Métodos de Identificação de Sistemas Auxiliados por Computador

Washington Luiz de Albuquerque Silva

Dissertação de Mestrado submetida à Coordenação dos Cursos de Pós-Gradução em Engenharia Elétrica da Universidade Federal da Paraíba - Campus II como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Antonio Marcus Nogueira Lima, Dr.
Orientador
Péricles Rezende Barros, Ph.D.
Orientador

Campina Grande, Paraíba, Brasil

S586m Silva, Washington Luiz de Albuquerque
Metodos de identificacao de sistemas auxiliados por computador / Washington Luiz de Albuquerque Silva,. - Campina Grande, 1997.
173 f. : il.

Dissertacao (Mestrado em Engenharia Eletrica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Sistemas Auxiliados por Computados 2. Processamento da Informacao 3. Dissertacao I. Lima, Antonio Marcus Nogueira, Dr. II. Barros, Pericles Rezende. Dr. III. Título

CDU 621.391(043)

MÉTODOS DE IDENTIFICAÇÃO DE SISTEMAS AUXILIADOS POR COMPUTADOR

WASHINGTON LUIZ DE ALBUQUERQUE SILVA

Dissertação Aprovada em 09.06.1997

PROF. ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFPB
Orientador

PROF. PÉRICLES REZENDE BARROS, Ph.D., UFPB
Orientador

PROF. CURSINO BRANDÃO JACOBINA , Dr.Ing., UFPB
Componente da Banca

PROF. CLIVALDO SILVA DE ARAÚJO, D.Sc., UFPB
Componente da Banca

CAMPINA GRANDE - PB
Junho - 1997

Dedicatória

Aos meus pais Luiz e Angelita, por três décadas de amor, dedicação e incentivo.

Aos engenheiros Roberto, Cezar e Júnior, meus fraternais companheiros, às respectivas, e à sobrinha Luize, a primeira da numerosa e talentosa geração que há de vir.

À minha namorada Ana Paula, por sua compreensão, apoio e carinho.

Agradecimentos

Ao professor Antônio Marcus N. Lima, pela intensiva e dedicada orientação, sem a qual não seria possível a conclusão deste trabalho e ao professor Péricles R. Barros, pela confiança, pelo incentivo na retomada dos trabalhos na segunda fase de sua realização e pela orientação.

À CAPES que financiou a realização da primeira fase deste trabalho.

Aos amigos Alberto, Hélcio, Vicente Lucena, Giovanni Barroso, Yuri, Geovany Borges e Gláucio Brandão pelo apoio, sugestões e contribuições para a lapidação deste trabalho.

Aos professores do grupo de Instrumentação e Controle, G.S.Deep, José Sérgio, Angelo e Freire, pelo apoio e colaboração.

Aos colegas que ajudaram no desenvolvimento e na conclusão deste trabalho.

Às senhoras Alaíde (*in memorium*), Regina, Ignês e familiares que carinhosamente me acolheram em seus lares, ao longo deste trabalho e de toda a minha formação profissional.

Ao irmão de fé Giovanni L. C. de Melo e família, amigos eternos.

Resumo

O presente trabalho trata da identificação de sistemas baseada nos sinais de entrada e saída do processo. Além de algorítmos de estimação conhecidos, são tratados o problema da estimação dos parâmetros de um modelo com a utilização da decomposição em valores singulares (*SVD - Singular Value Decomposition*), e a redução de ordem do modelo estimado com a utilização de *SVD*.

Neste trabalho é apresentado um pacote de rotinas com interface amigável para o ambiente *MATLAB*, o *ISAC* - **I**dentificação de **S**istemas **A**uxiliada por **C**omputador, que permite a realização de experimentos para a identificação de processos SISO, como também o projeto de controladores PID. Os métodos disponíveis são o dos mínimos quadrados, o das variáveis instrumentais, um método de estimação recursiva utilizando *SVD* e um método com redução de modelos baseada em *SVD*. O pacote é composto de rotinas compactas que utilizam os toolboxes básicos do *MATLAB*. A faixa de aplicação deste pacote de rotinas é bastante ampla, pois qualquer processo SISO com características dinâmicas representa uma aplicação em potencial.

São apresentados os resultados da comparação entre os métodos implementados e os resultados da aplicação do pacote ISAC a um protótipo de um secador de grãos industrial e a modelos simulados com o próprio programa.

Abstract

The main goal of this work was to implement some identification methods, with emphasis on the ones which use singular value decomposition. The readers will find the least square method, the instrumental variable method, a method based on the singular value decomposition of the process response and also a model order reduction method. Some results of the application of these methods are presented at the end of this work.

The algorithms were grouped in a toolbox, named *ISAC* (Computer Aided System Identification). The *ISAC* has a friendly interface, and so is not necessary to the user know the *MATLAB* language which is based on line commands. The toolbox has also some functions to help the user to find a solution to his modeling problems, like the frequency tools. The user can also design PID controlers with this toolbox. There is a model order reduction method based on the singular value decomposition *SVD* of the model impulse response.

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Símbolos e Abreviaturas

$\mathbf{x} \in \Re^n$	x pertence ao conjunto dos números reais de dimensão n.
$\mathbf{A} \in \Re_r^{m \times n}$	**A** pertence ao conjunto de matrizes reais, de m linhas, n colunas e posto r.
$\mathbf{A}^{-1}$	Inversa da matriz **A**.
$\mathbf{A}^T$	Transposta da matriz **A**.
$\mathbf{A}^\perp$	Matriz ortogonal.
$\mathbf{Z}^N \Longrightarrow \hat{\boldsymbol{\theta}}_N$	Mapeamento do conjunto de dados para o vetor de parâmetros estimados.
$N \rightarrow \infty$	N tende para ∞.
$\frac{d}{dt} f(t)$	Derivada da função $f(t)$ em relação ao tempo.
$D_{\mathcal{M}}$	Conjunto de modelos com a estrutura $\mathcal{M}$.
$\mathbf{Z}^N$	Conjunto de dados do processo, de comprimento N.
$\|\mathbf{A}\|$	Norma de **A**.
$\|\mathbf{A}\|_2$	Norma-2 de **A**.
$\|\mathbf{A}\|_F$	Norma de Frobenius de **A**.
$\mathbf{X} \subseteq \mathbf{Y}$	**X** é um subespaço de **Y**.
$\rho(\mathbf{A})$	Posto da matriz **A**.
$\mathcal{R}(\mathbf{A})$	Espaço imagem de **A**.
$\mathcal{N}(\mathbf{A})$	Espaço nulo de **A**.
$\forall \omega$	Qualquer que seja o ω.
$\Phi_u(\omega)$	Espectro de potência do sinal .
ε	Erro de predição.
$\ell(\varepsilon)$	Função de custo para avaliar o erro de predição ε.
$R_u(\tau)$	Função de covariância de u.
$u(t), y(t)$	Sinal de entrada e saída do processo, respectivamente.
$\boldsymbol{\varphi}(t)$	Vetor de regressão, composto pelos dados do processo.
$E\, f(t)$	Valor esperado da função $f(t)$.
$\overline{E}\, f(t)$	$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^{N} E\, f(t)$.
$\arg\min f(x)$	Argumento x que minimiza a função $f(x)$.

$\widehat{y}(t)$	Valor estimado de $y(t)$.
$\widehat{y}(t \mid \boldsymbol{\theta})$	Valor estimado de $y(t)$ com base no vetor de parâmetros $\boldsymbol{\theta}$.
$\boldsymbol{\theta}(t)$	Vetor de parâmetros estimados.
$V_N\left(\boldsymbol{\theta}, \mathbf{Z}^N\right)$	Critério de minimização dos mínimos quadrados.
$Q_{\boldsymbol{\theta}}, Q_X$	Funções que definem os fatores de correção para $\widehat{\boldsymbol{\theta}}_t$ e $X(t)$.
$\widehat{\boldsymbol{\theta}}_N^{LS}$	Vetor de parâmetros estimados pelo método dos mínimos quadrados.
$\widehat{\boldsymbol{\theta}}_N^{IV}$	Vetor de parâmetros estimados pelo método das variáveis instrumentais.
$\boldsymbol{\zeta}$	Variável instrumental.
$\beta(N,t)$	Função de custo.
$SISO$	Sistema com uma entrada e uma saída.
ARX	Modelo de identificação Autoregressivo com entrada extra.
$ARMAX$	Modelo de identif. Autoregressivo com entrada extra e média móvel.
OE	Modelo de identificação Erro na saída.
RLS	Método dos mínimos quadrados recursivo.
LS	Método dos mínimos quadrados.
IV	Método das variáveis instrumentais.
$PRBS$	Sinal binário pseudo-aleatório.
SVD	Decomposição em Valores Singulares.
ZOH	zero-order-hold.
FIR	Resposta finita ao impulso.
$ISAC$	Identificação de Sistemas Auxiliada por Computador.
$LIEC$	Laboratório de Instrumentação Eletrônica e Controle
PID	Controlador Proporcional-Integral-Derivativo.
GUI	Interface Gráfica com o Usuário.

Capítulo 1

Introdução

A modelagem das características dinâmicas do processo baseada em dados experimentais é importante para o projeto de sistemas de controle. De modo geral, as técnicas de controle que utilizam um modelo do processo como ponto de partida para o projeto do controlador não garantem o desempenho desejado se o modelo escolhido não for adequado [1]. Esta regra se aplica tanto aos métodos de controle clássicos baseados em modelos da planta, quanto aos métodos mais recentes como as técnicas de controle adaptativo.

O projeto de controladores visa a garantia de um desempenho satisfatório, sob determinados aspectos relevantes para o processo em questão [2] [3]. O controlador deve ser implementado de forma a obter-se estabilidade e desempenho satisfatórios, que inclui, seguir um sinal de referência desejado, atenuar os distúrbios, melhorar a eficiência e precisão do sistema e diminuir a energia consumida pelo processo. Ainda assim, na prática, os parâmetros dos sistemas tendem a modificar-se devido às variações do ambiente onde a planta está instalada, alterações nas condições de operação do processo e ação de atuadores não lineares. Quando tais variações paramétricas são pequenas, algumas das técnicas de controle existentes são capazes de suportá-las e ainda manter o desempenho desejado. Porém, se esta variação for significativa, torna-se necessária a busca por uma solução mais robusta.

