO, J. D. Modelling and control of a hot rolling mill. *Latin American Research Applied* 36 (2006), 199–204.
- [101] ROUHANI, R., AND MEHRA, R. K. Model algorithm control; basic theoretical properties. *Automatica* 18, 4 (1982), 401–414.
- [102] ROUX, G., DAHOU, B., AND QUEINNEC, I. Modelling and estimation aspects of adaptive predictive control in a fermentation process. *Control Engineering Practice* 4, 1 (1996), 56–66.
- [103] RUDOLPH, G. *Convergence Properties of Evolutionary Algorithms*. New York: Springer, Kovsc, Hamburg, 1997.
- [104] RUMELHART, D. E., AND MCCLELLAND, J. L. *Parallel distributed processing: Explorations in the microstructure of cognition*. The Mit Press, 1986.
- [105] SANCHOTENE, F. B. Modelagem e controle no sistema de nível de aço no molde do lingotamento contínuo da amt. Master’s thesis, Dissertação de Mestrado, DEL-UFES, 2008.
- [106] SANTOS, J. E. S. *Controle Preditivo Não Linear para Sistemas Hammerstein*. PhD thesis, 2007.
- [107] SANTOS, J. E. S., SUMAR, R. R., AND COELHO, A. A. R. Uma solução para a multiplicidade da lei de controle preditivo para o modelo hammerstein. *XV Congresso Brasileiro de Automática* (2004).
- [108] SCHMITT, F., AND ROTHLAUF, F. On the mean of the second largest of the second largest eigevalue on the convergence rate of genetic algorithms. Tech.

- rep., Working Paper in Information Systems, University of Bayreuth, Dep. of Information System, Germany, 2001.
- [109] SJOBERG, T. Nonlinear system identification with neural networks. Master's thesis, Electrical Engineer Department, Linkoping, Sweden, 1995.
  - [110] SODERSTRON, T. *Discrete-time Stochastic System: Estimation and Control*. 1994.
  - [111] SOETERBOEK, R. *Predictive Control: A unified approach*. Prentice-Hall, 1992.
  - [112] SRINIVAS, M., AND PATNAIK, L. M. Genetic algorithms: A survey. *IEEE Computer* 27, 6 (1994), 17–26.
  - [113] TANG, J., AND WANG, N. Fuzzy neuron hybrid control for continuous casting. *16 th IFAC World Congress* (2005).
  - [114] THOMAS, B. G., AND HUANG, X. Effect of argon gas on fluid flow in a continuous slab casting mold. *Steel Making Conference Proceedings* (1993).
  - [115] TONAMI, Y., SEKIGUCHI, K., TSUGENO, M., TAKESAKO, T., CHEN, C. R., AND WEN, C. H. Application of a temperature control system to a new hot strip finishing mill. *Proceedings of automation in mining, mineral and metal processing* (1998), 227–232.
  - [116] WANG, H., KWONG, S., JIN, Y., WEI, W., AND MAN, K. F. Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction. *Fuzzy Sets and Systems* 149 (2005).
  - [117] WANG, P., AND KWOK, D. P. Optimal design of pid process controllers based on genetic algorithms. *Control Engineering Practice* 2, 4 (1994), 641–648.
  - [118] WIDROW, B., AND HOFF, M. Adaptive switching circuits. *Neurocomputational Foundations of Research* (1988), 123–134.
  - [119] YDSTIE, B. E. Extended Horizon Adaptive Control. *In Proc. 9th IFAC World Congress* (1984).

- [120] YOO, S. R. Development of mold level controller using sliding mode control in a continuous casting processing. *Instrumentation and Control Research Team* (1999).
- [121] ZHANG, H., CHEN, Z., LI, M., XIANG, W., AND QIN, T. *A Nonlinear Adaptive Predictive Control Algorithm Based on OFS Model*. Springer-Verlag, 2005.
- [122] ZHU, Q. M., WARWICK, K., AND DOUCE, J. L. Adaptive general predictive control for nonlinear systems. *IEE Proceedings Part D 138*, 1 (1991), 33–40.
- [123] ZHU, X., AND SEBORG, D. E. Nonlinear predictive control based on hammerstein models. *Proc. 5th International Symposium on Process Systems Engineering* (1994), 995–1000.
- [124] ZOU, Z., DEHONG, Y., HU, Z., YU, L., FENG, W., AND GUO, N. Design and simulation of nonlinear hammerstein systems dynamic matrix control algorithm. *Proceedings of 6th World Congress on Intelligent Control and Automation* (2006), 1981–1985.
- [125] ZUBEN, F. J. V. Computação evolutiva uma abordagem pragmática. *Estudos em Computação de Piracicaba e Região 1* (2000).