O princípio do projeto em controle robusto é garantir estabilidade e desempenho satisfatórios para uma família de plantas possíveis e uma família de perturbações possíveis, ao invés da teoria convencional de controle linear em que um modelo e distúrbio específicos são considerados. Sendo assim, a aplicação desta técnica exige que sejam conhecidos modelos nominais para a planta e a faixa de variação da perturbação. Quanto mais próximos forem estes modelos nominais dos valores que a planta e distúrbio podem assumir, melhor será o desempenho do sistema [4]. Deste

modo, devem ser utilizados modelos nominais representativos para o processo.

Nas técnicas de controle adaptativo a planta é modelada a cada instante e o controlador tem seus parâmetros atualizados de acordo com as variações dos parâmetros da planta. Com isto obtem-se um método que visa garantir um desempenho adequado, mesmo com variações consideráveis no comportamento da planta [5].

A Identificação de Sistemas trata do problema da construção de modelos matemáticos de sistemas dinâmicos baseada na observação dos sinais de entrada e saída do sistema, e assim pode ser considerada como sendo o primeiro passo na solução de problemas de controle. Os modelos de sistemas dinâmicos constituem-se em instrumentos de grande utilidade para muitos propósitos: predição, controle, simulação, projeto de filtros, reconstrução de medições, entre outros [6] [7] [8]. Com as técnicas de identificação de sistemas pode-se construir *ferramentas versáteis* para a solução de diversos problemas na engenharia. Torna-se evidente o seu valor pelas numerosas aplicações nos mais diversos campos. Algumas aplicações podem também ser encontradas em [5], [9], [6], [10], [11], [12], [13] e [14].

Atualmente, para desempenhar a tarefa de modelagem paramétrica de sistemas, são desenvolvidos algorítmos para computadores digitais. De um modo geral os algorítmos mais conhecidos possuem alguns problemas que ainda estão em via da solução. O método dos mínimos quadrados, por exemplo, pode apresentar uma estimação pobre para o processo quando este é atacado por um ruído correlacionado com o vetor de regressores, ou quando o sinal de teste do processo não é suficiente para excitar todos os modos da planta [9] [15]. No último caso, e quando a implementação em tempo real do método requer a forma recursiva, esta poderá apresentar problemas na convergência dos parâmetros. A razão para tal é um *mal condicionamento* da matriz de covariância, causado pela *baixa persistência* do sinal de excitação. Quando o sinal de excitação não é rico em frequências o suficiente para excitar a planta no ponto de operação, a velocidade de convergência dos parâmetros do modelo, para o método dos mínimos quadrados, apresenta-se lenta. De um modo geral, as aplicações em que a velocidade de convergência dos parâmetros é fundamental, são candidatas a utilização de técnicas com convergência paramétrica mais veloz. A utilização de ferramentas com a capacidade de extrair mais informações sobre o processo, utilizando-se sinais que não perturbem demasiadamente o processo em questão. A decomposição em valores singulares (SVD) apresenta-se como uma ferramenta que pode ser utilizada para esta finalidade.

O objetivo do presente trabalho é fornecer uma contribuição para usuários de identificação de sistemas, provendo um estudo e implementação de algorítmos e rotinas de cálculo selecionados. São também abordados, métodos de modelagem que utilizam

decomposição em valores singulares, uma ferramenta matemática adequada para lidar com matrizes *mal condicionadas.*

Neste trabalho, é estudado o emprego de uma técnica que utiliza a decomposição em valores singulares (*SVD - **S**ingular **V**alue **D**ecomposition*) tanto na estimação de parâmetros quanto na redução da ordem do modelo estimado e no cálculo de uma *realização de ordem mínima* do sistema. Os algorítmos para SVD possuem comportamento estável e robusto e a sua utilização em métodos de estimação paramétrica comprova o seu potencial na obtenção de modelos para o processo em estudo.

Na área de instrumentação e controle, um dos principais objetivos da estimação paramétrica é gerar um modelo que permita o projeto do controlador. Neste sentido, também é objetivo deste trabalho avaliar a qualidade da estimação através do desempenho obtido em malha fechada quando projeta-se o controlador com o modelo estimado. Nesta avaliação considera-se tanto o projeto de controladores fixos quanto o projeto de controladores adaptativos. Para isto, este trabalho inclui tanto a análise teórica dos algorítmos, quanto a sua implementação e teste a nível de simulação digital. A meta é que as técnicas estudadas e implementadas sejam integradas num pacote denominado ISAC - **I**dentificação e **C**ontrole de **S**istemas **A**uxiliada por **C**omputador, desenvolvido com os recursos de programação do *MATLAB.* O pacote poupa esforços do usuário, que tem a sua disposição uma ferramenta com *interface* gráfica amigável.

Vale destacar que existem comercialmente vários pacotes de programas que podem ser utilizados na estimação paramétrica. Alguns destes pacotes foram analisados em [16], [17]. O *System Identification Toolbox* desenvolvido por L. Ljung, para o *MATLAB*, pode ser apreciado em [18]. Neste contexto, o desenvolvimento do ISAC não objetiva competir com tais produtos comerciais, mas sim disponibilizar de forma operacional os resultados obtidos neste trabalho, ao mesmo tempo que permite-nos dominar a utilização das modernas ferramentas de interface gráfica com o usuário do *MATLAB.*

1.1 Organização do Trabalho

O presente trabalho está organizado como descrito a seguir.

No capítulo 2, são apresentadas ferramentas matemáticas, aspectos da modelagem de plantas e identificação de sistemas. Os métodos de identificação analisados são, o dos mínimos quadrados e o método das variáveis instrumentais. A teoria de modelagem de processos é apresentadas na secção 2.1, assim como as classificações dos métodos de identificação quanto à implementação e à execução. Aspectos básicos dos principais métodos de estimação paramétrica são abordados na secção 2.2, o método

dos mínimos quadrados é apresentado na secção 2.2.2, o método de estimação das Variáveis Instrumentais (IV) é apresentada na secção 2.2.3.

No capítulo 3 são apresentadas definições e teoremas relacionados à Decomposição em Valores Singulares, como também uma técnica de redução de ordem de modelos. Apresentam-se definições introdutórias de teorias de matrizes, normas matriciais e ortogonalidade, na secção 3.1. A ferramenta matemática conhecida como decomposição em valores singulares (*SVD*) é apresentada brevemente na secção 3.2. Um método de estimação que utiliza SVD para atualização da matriz de covariância, foi analisado e implementado, e é apresentado na secção 3.3. Na secção 3.4 apresenta-se um método de identificação que inclui a redução de modelos utilizando-se *SVD*.

No capítulo 4 é apresentada a teoria básica do projeto de controladores. Uma técnica de projeto de controladores PID utilizando a técnica de alocação de pólos, é apresentada na secção 4.1.

No capítulo 5 são apresentados a descrição de funcionamento e os recursos do ISAC, discutindo-se os métodos aplicados, problemas enfrentados e soluções encontradas. Na seção 5.1, é apresentada uma descrição geral do pacote de rotinas implementado. Na seção 5.2, é apresentada a janela principal do programa. Nas seções 5.3 e 5.4, são apresentadas respectivamente a tela de edição da forma de onda do sinal de excitação e a função de transferência do processo a ser simulado. Na seção 5.5, são apresentadas as rotinas da janela de filtragem dos dados. Os métodos de estimação implementados e a janela implementada são apresentados na seção 5.6. Na seção 5.7, são apresentadas as telas de redução da ordem do modelo estimado e a janela para projeto do controlador. As ferramentas de análise em frequência presentes no ISAC são mencionadas na seção 5.6. E finalmente, na seção 5.9, são feitos comentários conclusivos sobre a ferramenta implementada.

No capítulo 6 são apresentados os resultados de testes experimentais e comparações com os algorítmos estudados nos capítulos precedentes. São comentados o desempenho dos algorítmos e forma de implementação adotada. Neste o leitor pode apreciar os gráficos gerados com a implementação de rotinas para o *MATLAB*, e verificar os resultados e conclusões obtidos. Os modelos utilizados são apresentados na secção 6.1, juntamente com as condições de teste, informações sobre as plantas utilizadas, funções de transferência e resposta ao degrau. Foram testados algorítmos em malha aberta utilizando-se uma função de custo e projeto de controlador PID pelo método de alocação de pólos para a validação dos modelos encontrados. Na seção 6.2, são apresentados os resultados com o método de Zhang, visto na seção 3.4. São apresentadas ainda, na seção 6.3, algumas telas com exemplos de aplicações do ISAC em sistemas

reais montados em laboratório. Ainda no capítulo 6 é apresentada a aplicação do ISAC para o projeto de controladores PID.

No capítulo 7 são apresentadas as conclusões finais do trabalho e sugestões para trabalhos posteriores.

Capítulo 2

Modelagem de Sistemas

Neste capítulo são apresentadas algumas definições da teoria de modelagem de sistemas e estruturas de modelos utilizadas em identificação de sistemas. Os métodos de identificação dos mínimos quadrados convencional não-recursivo e recursivo são apresentados, como também o método das variáveis instrumentais.

2.1 Modelos de Processos

Os modelos referidos neste trabalho, são equações matemáticas que visam à uma representação de um processo físico real. De um modo geral, quanto mais complexa for a equação matemática associada ao sistema, mais próxima do real será a representação do sistema físico. Porém, na maioria dos casos, a simplificação destes modelos além de facilitar bastante a tarefa de identificação do processo, não causam problemas para as aplicações nas quais estes serão utilizados. Para fins de controle, aproximações desta natureza são suficientes na maioria dos casos.

Foram definidas algumas classes de modelos para identificação que estão muito relacionadas com os procedimentos de identificação típicos. No *modelo caixa branca*, ou modelagem física, o modelo é obtido pela utilização de leis físicas. Há também o modelo *caixa cinza*, no qual as características do modelo físico são encontradas a partir de dados medidos e utilizando informações *a priori* do sistema. Já no modelo *caixa preta*, ou de identificação, o modelo é construido baseando-se em dados de entrada e saída adquiridos do sistema [9] [4]. Neste trabalho foram utilizados os modelos do tipo *caixa preta*.

2.1.1 Classe de Modelos

A classe de modelos considerada neste trabalho é definida a partir da equação (2.1), que representa um sistema linear invariante e discreto no tempo.

$$y(t) = \sum_{k=1}^{\infty} g(k)\, u(t-k), \qquad t = 0, 1, 2, ... \tag{2.1}$$

Na equação (2.1), $g(k)$ é a resposta ao impulso do sistema e $u(t)$ é o sinal de entrada do sistema [9].

Utilizando o operador de deslocamento unitário por atraso, q^{-1}, pode-se obter uma representação mais compacta para a equação (2.1), isto é,

$$y(t) = \sum_{k=1}^{\infty} g(k)\, u(t-k) = \sum_{k=1}^{\infty} g(k) \left(q^{-k} u(t)\right) \tag{2.2}$$

$$= \left[\sum_{k=1}^{\infty} g(k) q^{-k}\right] u(t) = G(q)\, u(t). \tag{2.3}$$

Na equação (2.3), $G(q)$ é denominada de função de transferência do sistema [9]. Admitindo a presença de um termo de perturbação ou ruído aditivo, $v(t)$,

$$v(t) = H(q)\, e(t), \tag{2.4}$$

com

$$H(q) = \sum_{k=1}^{\infty} h(k)\, q^{-k}, \tag{2.5}$$

tem-se então que,

$$y(t) = G(q)\, u(t) + v(t) = G(q)\, u(t) + H(q)\, e(t), \tag{2.6}$$

na qual, $e(t)$ é uma sequência de variáveis aleatórias independentes com média zero e variância λ_e.

Um caso particular porém de muita aplicação da equação (2.6), é a equação (2.7), que representa um refinamento da classe de modelos definida anteriormente,

$$y(t) + a_1 y(t-1) + ... + a_{n_a} y(t-n_a) = b_1 u(t-1) + ... + b_{n_b} u(t-n_b) + e(t). \tag{2.7}$$

Nesta equação, o termo aditivo $e(t)$ representa um ruído branco que entra direto na equação a diferenças. Por causa deste termo a equação é comumente conhecida como *modelo de erro na equação*. Utilizando a notação de operador deslocamento, pode-se ainda, escrever esta equação como:

$$y(t) = G(q, \boldsymbol{\theta})\; u(t) + H(q, \boldsymbol{\theta})\; e(t), \tag{2.8}$$

com $G(q,\boldsymbol{\theta})$ e $H(q,\boldsymbol{\theta})$ dados por,

$$G(q,\boldsymbol{\theta}) = \frac{B(q)}{A(q)} \qquad \text{e} \qquad H(q,\boldsymbol{\theta}) = \frac{1}{A(q)},$$

em que,

$$A(q) = 1 + a_1 q^{-1} + ... + a_{n_a} q^{-n_a} \tag{2.9}$$

$$B(q) = b_1 q^{-1} + ... + b_{n_b} q^{-n_b} \tag{2.10}$$

e o vetor de parâmetros $\boldsymbol{\theta}$ pode ser definido como,

$$\boldsymbol{\theta} = \left[\begin{array}{cccccc} a_1 & ... & a_{n_a} & b_1 & ... & b_{n_b} \end{array} \right]^T.$$

O modelo definido pela equação (2.8) é conhecido como ARX (Figura 2.1), na qual o termo AR se refere à parte autoregressiva (**Auto**Regressive) $A(q)\,y(t)$, e X à entrada extra (e**X**tra) $B(q)\,u(t)$.

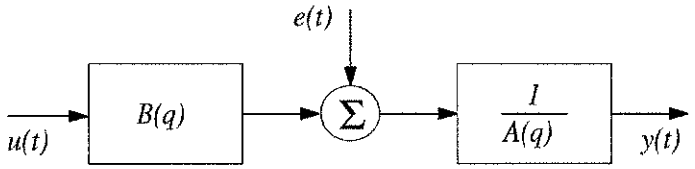

Figura 2.1: Modelo ARX

No modelo (2.7) não há liberdade na descrição das propriedades do termo $e(t)$. Pode-se então adicionar flexibilidade ao modelo, escrevendo a equação (2.7) com representação do distúrbio como um ruído branco com média móvel. Isto leva ao modelo:

$$\begin{aligned} y(t) + a_1 y(t-1) + \cdots + a_{n_a} y(t-n_a) &= b_1 u(t-1) + \cdots + b_{n_b} u(t-n_b) + e(t) + \\ &\quad c_1 e(t-1) + \cdots + c_{n_c} e(t-n_c), \end{aligned} \tag{2.11}$$

Definindo-se, $C(q) = 1 + c_1\, q^{-1} + \cdots + c_{n_c}\, q^{-n_c}$, tem-se o modelo na forma compacta,

$$A(q)\,y(t) = B(q)\,u(t) + C(q)\,e(t). \tag{2.12}$$

Este modelo é conhecido como ARMAX (Figura 2.2) devido ao ruído com média móvel (MA - **M**oving **A**verage) $C(q)\,e(t)$ que foi adicionado.

Percebe-se que, no modelo ARMAX, o caso em que $C(q) \equiv 1$,

$$A(q)\,y(t) = B(q)\,u(t) + e(t),$$

leva ao modelo ARX.

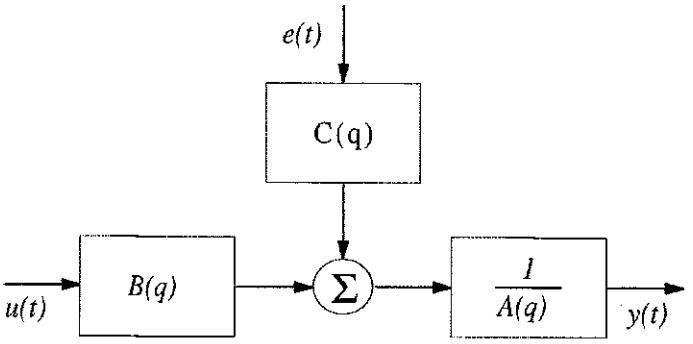

Figura 2.2: Modelo ARMAX

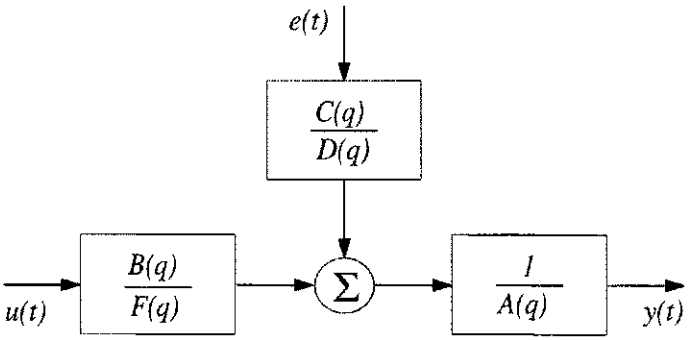

Figura 2.3: Modelo de estimação genérico

Pode-se então definir uma estrutura de modelos generalizada (Figura 2.3):

$$A(q)\,y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t). \tag{2.13}$$

Assim, por exemplo, para o modelo ARMAX, utiliza-se o mesmo denominador para $G(q,\boldsymbol{\theta})$ e $H(q,\boldsymbol{\theta})$:

$$F(q) = D(q) = A(q) = 1 + a_1 q^{-1} + ... + a_{n_a} q^{-n_a}. \tag{2.14}$$

No modelo conhecido por OE (**O**utput **E**rror) os polinômios $C(q)$ e $D(q)$ são dados por,

$$C(q) = D(q) = 1. \tag{2.15}$$

Estes são alguns dos possíveis modelos para o processo em estudo. O leitor que desejar um estudo mais aprofundado destas classes de modelos pode consultar, por exemplo Ljung [9] [19] [20]. A escolha do modelo é uma das tarefas mais importantes e difíceis na identificação de sistemas. Os modelos mais comuns são as descrições da equação diferença, como modelos ARX e ARMAX apresentados acima, e também os modelos lineares no espaço de estados, conforme apresentado por Ljung, [19]. Neste trabalho foram utilizados os modelos ARX e ARMAX para teste dos algorítmos que

utilizam SVD em sua estrutura. Os programas apresentados no final deste trabalho podem ser facilmente modificados para simular o modelo OE.

2.2 Identificação de Sistemas

A identificação de sistemas é constituída de procedimentos matemáticos, aplicados a um conjunto de dados, com o objetivo de obter as características de um determinado modelo do sistema dinâmico que gerou os dados. Entende-se por conjunto de dados os sinais de entrada e saída do sistema $u(t)$ e $y(t)$, respectivamente.

2.2.1 Classificação dos Métodos de Identificação

Os métodos de identificação podem ser **paramétricos** ou **não-paramétricos.** Nos métodos **paramétricos** toma-se por base um conjunto de possíveis modelos, ou seja, a identificação é feita a partir de descrições matemáticas pré-determinadas. Nos métodos de identificação **não-paramétricos** não se emprega, explicitamente um modelo paramétrico conhecido, na busca de uma melhor descrição para o processo, ou seja, determina-se a função de transferência diretamente da relação entrada/saída, sem necessitar da seleção prévia de um conjunto restrito de modelos prováveis. Por outro lado, os métodos de identificação paramétricos utilizam um vetor de parâmetros finito na busca de um modelo representativo.