# Apêndice A

## Modelo NCARMA (Nonlinear Controlled Auto-Regressive Moving Average)

Este modelo, que na literatura aparece comumente como NARMAX (Nonlinear Auto-Regressive Moving Average Model with Exogenous Inputs), e representa o sistema através de uma função polinomial com grau de não-linearidade  $l$  cuja parcela determinística é apresentada como o somatório de termos com graus de não-linearidade  $m$  ( $1 \leq m \leq l$ ). Cada termo de grau  $m$  pode conter um fator de grau  $p$  do tipo  $y(t - i)$  e um fator de grau  $(m - p)$  do tipo  $u(t - i)$  multiplicados por um parâmetro  $h_{p,m-p}(n_1, \dots, n_m)$ , ou seja,

$$y(t) = \sum_{m=0}^l \sum_{p=0}^m \sum_{n_1, n_m}^{n_y, n_u} h_{p,m-p}(n_1, \dots, n_m) \prod_{i=1}^p y(t - n_i) \prod_{i=p+1}^m u(t - n_i) \quad (\text{A.1})$$

onde  $\sum_{n_1, n_m}^{n_y, n_u} \equiv \sum_{n_1=1}^{n_y} \cdots \sum_{n_m=1}^{n_u}$ , podendo ainda ser representado na forma:

$$\begin{aligned} y(t) &= h_0 + \sum_{n_1=1}^{n_y} h_{1,0}(n_1)y(t - n_1) + \sum_{n_1=1}^{n_u} h_{0,1}(n_1)u(t - n_1) \\ &+ \sum_{n_1=1}^{n_y} \sum_{n_2=n_1}^{n_y} h_{2,0}(n_1, n_2)y(t - n_1)y(t - n_2) \\ &+ \sum_{n_1=1}^{n_y} \sum_{n_2=1}^{n_u} h_{1,1}(n_1, n_2)y(t - n_1)u(t - n_2) \\ &+ \sum_{n_1=1}^{n_y} \sum_{n_2=n_1}^{n_y} h_{0,2}(n_1, n_2)u(t - n_1)u(t - n_2) + \dots \end{aligned} \quad (\text{A.2})$$

O modelo NCARMA apresenta-se como o caso mais geral de representação de sistemas não-lineares, cujos casos particulares podem representar os modelos Volterra e Bilinear, dentre outros ([35]).

### Modelo de Volterra

A representação de um sistema não-linear através de uma série de Volterra pode ser vista como uma generalização da representação de resposta impulsiva para sistemas lineares, que no seu caso discreto é dada por:

$$\begin{aligned}
y(t) = & h_0 + \sum_{i=1}^{\infty} h_{1i}u(t-d-i) + \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} h_{2ij}u(t-d-i)u(t-d-j) \\
& + \dots + \sum_{i=1}^{\infty} \dots \sum_{i=1}^{\infty} h_{mi\dots}u(t-d-i)\dots u(t-d-\dots) + \epsilon(t)
\end{aligned} \tag{A.3}$$

No caso particular de um sistema linear a equação A.3 fica reduzida ao modelo da resposta impulsiva, equação A.4. Para viabilizar a aplicação prática deste modelo utiliza-se uma série de Volterra truncada numa ordem desejada e com memória finita ([39]), isto é,

$$\begin{aligned}
y(t) = & h_0 + \sum_{i=1}^N h_{1i}u(t-d-i) + \sum_{i=1}^N \sum_{j=1}^N h_{2ij}u(t-d-i)u(t-d-j) \\
& + \dots + \sum_{i=1}^N \dots \sum_{i=1}^N h_{mi\dots}u(t-d-i)\dots u(t-d-\dots) + \epsilon(t)
\end{aligned} \tag{A.4}$$

onde os parâmetros  $h_0$ ,  $h_{1i}$  e  $h_{2ij}$  são coeficientes do modelo,  $N$  representa a memória e o número de parcelas e está relacionado, também, à ordem  $m$  do modelo. O modelo de Volterra pode ser visto como um caso particular do NCARMA onde todas as parcelas  $h_{ij}$  associadas à saída são nulas. O número de termos do modelo pode ser representado pela seguinte expressão:

$$N^{\circ}termos = \frac{(N+m)! - N!}{m!} \tag{A.5}$$

A tentativa de explicar a saída do processo utilizando apenas informação da entrada pode causar a necessidade de um grande número de parâmetros para

o modelo de Volterra ([35]). Uma alternativa a esta representação é utilizar, também, informação das saídas passadas. A representação da equação A.4 na forma paramétrica ou AR-Volterra (Auto-Regressive Volterra) torna-se ([35]).