Na estimação paramétrica o algorítmo do estimador monta um modelo matemático do sistema utilizando os sinais de entrada e saída do processo. O diagrama de uma estimação paramétrica pode ser visto na Figura 2.4. Com base nos dados $u(t)$ e $y(t)$, o estimador monta um modelo para a planta. A diferença entre a saída da planta, $y(t)$, e a saída do modelo, $\hat{y}(t)$, para um mesmo sinal de entrada, $u(t)$, é chamada de *erro de predição* e é utilizada na busca de um modelo que represente a planta. Na próxima secção, o *erro de predição* será apresentado de uma maneira formal.

Quanto à implementação pode-se classificá-los como método de identificação *em tempo real* ou método de identificação *em lote*. Na implementação *em tempo real* o processo é identificado em condições normais de funcionamento ou seja, com a planta no seu ambiente natural de trabalho e com ruídos reais característicos do ambiente. Na implementação *em lote*, a planta é submetida a testes fora de seu ambiente normal de trabalho.

No que se refere à implementação os algorítmos podem ser chamados de **recursivos** ou **não-recursivos**. Nos algorítmos **recursivos** o vetor paramétrico é construido progressivamente à medida que os dados são coletados. Nos métodos **não-recursivos**,

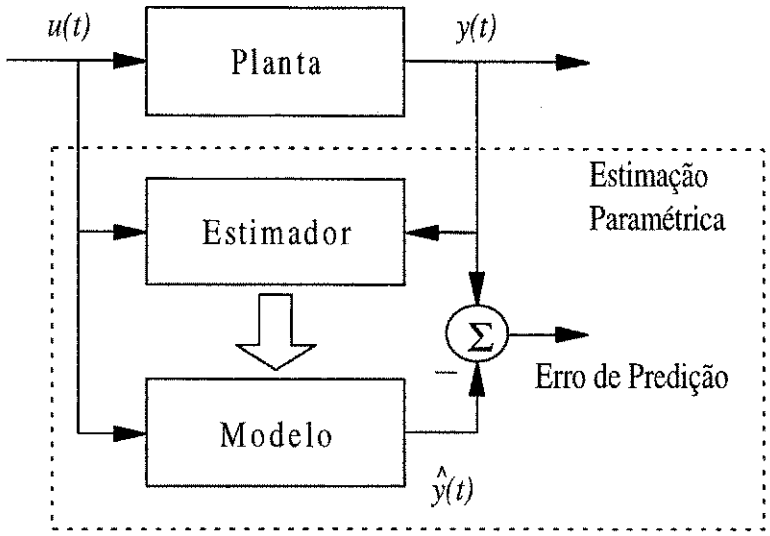

Figura 2.4: Estimação Paramétrica

a determinação dos parâmetros é feita a partir de um lote de dados, ou seja, necessita-se de um bloco de dados para os cálculos dos parâmetros.

Neste trabalho foram estudadas técnicas paramétricas recursivas e não-recursivas. Os métodos não-paramétricos não foram estudados neste trabalho. O estudo das técnicas incluiu simulações e implementações de algorítmos tanto em tempo real como em processamento de lotes de dados.

2.2.2 Estimação pelo Método dos Mínimos Quadrados

O método dos mínimos quadrados é um dos mais básicos métodos de estimação paramétrica e é apresentado logo a seguir.

Definindo-se um vetor $\varphi(t)$, com os elementos da equação (2.7), tem-se,

$$\varphi(t) = \left[\begin{array}{cccccccc} -y(t-1) & -y(t-2) & \cdots & -y(t-n_a) & u(t-1) & \cdots & u(t-n_b) \end{array} \right]^T, \tag{2.16}$$

e definindo-se um vetor de parâmetros θ como,

$$\boldsymbol{\theta} = \left[\begin{array}{cccccccc} a_1 & a_2 & \cdots & a_{n_a} & b_1 & \cdots & b_{n_b} \end{array} \right]^T, \tag{2.17}$$

a expressão para o preditor de um passo a frente para a expressão (2.1) é dada por

$$\hat{y}(t|\boldsymbol{\theta}) = B(q)\,u(t) + [1 - A(q)]\,y(t). \tag{2.18}$$

Os métodos de estimação paramétricos arranjam a saída estimada do modelo como uma função linear do vetor de parâmetros $\boldsymbol{\theta}$. Existe então um vetor composto pelas

variáveis medidas conhecido como vetor de regressão $\varphi(t)$, definido na equação (2.16), de forma que a saída estimada pode ser escrita como

$$\hat{y}(t|\boldsymbol{\theta}) = \varphi^T(t)\,\boldsymbol{\theta}. \tag{2.19}$$

Nos métodos de estimação paramétricos, o erro de predição é dado por

$$\begin{aligned} \varepsilon(t,\boldsymbol{\theta}) &= y(t) - \hat{y}(t|\boldsymbol{\theta}) && (2.20)\\ &= y(t) - \varphi^T(t)\,\boldsymbol{\theta}. && (2.21) \end{aligned}$$

A equação (2.20) estabelece uma forma de se verificar a qualidade do modelo estimado. Assim, um bom modelo é aquele que apresenta os menores valores do erro de predição. Os métodos de estimação nos quais minimiza-se o erro de predição, são conhecidos como *métodos de estimação por erro de predição* [9] [21].

Neste contexto, um método de estimação paramétrica é o mapeamento do conjunto de dados da planta,

$$\mathbf{Z}^N \Longrightarrow \hat{\boldsymbol{\theta}}_N \in D_{\mathcal{M}},$$

em que $D_{\mathcal{M}}$ é um conjunto de modelos com a estrutura $\mathcal{M}$.

É evidente que um bom modelo é aquele que produz valores pequenos de erro de predição. Para se avaliar a dimensão do erro de predição pode-se escolher a função de custo $\ell(\varepsilon)$ como sendo

$$\ell(\varepsilon) = \frac{1}{2}\varepsilon^2. \tag{2.22}$$

Utilizando a seguinte norma,

$$V_N\left(\boldsymbol{\theta}, Z^N\right) = \frac{1}{N}\sum_{t=1}^{N}\ell\left(\varepsilon(t,\boldsymbol{\theta})\right). \tag{2.23}$$

E com a função de custo (2.22), a equação (2.23) pode ser escrita como,

$$V_N\left(\boldsymbol{\theta}, Z^N\right) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\left[y(t) - \varphi^T(t)\,\boldsymbol{\theta}\right]^2, \tag{2.24}$$

que constitui o *critério de minimização dos mínimos quadrados*. O vetor de parâmetros estimados $\hat{\boldsymbol{\theta}}_N^{LS}$ é obtido minimizando-se a equação (2.24), ou seja,

$$\hat{\boldsymbol{\theta}}_N^{LS} = \arg\min_{\boldsymbol{\theta}\in D_{\mathcal{M}}} V_N\left(\boldsymbol{\theta}, \mathbf{Z}^N\right). \tag{2.25}$$

Como V_N é quadrática em $\boldsymbol{\theta}$, pode-se encontrar o valor mínimo derivando a equação (2.24). Deste modo,

$$\frac{d}{d\boldsymbol{\theta}}V_N\left(\boldsymbol{\theta}, Z^N\right) = \frac{1}{N}\sum_{t=1}^{N}\varphi(t)\left(y(t) - \varphi^T(t)\,\boldsymbol{\theta}\right),$$

e

$$\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\,y(t)=\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\,\boldsymbol{\varphi}^{T}(t)\,\boldsymbol{\theta}.$$

O vetor paramétrico é calculado através de

$$\hat{\boldsymbol{\theta}}_{N}^{LS}=\left[\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\,\boldsymbol{\varphi}^{T}(t)\right]^{-1}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\,y(t),$$

que corresponde a

$$\hat{\boldsymbol{\theta}}_{N}^{LS}=\underset{\boldsymbol{\theta}\in D_{\mathcal{M}}}{\arg\min}\,V_{N}\left(\boldsymbol{\theta},\mathbf{Z}^{N}\right)=\left[\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\,\boldsymbol{\varphi}^{T}(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\,y(t). \tag{2.26}$$

Percebe-se que nas equações de estimação, para o cálculo do vetor de parâmetros $\hat{\boldsymbol{\theta}}_{N}^{LS}$, há uma inversão matricial. A inversão da matriz só pode ser efetuada após somatório dos dados do sistema considerando todas as amostras disponíveis. Desta forma, quando novos pontos de entrada e saída do processo estão disponíveis para ser incluídos no cálculo dos parâmetros, todos os cálculos devem ser refeitos, inclusive a inversão matricial.

Em casos de modelagem em tempo real do sistema, deve-se atualizar os parâmetros estimados sem repetir os cálculos já executados anteriormente, pois antes de uma nova amostra todos os cálculos dos parâmetros devem estar concluídos. Utiliza-se então uma *implementação recursiva do método* de estimação.

Uma motivação para identificação paramétrica em tempo real é que em geral os parâmetros de um processo são variantes no tempo. Nota-se que um algorítmo de identificação mais apropriado para o caso variante no tempo é aquele que dá ênfase aos dados mais recentes do sistema, descartando os dados mais antigos. Alguns dos algorítmos elaborados para a identificação possuem uma variável conhecida como *fator de esquecimento* que atende a esta necessidade e será citada posteriormente.

A função de custo (2.22) considera que os dados do processo possuem o mesmo grau de representatividade para o sistema, ou que o ruído ataca o sistema de forma constante e contínua, mas isto nem sempre é verdade. Para contornar este problema, pode ser utilizada uma função de custo que varie com o tempo. Desta forma é implementada uma ponderação sobre os dados do processo.

A equação (2.23) na forma ponderada torna-se então

$$V_{N}\left(\boldsymbol{\theta},\mathbf{Z}^{N}\right)=\frac{1}{N}\sum_{t=1}^{N}l\left(\varepsilon\left(t,\boldsymbol{\theta}\right),\boldsymbol{\theta},t\right). \tag{2.27}$$

Utilizando a função de ponderação $\beta(N,t)$ a expressão (2.27) modifica-se para

$$V_N\left(\boldsymbol{\theta}, \mathbf{Z}^N\right) = \sum_{t=1}^{N} \beta(N,t)\, l\left(\varepsilon(t,\boldsymbol{\theta}), \boldsymbol{\theta}\right), \tag{2.28}$$

e a solução dada em (2.26) torna-se,

$$\begin{aligned} \hat{\boldsymbol{\theta}}_N^{LS} &= \arg\min \left[\sum_{t=1}^{N} \beta(N,t)\left[y(t) - \boldsymbol{\varphi}^T(t)\,\boldsymbol{\theta}\right]^2\right], \\ \hat{\boldsymbol{\theta}}_N^{LS} &= \left[\sum_{t=1}^{N} \beta(N,t)\,\boldsymbol{\varphi}(t)\,\boldsymbol{\varphi}^T(t)\right]^{-1} \sum_{t=1}^{N} \beta(N,t)\,\boldsymbol{\varphi}(t)\,y(t). \end{aligned} \tag{2.29}$$

Os algorítmos de identificação comparam a saída real do sistema $y(t)$ com a saída do modelo paramétrico $\hat{y}(t)$, sobre um intervalo de tempo suficiente e com uma excitação persistente na entrada do sistema, Figura 2.4.

Como foi visto, os métodos de identificação paramétricos mapeiam o conjunto de dados amostrados do sistema, $\mathbf{Z}^N$, para o espaço dos parâmetros da planta, $D_{\mathcal{M}}$,

$$\mathbf{Z}^N \Longrightarrow \hat{\boldsymbol{\theta}}_N \in D_{\mathcal{M}},$$

ou

$$\hat{\boldsymbol{\theta}}_t = F\left(t, \mathbf{Z}^t\right),$$

nos métodos não-recursivos.

Nos métodos recursivos

$$\mathbf{X}(t) = H\left(t, \mathbf{X}(t-1), y(t), u(t)\right),$$

e

$$\hat{\boldsymbol{\theta}}_t = h\left(\mathbf{X}(t)\right),$$

no qual $\mathbf{X}(t)$ é um vetor de dimensão fixa que representa alguma informação de estado. $\hat{\boldsymbol{\theta}}_t$ é avaliado baseando-se nos valores atuais das amostras no tempo t e do valor passado da informação do estado anterior $\mathbf{X}(t-1)$. Como a informação contida no par $y(t)$ e $u(t)$ é pequena em comparação com a informação acumulada, as equações de estimação têm a forma

$$\begin{aligned} \hat{\boldsymbol{\theta}}_t &= \hat{\boldsymbol{\theta}}_{t-1} + \gamma_t Q_{\boldsymbol{\theta}}\left(\mathbf{X}(t), y(t), u(t)\right), \\ \mathbf{X}(t) &= \mathbf{X}(t-1) + \mu_t Q_{\mathbf{X}}\left(\mathbf{X}(t-1), y(t), u(t)\right), \end{aligned}$$

nas quais γ e μ são números pequenos que ponderam os valores amostrados, Q_θ e Q_X são funções que definem os fatores de correção para $\hat{\boldsymbol{\theta}}_t$ e $\mathbf{X}(t)$, respectivamente.

Para deduzir a forma recursiva do algoritmo, retorna-se para a equação (2.29), separando-a em dois termos,

$$\hat{\boldsymbol{\theta}}_t = \bar{\mathbf{R}}^{-1}(t) f(t), \tag{2.30}$$

em que,

$$\bar{\mathbf{R}}(t) = \sum_{k=1}^{t} \beta(t,k) \boldsymbol{\varphi}(k) \boldsymbol{\varphi}^T(k)$$

e

$$f(t) = \sum_{k=1}^{t} \beta(t,k) \boldsymbol{\varphi}(k) y(k),$$

e supõe-se que $\beta(t,k)$ possua a seguinte propriedade,

$$\begin{aligned} \beta(t,k) &= \lambda(t)\beta(t-1,k) \qquad 1 \leq k \leq t-1 \\ \beta(t,t) &= 1 \end{aligned}$$

então,

$$\bar{\mathbf{R}}(t) = \lambda(t)\, \bar{\mathbf{R}}(t-1) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t), \tag{2.31}$$

$$f(t) = \lambda(t)\, f(t-1) + \boldsymbol{\varphi}(t) y(t), \tag{2.32}$$

e, das equações (2.30), (2.31) e (2.32), tem-se,

$$\begin{aligned} \hat{\boldsymbol{\theta}}_t &= \bar{\mathbf{R}}^{-1}(t) f(t) = \bar{\mathbf{R}}^{-1}(t)\left[\lambda(t)\, f(t-1) + \boldsymbol{\varphi}(t) y(t)\right] \\ &= \bar{\mathbf{R}}^{-1}(t)\left[\lambda(t)\bar{\mathbf{R}}(t-1)\hat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{\varphi}(t) y(t)\right] \\ &= \bar{\mathbf{R}}^{-1}(t)\left\{\left[\bar{\mathbf{R}}(t) - \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\right]\hat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{\varphi}(t) y(t)\right\} \\ &= \hat{\boldsymbol{\theta}}_{t-1} + \bar{\mathbf{R}}^{-1}(t)\boldsymbol{\varphi}(t)\left[y(t) - \boldsymbol{\varphi}^T(t)\hat{\boldsymbol{\theta}}_{t-1}\right], \end{aligned}$$

$\hat{\boldsymbol{\theta}}_t$ e $\bar{\mathbf{R}}(t)$ formam um algorítmo recursivo.

Para evitar a inversão matricial de $\bar{\mathbf{R}}(t)$ a cada passo, pode-se introduzir,

$$\mathbf{P}(t) = \bar{\mathbf{R}}^{-1}(t),$$

aplicar o lema da inversão matricial,

$$[\mathbf{A} + \mathbf{BCD}]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left[\mathbf{DA}^{-1}\mathbf{B} + \mathbf{C}^{-1}\right]^{-1}\mathbf{DA}^{-1},$$

com $\mathbf{A} = \lambda(t)\,\bar{\mathbf{R}}(t-1)$, $\mathbf{B} = \mathbf{D}^T = \boldsymbol{\varphi}(t)$ e $\mathbf{C} = 1$, para obter,

$$\mathbf{P}(t) = \left[\mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\mathbf{P}(t-1)}{\lambda(t) + \boldsymbol{\varphi}^T(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}\right]\frac{1}{\lambda(t)} \tag{2.33}$$

e

$$\bar{\mathbf{R}}^{-1}(t)\,\boldsymbol{\varphi}(t) = \frac{\mathbf{P}(t-1)\,\boldsymbol{\varphi}(t)}{\lambda(t) + \boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)\,\boldsymbol{\varphi}(t)}.$$

Resumindo, o algorítmo recursivo é dado por,

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}(t) &= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{L}(t)\left[y(t) - \boldsymbol{\varphi}^T(t)\,\widehat{\boldsymbol{\theta}}(t-1)\right], \\
\mathbf{L}(t) &= \frac{\mathbf{P}(t-1)\,\boldsymbol{\varphi}(t)}{\lambda(t) + \boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)\,\boldsymbol{\varphi}(t)}, \\
\mathbf{P}(t) &= \left[\mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\,\boldsymbol{\varphi}(t)\,\boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)}{\lambda(t) + \boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)\,\boldsymbol{\varphi}(t)}\right]\frac{1}{\lambda(t)}.
\end{aligned} \tag{2.34}$$

A matriz de covariância $\mathbf{P}(t)$ da estimativa $\widehat{\boldsymbol{\theta}}_N$ é inteiramente determinada pelas propriedades da entrada e pelo nível de ruído. A covariância é proporcional à variância do ruído e inversamente proporcional à potência da entrada [19]. Nos casos em que necessita-se de um modelo enquanto o sistema está em funcionamento normal, como por exemplo em sistemas adaptativos, lança-se mão das equações recursivas (2.34).

Todos os cálculos devem ser efetuados no intervalo de tempo transcorrido entre duas amostragens consecutivas dos sinais. O tempo de identificação deve ser curto comparado com a taxa de variação paramétrica da planta de forma a permitir que estas modificações sejam percebidas pois, de outra forma, será encontrado um modelo que não representa a realidade, isto é, no momento em que os cálculos do modelo estimado forem concluídos o modelo real já será bem diferente daquele no qual tais cálculos se basearam. O tempo de identificação e a taxa de amostragem também devem ser compatíveis com as constantes de tempo do sistema [6].

O tempo de convergência dos parâmetros depende de fatores como o método de identificação utilizado ou os valores de inicialização dos vetores do algoritmo, e deve ser o menor possível se o resultado estiver sendo usado para sistemas de controle adaptativo. Em alguns algorítmos de identificação pode-se partir de valores de parâmetros próximos dos valores reais. Portanto, quanto mais próximos do valor real estiverem os valores iniciais, mais rápido o modelo estimado irá convergir para o modelo real. Deste modo, pode-se utilizar um algorítmo de identificação veloz, porém menos preciso, para dar a partida em outro algorítmo que embora mais lento, propicie uma maior precisão nos valores dos parâmetros encontrados.

Para compensar as variações paramétricas da planta, deve ser feita uma ponderação sobre os novos dados adquiridos, pode ser utilizado um *fator de esquecimento*. A escolha do *fator de esquecimento* deve ser feita de modo que a informação relevante

para a caracterização do sistema seja corretamente ponderada. Tal ponderação pode ser definida pela escolha particular de $\beta(N,t)$ na equação (2.28).

Escolha do fator de esquecimento

Com os parâmetros do processo variando com o tempo, geralmente só se utiliza uma quantidade finita de dados passados para realizar a estimação. Isto é feito para que o modelo estimado seja fiel à planta. Esta seleção de segmentos de entradas e saídas da planta pode ser feita com a inclusão do fator de esquecimento em algorítmos de estimação paramétrica recursivos.