$$A(q^{-1})y(t) = b_0 + B_1(q^{-1})u(t-d) + \sum_{i=1}^{nb} \sum_{j=1}^{nb} b_{2ij}u(t-d-i)u(t-d-j) + \dots + \epsilon(t) \quad (\text{A.6})$$

$$\text{onde: } B_1(q^{-1}) = b_{10} + b_{11}q^{-1} + \dots + b_{1nb}q^{-nb}$$

A representação de processos não-lineares por meio de séries de Volterra possibilita a descrição de dinâmicas assimétricas e variações no sinal do ganho do processo ([80]) possuindo várias aplicações bem sucedidas em controle de processos nas áreas de telecomunicações, processos químicos, sistemas biológicos, eletrônica, forno de gesso, controle de pressão ([54]; [1]; [51] e [121]).

**Modelo Bilinear** O modelo bilinear é baseado em um modelo linear do tipo ARMA mais termos não-lineares constituídos pelos produtos entre entradas e saídas na equação:

$$A(q^{-1})y(t) = b_0 + B_1(q^{-1})u(t-d) + \sum_{i=0}^{ny} \sum_{j=0}^{nu} b_{2ij}y(t-d-i)u(t-d-j) + \dots + \epsilon(t) \quad (\text{A.7})$$

onde os termos do polinômio  $b_{2ij}$  são coeficientes não-lineares,  $ny$  e  $nu$  representam o grau de não-linearidade. Também pode ser visto como um caso particular do NCARMA onde apenas os parâmetros associados à entrada e aos termos cruzados de segunda ordem são diferentes de zero.

A aplicação de um modelo bilinear na representação de um processo industrial está associada às plantas cujas características são inerentemente bilineares como processos de fermentação, colunas de destilação, reatores nucleares e reatores químicos. Como a estrutura do modelo bilinear é linear em relação aos parâmetros é possível aplicar as mesmas técnicas de identificação empregadas nos modelos lineares ([54]; [81]; [47]).

## Modelo de Wiener

Este modelo apresenta um sistema dinâmico linear seguido por um elemento não-linear, de forma contrária ao modelo de Hammerstein, como ilustrado na Figura A.1.

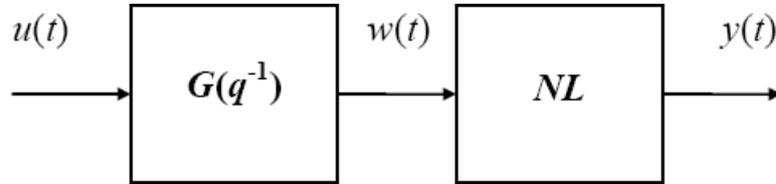


Figura A.1: Modelo de Wiener.

Da mesma forma que o caso de Hammerstein, a parcela linear pode ser representada por um dos modelos apresentados na seção 2.1 enquanto que a não-linearidade (NL) pode ser representada por um polinômio do tipo:

$$y(t) = \gamma_1 w(t) + \gamma_2 w^2(t) + \dots + \gamma_m w^m(t) \quad (\text{A.8})$$

onde  $w(t)$  é a pseudo-saída do bloco linear ou, ainda, baseada nas outras formas de representação válidas para o modelo de Hammerstein visto que o modelo de Wiener é considerado o seu dual ([35]).

O modelo de Wiener conta com diversas aplicações registradas na literatura de controle de processos como na representação do comportamento muscular sob anestesia ([79]), controle de pH ([88]), colunas de destilação ([91]), controle de válvulas ([3]), turbina a gás ([16]), reatores químicos ([4]), além de qualquer processo do Tipo Wiener, que possa ser representado por uma parcela dinâmica linear seguida de uma não-linearidade estática ([50]).