Fatores de esquecimento constante geram problemas sempre que o processo está em repouso pois nenhuma informação é obtida, isto é, pode ocorrer um estouro da matriz de covariância. Fatores de esquecimento variáveis tentam resolver este problema aplicando valores que reduzem o peso dos dados sempre que o processo está em repouso e valores que aumentam o peso dos dados quando mudanças significativas ocorrem na operação do sistema [4]. As variações no tempo são modeladas por esquecimento, ou seja, determinação e esquecimento de informações obsoletas.

Pode ser associado um peso menor para amostras mais antigas que não são mais representativas do processo, assim escolhe-se $\beta(t,k) = \lambda^{t-k}$, com $\lambda < 1$ e $k = 1, 2, \cdots, t$, esta escolha representa um fator de esquecimento exponencial.

A escolha do fator de esquecimento deve ser feita de modo a descartar os pontos mais antigos e menos representativos e associar um peso maior aos pontos recentes e mais representativos do sistema atual, considerando os pontos relevantes para a dinâmica do sistema. Para um sistema com variação gradual e lenta, um fator de esquecimento constante é a escolha mais comum, ou seja, $\lambda(t) \equiv \lambda$. Escolhas típicas para λ estão na faixa entre $0,98$ e $0,995$ [9], porém o sistema não deve ir para a condição de repouso.

Se o sistema muda de forma repentina e abruptamente, uma boa escolha para λ é adotar uma forma adaptativa, $\lambda(t)$, diminuindo seu valor nas mudanças abruptas e aumentando-o nas fases mais estáveis *ou estáticas* do sistema. Isto é discutido em [9] e [21].

Sinal de Excitação

Nos experimentos de estimação paramétrica o sinal de entrada pode ser injetado pelo usuário em experimentos com o processo desconectado do seu local de trabalho ou o sinal de entrada pode simplesmente já estar presente no processo e apenas ser lido pelo sistema de identificação em tempo real.

Os sinais de entrada determinam o ponto de operação do sistema e que partes e modos do sistema são excitados durante o experimento. A liberdade de escolher as características do sinal de entrada varia consideravelmente com a aplicação, pois a excitação não deve danificar a planta ou por em risco a segurança do sistema.

Num processo industrial pode não ser permitido manipular integralmente o sistema de produção de modo contínuo. Para outros sistemas, tais como econômicos e ecológicos, simplesmente não é permitido perturbar o sistema para o propósito do experimento de identificação, pois seriam causados danos inconcebíveis. Por outro lado, em aplicações de laboratório e durante fases de desenvolvimento de novos equipamentos a escolha da entrada é talvez restrita apenas por limites de potência e efeito [7].

A escolha do sinal de excitação deve ser feita de modo que o sinal seja *persistentemente excitante*, ou seja, deve excitar o sistema em uma faixa ampla de espectro de frequência semelhante à faixa de operação a que o sistema será submetido, para que o modelo estimado represente o sistema em uma faixa de frequência desejada. O sinal de excitação deve ser rico o suficiente em freqüências para uma boa estimação do processo. Assim, na escolha do sinal de excitação deve ser levado em consideração tanto a segurança como a excitabilidade do mesmo. Este nível de excitação pode ser medido em termos de ordem de persistência da excitação.

Persistência da Excitação Sistemas lineares invariantes no tempo e estáveis, em que a resposta ao impulso tende para zero com o tempo, são conhecidos como modelos *FIR*, isto é, de resposta finita ao impulso. Estes sistemas podem ser descritos pelas equações,

$$y(t) = B(q)\,u(t) = \boldsymbol{\varphi}^T(t-1)\,\boldsymbol{\theta}, \tag{2.35}$$

em que,

$$\begin{aligned} \boldsymbol{\theta}^T &= [b_1 \ \cdots \ b_n], \\ \boldsymbol{\varphi}^T(t-1) &= [u(t-1) \ \cdots \ u(t-n)] \end{aligned}$$

os parâmetros da equação (2.35) não podem ser determinados a menos que algumas condições sejam impostas ao sinal de entrada. Para que o $V_N\left(\boldsymbol{\theta}, \mathbf{Z}^N\right)$, visto na equação (2.28) seja mínimo, e único, é necessário que a matriz $\boldsymbol{\varphi}(t)\,\boldsymbol{\varphi}^T(t)$ tenha posto completo. Esta condição é chamada de *condição de excitação.*

Definição: Um sinal de entrada para um sistema FIR é dito ter persistência de

ordem n se a matriz,

$$\mathbf{R}_u = \begin{bmatrix} r_u(0) & r_u(1) & \cdots & r_u(n-1) \\ \vdots & \vdots & \ddots & \vdots \\ r_u(1-n) & \cdots & \cdots & r_u(0) \end{bmatrix},$$

for *definida positiva*, na qual,

$$r_u(s) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} u(t+s)\, u(t),$$

Lema 1 *Uma matriz* $\mathbf{A} \in \Re^{n\times n}$ *é dita* definida positiva *se* $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ *para todo* $\mathbf{x}$ *não nulo, em que* $\mathbf{x} \in \Re^n$.

Uma apresentação mais detalhada sobre matrizes *definidas positivas* pode ser encontrada Golub e Van Loan [22].

A ordem de excitação de alguns sinais de entrada comumente utilizados são então apresentadas na Tabela 2.1, em que PRBS representa um sinal binário pseudo-aleatório (*Pseudo-Random Binary Signal*).

A ordem de excitação do sinal de entrada para o processo é crítica para a obtenção de informação suficiente, a partir dos dados de entrada e saída, para uma modelagem adequada do processo. A ordem de excitação que deve ser utilizada depende da ordem do processo a ser modelado. Daí conclui-se que sinais do tipo degrau e impulso são, de modo geral, inadequados para estimação paramétrica de sistemas de ordem superior a primeira ordem ou sistemas de qualquer ordem no caso do impulso, respectivamente. Em [23], é apresentada a síntese de um sinal com características de excitabilidade semelhantes ao ruído branco.

Lema 2 *Um sinal* $\{u(t)\}$ *é dito* ***quasi-estacionário*** *se o valor esperado de* $u(t)$ *e a covariância são respectivamente,*

$$\begin{aligned} Eu(t) &= m_u(t), \qquad |m_u(t)| \le C, \qquad \forall t \\ Eu(t)\, u(r) &= R_u(t,r), \qquad |R_u(t,r)| \le C, \end{aligned}$$

em que,

$$R_u(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} R_u(t, t-\tau), \qquad \forall \tau$$

Sinal	Ruído Branco	PRBS	degrau	impulso
Ordem de excitação	∞	∞	1	0

Tabela 2.1: Tabela com ordem de excitação de alguns sinais conhecidos.

na qual C é um valor constante. Ou seja, se o valor esperado de $u(t)$ e a autocorrelação forem limitados.

Lema 3 *Seja $u(t)$ um sinal **quasi-estacionário** e seja $\overline{\mathbf{R}}_n$ uma matriz $n \times n$ definida por,*

$$\overline{\mathbf{R}}_n = \begin{bmatrix} r_u(0) & r_u(1) & \cdots & r_u(n-1) \\ r_u(1) & r_u(0) & \cdots & r_u(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_u(n-1) & r_u(n-2) & \cdots & r_u(0) \end{bmatrix},$$

com r_u representando a autocorrelação de u, então $u(t)$ é persistentemente excitante de ordem n, se e somente se $\overline{\mathbf{R}}_n$ é não singular [9].

Definição: Um sinal **quasi-estacionário** $\{u(t)\}$, com espectro $\Phi_u(\omega)$ é dito ser persistentemente excitante de ordem n, se para todo filtro da forma,

$$M_n(q) = m_1 q^{-1} + \cdots + m_n q^{-n},$$

a relação $|M_n(e^{i\omega})|^2 \Phi_u(\omega) \equiv 0$ implica que $M_n(e^{i\omega}) \equiv 0$ [9].

Diz-se que $u(t)$ é persistentemente excitante de ordem n, se $\Phi_u(\omega)$ é diferente de zero em pelo menos n pontos no intervalo $-\pi < \omega \leq \pi$.

Definição: O sinal $\{u(t)\}$, é dito ser persistentemente excitante se o espectro $\Phi_u(\omega)$, definido por,

$$\Phi_u(\omega) = \sum_{\tau=-\infty}^{\infty} R_u(\tau) e^{-i\tau\omega},$$

for maior que zero para qualquer ω,

$$\Phi_u(\omega) > 0, \qquad \forall\omega.$$

Na ausência de um sinal de exitação com ordem de persistência adequada para o processo, os parâmetros estimados podem exibir um deslocamento em relação aos valores reais, ou até mesmo obter-se um modelo instável do processo com o *estouro da matriz de covariância.* Para evitar tais problemas, propoem-se alguns algorítmos nos quais suspende-se a atualização do vetor de parâmetros estimados nestas situações, é o caso por exemplo do método dos mínimos quadrados recursivo incremental, por Barros em [24], esta opção também é utilizada por outros autores, como Lucena Jr. em [25], e Sripada & Fisher em [26].

2.2.3 Método das Variáveis Instrumentais

Seja $\boldsymbol{\theta}_0$ o vetor de parâmetros reais da planta e um certo $v_0(t)$, associado aos ruídos e imperfeições no sistema. Quando os dados observados na saída foram gerados por,

$$y(t) = \boldsymbol{\varphi}^T(t)\,\boldsymbol{\theta}_0 + v_0(t), \tag{2.36}$$

então inserindo (2.36) na equação (2.26), a estimação pelo método dos mínimos quadrados é dada por,

$$\widehat{\boldsymbol{\theta}}_N^{LS} = \left[\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\,\boldsymbol{\varphi}^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\left[\boldsymbol{\varphi}^T(t)\,\boldsymbol{\theta}_0 + v_0(t)\right],$$

$$\widehat{\boldsymbol{\theta}}_N^{LS} = \boldsymbol{\theta}_0 + [\mathbf{R}(N)]^{-1}\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\,v_0(t). \tag{2.37}$$

Desta forma, pode ser visto que neste caso o método dos mínimos quadrados (MMQ) fornece uma estimação tendenciosa do modelo, assim este não fornece uma boa aproximação para os parâmetros do sistema. Isto ocorre quando $v_0(t)$ e $\boldsymbol{\varphi}(t)$ são correlacionados, na expressão (2.37), adicionando um resíduo ao vetor $\widehat{\boldsymbol{\theta}}_N^{LS}$.

As propriedades desejadas para $\widehat{\boldsymbol{\theta}}_N^{LS}$ são: que $\widehat{\boldsymbol{\theta}}_N^{LS}$ seja próximo de $\boldsymbol{\theta}_0$ e que $\widehat{\boldsymbol{\theta}}_N^{LS}$ convirja para $\boldsymbol{\theta}_0$ quando N tender para infinito [9].

Para que estas propriedades ocorram, de (2.37) pode ser visto que é preciso que $v_0(t)$ seja pequeno em relação a $\boldsymbol{\varphi}(t)$. Se $v_0(t)$ é pequeno em comparação a $\boldsymbol{\varphi}(t)$, então o segundo termo da equação (2.37) também será pequeno e assim $\widehat{\boldsymbol{\theta}}_N^{LS}$ será próximo de $\boldsymbol{\theta}_0$. Ou então, que $v_0(t)$ seja uma sequência de variáveis aleatórias independentes com média zero, um ruído branco, ou seja, não depender do que aconteceu no tempo $t-1$, e assim ser descorrelacionado com $\boldsymbol{\varphi}(t)$, ou

$$E\{\boldsymbol{\varphi}(t)\,v_0(t)\} = 0.$$

Em relação à segunda propriedades desejada para $\widehat{\boldsymbol{\theta}}_N^{LS}$, pode ser mostrado que quando,

$$N \to \infty, \qquad \frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\,v_0(t) = E\{\boldsymbol{\varphi}(t)\,v_0(t)\}.$$

Porém $v_0(t)$ não é um ruído branco e N não é infinito. Assim, com $v_0(t)$ e $\boldsymbol{\varphi}(t)$ correlacionados e N um número finito, pode-se dizer que neste caso o método MMQ é uma estimação tendenciosa, ou polarizada, dos parâmetros do processo.

Então o método estima não só o modelo da planta, mas também o modelo do ruído, só que os dois são representados em $\widehat{\boldsymbol{\theta}}_N^{LS}$. Assim, torna-se difícil determinar o

que é parte da planta e o que é parte da dinâmica da perturbação. Assim, o vetor de parâmetros estimado $\widehat{\boldsymbol{\theta}}_N^{LS}$ não converge para $\boldsymbol{\theta}_0$ nos casos típicos da estimação pelo método *MMQ*.

Para melhorar as estimativas nestes casos poderia-se utilizar o *método das variáveis instrumentais (IV)*, no qual escolhe-se um vetor $\boldsymbol{\zeta}(t)$, conhecido como vetor de instrumentos, tal que este não seja correlacionado com $v_0(t)$, e com isto melhorar a convergência para os parâmetros reais da planta. Desta forma tem-se, como visto na equação (2.26),

$$\widehat{\boldsymbol{\theta}}_N^{IV} = \left[\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\zeta}(t)\,\boldsymbol{\varphi}^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\zeta}(t)\left[\boldsymbol{\varphi}^T(t)\,\boldsymbol{\theta}_0 + v_0(t)\right], \tag{2.38}$$

$$\widehat{\boldsymbol{\theta}}_N^{IV} = \boldsymbol{\theta}_0 + \left[\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\zeta}(t)\,\boldsymbol{\varphi}^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\zeta}(t)\,v_0(t), \tag{2.39}$$

contanto que exista a inversa,

$$\left[\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\zeta}(t)\,\boldsymbol{\varphi}^T(t)\right]^{-1}.$$

Para $\widehat{\boldsymbol{\theta}}_N^{IV}$ convergir para $\boldsymbol{\theta}_0$ com um valor grande de N, nota-se de (2.38) que então,

$$\frac{1}{N}\sum_{t=1}^{N}\boldsymbol{\zeta}(t)\,v_0(t),$$

deve tender para zero. Para que o método seja aplicado com sucesso no sistema (2.36), a variável instrumental deve estar condicionada às seguintes propriedades:

$$\overline{E}\left\{\boldsymbol{\zeta}(t)\,\boldsymbol{\varphi}^T(t)\right\} \quad \text{deve ser não singular,} \tag{2.40}$$

$$\overline{E}\left\{\boldsymbol{\zeta}(t)\,v_0(t)\right\} = 0 \quad \text{para que} \quad \widehat{\boldsymbol{\theta}}_N^{IV} \to \boldsymbol{\theta}_0. \tag{2.41}$$

O instrumento $\boldsymbol{\zeta}(t)$ é escolhido de forma que $\overline{E}\left\{\boldsymbol{\zeta}(t)\,\boldsymbol{\varphi}^T(t)\right\}$ deve ser não singular e $\overline{E}\left\{\boldsymbol{\zeta}(t)\,v_0(t)\right\} = 0$, em outras palavras, deve-se ter correlação entre $\boldsymbol{\zeta}(t)$ e $\boldsymbol{\varphi}(t)$ mas descorrelação entre $\boldsymbol{\zeta}(t)$ e $v_0(t)$, ou seja, $\boldsymbol{\zeta}(t)$ correlacionado com as variáveis de regressão mas descorrelacionado com o ruído [9].

Para garantir que a propriedade (2.40) seja atendida e o vetor de instrumentos $\boldsymbol{\zeta}(t)$ não seja influenciado por $\{v_0(t)\}$, pode-se gerar o vetor da seguinte forma:

$$\boldsymbol{\zeta}(t) = K(q)\left[\begin{array}{cccccccc} -x(t-1) & -x(t-2) & \cdots & -x(t-n_a) & u(t-1) & \cdots & u(t-n_b) \end{array}\right]^T, \tag{2.42}$$

em que $K(q)$ é um filtro linear e $x(t)$ é gerado a partir do sinal de entrada $u(t)$ do sistema, filtrado pelo sistema linear:

$$N(q)\, x(t) = M(q)\, u(t),$$

com

$$N(q) = 1 + n_1 q^{-1} + ... + n_{n_n} q^{-n_n},$$

e

$$M(q) = m_0 + m_1 q^{-1} + ... + m_{n_m} q^{-n_m}.$$

Uma escolha comum é montar um vetor $\boldsymbol{\zeta}(t)$ com $y_M(t)$, ao invés de $y(t)$, em que $y_M(t)$ é a saída de um sistema determinístico excitado pela entrada $u(t)$.

Se a entrada é gerada em malha aberta de forma que não depende do ruído $v_0(t)$, então a propriedade (2.41) é atendida. E desde que o vetor $\boldsymbol{\varphi}(t)$ e o vetor $\boldsymbol{\zeta}(t)$ foram gerados da mesma sequência de entrada, então a condição (2.40) é satisfeita [9].

Uma forma prática para montar o vetor de instrumentos é utilizar o modelo obtido da estimação pelo método dos mínimos quadrados para $N(q)$ e $M(q)$. Os instrumentos são então escolhidos como em (2.42), com $K(q) = 1$. Esta foi a forma adotada nas implementações do algorítmo *IV* durante o presente trabalho.

Método IV Recursivo

Do mesmo modo como foi visto para o caso do método dos mínimos quadrados, as equações recursivas para o método das variáveis instrumentais são implementadas como segue:

$$\widehat{\boldsymbol{\theta}}_{IV}(t) = \widehat{\boldsymbol{\theta}}_{IV}(t-1) + \mathbf{L}(t)\left[y(t) - \boldsymbol{\varphi}^T(t)\,\widehat{\boldsymbol{\theta}}_{IV}(t-1)\right], \tag{2.43}$$

$$\mathbf{L}(t) = \frac{\mathbf{P}(t-1)\,\boldsymbol{\zeta}(t)}{\lambda(t) + \boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)\,\boldsymbol{\zeta}(t)}, \tag{2.44}$$

$$\mathbf{P}(t) = \left[\mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\,\boldsymbol{\zeta}(t)\,\boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)}{\lambda(t) + \boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)\,\boldsymbol{\zeta}(t)}\right]\frac{1}{\lambda(t)}. \tag{2.45}$$

2.3 Conclusões

Neste capítulo foram apresentados e discutidos aspectos relativos à identificação de sistemas. Os modelos de processos comumente utilizados para estimação foram apresentados. O método dos mínimos quadrados pode ser considerado como um método básico de identificação que pode ser utilizado como algorítmo de partida para outros métodos, como ocorre com o método das variáveis instrumentais e outros que serão

vistos nos próximos capítulos. Para os métodos recursivos a escolha do fator de esquecimento é de fundamental importância para a obtenção de resultados satisfatórios. O sinal utilizado para a excitação também deve ser cuidadosamente elaborado, sendo que a persistência do sinal é uma característica crítica para o experimento de identificação.

Apesar da identificação de sistemas envolver diversos parâmetros, abordá-los exaustivamente depende da necessidade do leitor, foram fornecidas diversas referências sobre o assunto, e além disto os parâmetros de interesse foram tratados até um nível que achou-se necessário para a compreensão das rotinas implementadas neste trabalho.

Como visto, nos modelos *caixa-preta* não são utilizados conhecimentos *a priori* do sistema, portanto torna-se necessário que o usuário destes métodos utilize formas alternativas para determinação da ordem adequada para o modelo estimado. No próximo capítulo será análisado um método para a redução da ordem de modelos, como também uma técnica de estimação que utiliza uma forma alternativa para a atualização da matriz de covariância $\mathbf{P}(t)$. A teoria básica sobre decomposição em valores singulares, ferramenta utilizada em alguns dos métodos comentados neste parágrafo, será vista no próximo capítulo.