# Apêndice B

## Filtro de Kalman Linear

O filtro de Kalman (FK) foi criado por Rudolph E. Kalman em 1960 e é um algoritmo recursivo, capaz de estimar as variáveis de estado de sistemas representados por equações de estado lineares. Para tal, o mesmo utiliza equações matemáticas que implementam um estimador preditivo, buscando corrigir iterativamente a resposta de um determinado sistema através de múltiplas variáveis relacionadas a ele [24].

O FK incorpora toda informação obtida a partir de medições ruidosas para estimar as variáveis desejadas [24]. Tal característica faz com que a utilização do FK no problema em estudo seja muito útil, dado que a medição de nível do molde do lingotamento contínuo efetuada pela instrumentação de campo é comprometida pela presença do ruído ocasionado pela injeção do gás argônio. O algoritmo usado no FK é baseado na estimação das variáveis de estado de sistemas representados por equações de estado lineares, e, desse modo, deve-se considerar que o sistema sob estudo é descrito pelo seguinte modelo linear discreto:

$$\begin{cases} xe_{t+1} = \Phi_t xe_t + \Gamma_t u_t + Y_t w_t \\ y_{k+1} = H_{t+1} xe_{t+1} + v_{t+1} \end{cases} \quad (\text{B.1})$$

onde  $xe$  representa as variáveis de estado do sistema,  $u$  é a entrada,  $y$  é a saída do processo e  $w$  e  $v$  são variáveis aleatórias independentes, de média nula, e que satisfazem às seguintes relações:

$$E[w_t w_t^T] = Q_t$$

$$E[v_t v_t^T] = R_t$$

$$E[v_t w_j^T] = 0, \forall i, j$$

A variável  $w$  é comumente chamada de ruído de processo ou dinâmico, enquanto que  $v$  é conhecida como ruído de medição. Conforme pode ser observado na equação acima, o modelo propriamente dito é definido por duas equações: a primeira equação em (B.1) é a equação do processo e a segunda é a equação de medida, também conhecida por equação de observação. A equação de processo estima o estado atual através da soma de três parcelas matemáticas, onde a primeira é relacionada ao estado anterior, a segunda relacionada aos pontos de monitoramento de entrada do processo e a última parcela relacionada a um ruído de processo que simula uma alteração gradativa de seu estado ao longo do tempo. Já a equação de medida tem como objetivo associar o estado de entrada à saída do sistema através de um histórico de medidas. Para isto, é criado um novo valor de medida através da soma de uma parcela relacionada ao estado atual e de uma parcela com o ruído de medição presente no processo. O algoritmo responsável por estimar os estados do sistema no FK possui uma estrutura do tipo predição-correção (na etapa de predição diz-se que o estado estimado é um estado a priori, enquanto que na etapa de correção diz-se que o estado estimado é um estado a posteriori) Na etapa de predição, faz-se o uso do conhecimento da lei de movimento do sistema (primeira equação em (B.1) do modelo linear discreto) para estimar o estado a priori. Nesta etapa chega-se à melhor predição do estado do sistema levando em conta somente as informações de estados e entradas anteriores ao instante de tempo em questão (sem levar em conta ainda a nova medição  $y_{t+1}$ , disponível no instante em que é feita a predição). A estimação do estado a priori pode ser melhor descrita através da equação abaixo:

$$x e_{t+1}^- = \Phi_t x e_t^+ + \Gamma_t u_t \quad (\text{B.2})$$

Onde os índices sobrescritos '-' e '+' indicam, respectivamente, as predições dos estados a priori e a posteriori. Na etapa de correção faz-se a correção da estimativa propagada (estado a priori) levando em consideração a nova informação que chegou no instante atual (nova medição  $y_{t+1}$ ). Esta nova estimativa é descrita

pela equação abaixo:

$$xe_{t+1}^+ = xe_{t+1}^- + K_{t+1}[y_{t+1} - H_{t+1}xe_{t+1}^-] \quad (\text{B.3})$$

onde  $y_{t+1}$  é a medida no instante  $t + 1$  e  $H_{t+1}xe_{t+1}^-$  é o valor esperado para a saída do modelo neste instante, utilizando informações até o instante  $t$ . A matriz  $K_{t+1}$  é uma matriz de ganhos (conhecidos como ganhos de Kalman) que deve ser obtida com o objetivo de minimizar o valor do traço da matriz de covariância  $P_{t+1}^+$  da estimativa do estado feito pelo filtro de Kalman.

Sendo assim, considerando as duas etapas citadas acima, pode-se definir uma sequência de cálculos recursivos que constituem o algoritmo utilizado pelo filtro de Kalman:

$$\left\{ \begin{array}{l} xe_{t+1}^- = \Phi_t xe_t^+ + \Gamma_t u_t \\ P_{t+1}^- = \Phi_t P_t^+ \Phi_t^T + Y_t Q_t Y_t^T \\ K_{t+1} = P_{t+1}^- H_{t+1}^T [H_{t+1} P_{t+1}^- H_{t+1}^T + R_{t+1}]^{-1} \\ xe_{t+1}^+ = xe_{t+1}^- + K_{t+1} [y_{t+1} - H_{t+1} xe_{t+1}^-] \\ P_{t+1}^+ = P_{t+1}^- - K_{t+1} H_{t+1} P_{t+1}^- \end{array} \right.$$

### Filtro de Kalman Estendido

Como a grande maioria dos processos da natureza são modelados por funções não lineares, o filtro de Kalman original (que estima variáveis de estado de sistemas representados por equações de estado lineares) precisou sofrer algumas adaptações no seu conceito, de modo a permitir a generalização destas aplicações a sistemas não-lineares que possuem a seguinte forma:

$$\begin{cases} xe_{t+1} = f(xe_t, u_t, w_t) \\ y_{t+1} = h(xe_{t+1} + v_{t+1}) \end{cases}$$

onde  $f$  e  $h$  são funções não-lineares.

Uma solução encontrada para este problema consiste em linearizar analiticamente o sistema em torno do atual estado e posteriormente aplicar as equações do filtro de Kalman [24]. Esta solução é conhecida como Filtro de Kalman Estendido

(FKE). Para a linearização, faz-se uso de expansões em série de Taylor para linearizar a estimativa de um dado em torno da estimativa atual, utilizando as derivadas parciais das funções de processo e medida, mesmo com relações não lineares. Desse modo, o FKE nada mais é do que o filtro de Kalman linear implementado utilizando as matrizes jacobianas de  $f$  e  $h$  [24]. Como as funções  $f$  e  $h$  podem ser vetoriais, será indicado por  $f_i$  a  $i$ -ésima componente de  $f$ . Sendo assim, a matriz jacobiana de  $f$  é definida como:

$$Df(xe) = \begin{bmatrix} \frac{\partial f_1}{\partial xe_1} & \frac{\partial f_1}{\partial xe_2} & \cdots & \frac{\partial f_1}{\partial xe_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial xe_1} & \frac{\partial f_n}{\partial xe_2} & \cdots & \frac{\partial f_n}{\partial xe_n} \end{bmatrix} \quad (\text{B.4})$$

Do mesmo modo podem ser definidas as matrizes  $Df(w)$  (matriz análoga, mas relativa ao vetor de ruído de processo) e  $Dh(x)$ :

$$Df(w) = \begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \frac{\partial f_1}{\partial w_2} & \cdots & \frac{\partial f_1}{\partial w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial w_1} & \frac{\partial f_n}{\partial w_2} & \cdots & \frac{\partial f_n}{\partial w_n} \end{bmatrix} \quad (\text{B.5})$$

$$Dh(x) = \begin{bmatrix} \frac{\partial h_1}{\partial xe_1} & \frac{\partial h_1}{\partial xe_2} & \cdots & \frac{\partial h_1}{\partial xe_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial xe_1} & \frac{\partial h_n}{\partial xe_2} & \cdots & \frac{\partial h_n}{\partial xe_n} \end{bmatrix} \quad (\text{B.6})$$

De posse das matrizes jacobianas mostradas acima, pode-se definir o seguinte algoritmo para estimação do FKE:

$$\left\{ \begin{array}{l} xe_{t+1}^- = f(xe_t^+, u_t) \\ P_{t+1}^- = Df(xe_t)P_t^+ Df(xe_t)^T + Df(w_k)Q_k Df(w_k)^T \\ K_{t+1} = P_{t+1}^- Dh(xe_{t+1})^T [Dh(xe_{t+1})Dh(xe_{t+1})^T + Dh(xe_{t+1})R_{t+1}Dh(xe_{t+1})^T]^{-1} \\ xe_{t+1}^+ = xe_{t+1}^- + K_{t+1}[y_{t+1} - Dh(xe_{t+1}^-)] \\ P_{t+1}^+ = P_{t+1}^- - K_{t+1}Dh(xe_{t+1})P_{t+1}^- \end{array} \right.$$