Capítulo 3

Identificação e Redução de Modelos utilizando SVD

Neste capítulo são apresentadas definições introdutórias da teoria de matrizes, normas matriciais e ortogonalidade, decomposição em valores singulares *(SVD - Singular Value Decomposition)* e as vantagens da utilização deste tipo de transformação matricial. É apresentada uma técnica que utiliza o método dos mínimos quadrados e o cálculo de uma *realização mínima* no espaço de estado para estimar um modelo de ordem reduzida para o processo. Apresenta-se ainda, um algorítmo de identificação recursiva baseada em *SVD*.

3.1 Definições Básicas na Teoria de Matrizes

Nesta seção apresentam-se definições básicas da teoria de matrizes, que são relevantes para este trabalho. Uma apresentação mais detalhada o leitor poderá encontrar em literaturas tais como [22] e [27].

3.1.1 Espaço Imagem e Espaço Nulo

Espaço imagem e espaço nulo de uma transformação são conceitos importantes, definidos em seguida.

1. Se $\mathbf{A} \in \Re_r^{m \times n}$, diz-se que $\mathbf{A}$ pertence ao conjunto de todas as matrizes reais de m linhas e n colunas com posto r.

2. Dado Y um espaço vetorial, diz-se que $X \subseteq Y$ é um subespaço de Y, se X tem as propriedades de um espaço vetorial.

Para $\mathbf{A} \in \Re_r^{m\times n}$, $\mathbf{x} \in \Re^n$ e $\mathbf{y} \in \Re^m$, a equação,

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

diz que $\mathbf{y}$ é uma combinação linear das colunas de $\mathbf{A}$ com pesos iguais aos componentes correspondentes de $\mathbf{x}$. Se $\mathbf{y}$ é alguma combinação linear das colunas de $\mathbf{A}$ então é dito que ele pertence ao *espaço imagem* de $\mathbf{A}$, denotado $\mathcal{R}(\mathbf{A})$, isto é,

$$\mathcal{R}(\mathbf{A}) = \{\mathbf{x}, \mathbf{y} : \mathbf{x} \in \Re^n,\ \mathbf{y} \in \Re^m \quad \text{e} \quad \mathbf{y} = \mathbf{A}\mathbf{x}\}. \tag{3.1}$$

Se $\mathbf{y}_1$ e $\mathbf{y}_2$ estão na imagem de $\mathbf{A}$, então quaisquer combinações lineares deles $c_1\mathbf{y}_1 + c_2\mathbf{y}_2$, em que c_1 e c_2 são escalares, também farão parte da imagem de $\mathbf{A}$. Se $\mathbf{A}\mathbf{x}_1 = \mathbf{y}$ e $\mathbf{A}\mathbf{x}_2 = \mathbf{y}$ então $\mathbf{A}(\mathbf{x}_1 - \mathbf{x}_2) = 0$.

O *espaço nulo* de $\mathbf{A}$ denotado por $\mathcal{N}(\mathbf{A})$ é o espaço de todas as soluções da equação $\mathbf{A}\mathbf{x} = 0$.

$$\mathcal{N}(\mathbf{A}) = \{\mathbf{x} : \mathbf{x} \in \Re^n \quad \text{e} \quad \mathbf{A}\mathbf{x} = 0\}. \tag{3.2}$$

3.1.2 Norma Matricial

Para quantificar a noção de proximidade da singularidade de uma certa matriz faz-se uso da *norma matricial*, que fornece uma medida de distância no espaço de matrizes. Isto facilita a definição da qualidade da solução do problema linear. Seja $f(\mathbf{A}) = \|\mathbf{A}\|$, a norma matricial de $\mathbf{A}$, então,

$$f(\mathbf{A}) \geq 0, \qquad \mathbf{A} \in \Re^{m\times n}, \quad (f(\mathbf{A}) = 0 \text{ se } \mathbf{A} = 0), \tag{3.3}$$

$$f(\mathbf{A}+\mathbf{B}) \leq f(\mathbf{A}) + f(\mathbf{B}), \qquad \mathbf{A}, \mathbf{B} \in \Re^{m\times n}, \tag{3.4}$$

$$f(\alpha\mathbf{A}) = |\alpha| f(\mathbf{A}), \qquad \alpha \in \Re, \quad \mathbf{A} \in \Re^{m\times n}. \tag{3.5}$$

A *norma quadrática* de uma matriz $\mathbf{A} \in \Re^{m\times n}$ pode ser definida como,

$$\|\mathbf{A}\|_2 = \left(\lambda_{\max}\left(\mathbf{A}^T\mathbf{A}\right)\right)^{\frac{1}{2}}.$$

Uma norma matricial comumente usada na algebra linear é a *Norma de Frobenius*

$$\|\mathbf{A}\|_F = \left(\sum_{i=1,j=1}^{m}\sum^{n} |a_{ij}|^2\right)^{\frac{1}{2}}. \tag{3.6}$$

3.1.3 Ortogonalidade

Algumas definições sobre ortogonalidade e outras extraídas da teoria de sistemas lineares também serão mencionadas por se tratarem de ferramentas básicas na análise da teoria de decomposição em valores singulares, que será tratada adiante. Estas definições estão listadas a seguir,

1. Um conjunto de vetores $\{\mathbf{x}_1, \cdots, \mathbf{x}_p\}$ em $\Re^m$ é *ortogonal* se $\mathbf{x}_i^T \mathbf{x}_j = 0$ sempre que $i \neq j$ e *ortonormal* se $\mathbf{x}_i^T \mathbf{x}_j = \boldsymbol{\delta}_{ij}$. O $\boldsymbol{\delta}_{ij}$ é conhecido como o *delta de Kronecker* e $\boldsymbol{\delta}_{ij} = 0$, para $i \neq j$ e $\boldsymbol{\delta}_{ij} = 1$, para $i = j$.

2. Uma coleção de subespaços $S_1, \cdots, S_p$ em $\Re^m$ é mutuamente ortogonal se $\mathbf{x}^T \mathbf{y} = 0$ sempre que $\mathbf{x} \in S_i$ e $\mathbf{y} \in S_j$ para $i \neq j$.

3. O complemento ortogonal de um subespaço $S \subseteq \Re^m$ é definido por,

$$S^{\perp} = \{\mathbf{y} \in \Re^m : \mathbf{y}^T \mathbf{x} = 0 \text{ para todo } \mathbf{x} \in S\}.$$

4. Pode ser mostrado que, $\mathcal{R}(\mathbf{A})^{\perp} = \mathcal{N}\left(\mathbf{A}^T\right)$ [22]

5. Uma matriz $\mathbf{P} \in \Re^{m \times m}$ é dita ser *ortogonal* se $\mathbf{P}^T \mathbf{P} = \mathbf{I}$, assim $\mathbf{P}^T = \mathbf{P}^{-1}$.

6. Se $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_m]$ é *ortogonal*, então o vetor $\mathbf{p}_i$ forma uma *base ortonormal* para $\Re^m$.

7. Dada uma coleção de vetores $\{\mathbf{a}_1, \cdots, \mathbf{a}_n\}$ em $\Re^m$, o conjunto de todas as combinações lineares possíveis destes vetores é um subespaço referido como *span* de $\{\mathbf{a}_1, \cdots, \mathbf{a}_n\}$, denotado por,

$$span\{\mathbf{a}_1, \cdots, \mathbf{a}_n\} = \left\{ \sum_{j=1}^{n} \beta_j \mathbf{a}_j : \beta_j \in \Re \right\}. \tag{3.7}$$

8. O *posto* de uma matriz é dado pelo número de linha, ou colunas *linearmente independentes*, e

$$\rho(\mathbf{A}) = \rho\left(\mathbf{A}^T\right). \tag{3.8}$$

9. Para qualquer $\mathbf{A} \in \Re^{m \times n}$,

$$\dim(\mathcal{N}(\mathbf{A})) + \rho(\mathbf{A}) = n. \tag{3.9}$$

$$\rho(\mathbf{A}) = \dim(\mathcal{R}(\mathbf{A})). \tag{3.10}$$

10. Sejam $\mathbf{Q}$ e $\mathbf{Z}$ matrizes ortogonais de dimensões apropriadas, então

$$\|\mathbf{QAZ}\|_2 = \mathbf{Z}^T\mathbf{A}^T\mathbf{Q}^T\mathbf{QAZ} = \mathbf{Z}^T\mathbf{A}^T\mathbf{AZ} = \|\mathbf{A}\|_2, \tag{3.11}$$

$$\|\mathbf{A}\|_2 = \lambda_{\max}\left(\mathbf{A}^T\mathbf{A}\right), \tag{3.12}$$

em que, $\lambda_{\max}\left(\mathbf{A}^T\mathbf{A}\right)$ é o maior autovalor de $\mathbf{A}^T\mathbf{A}$.

11. A *norma quadrática* é invariante sob uma transformação ortogonal, e isto pode ser demonstrado a seguir. Se $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, então

$$\|\mathbf{Qx}\|_2^2 = \mathbf{x}^T\mathbf{Q}^T\mathbf{Qx} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|_2^2. \tag{3.13}$$

3.2 Decomposição em Valores Singulares

Considerada uma das ferramentas básicas mais importantes da análise numérica moderna, particularmente álgebra linear numérica, a Decomposição em Valores Singulares (*SVD*) foi estabelecida para matrizes quadradas genéricas, reais e complexas por volta de 1870 [28].

Estabilidade numérica e condicionamento de um problema são conceitos de fundamental importância para a análise da decomposição *SVD*. Considerando-se um lote de dados $d \in \mathcal{D}$, no qual $\mathcal{D}$ é algum conjunto de dados. Deseja-se encontrar uma solução $f(d)$, na qual f representa um problema matemático. Produzir uma solução para este problema implica em computar $f(d) \in \mathcal{S}$, em que $\mathcal{S}$ representa algum conjunto de soluções. Considerando que seja conhecida apenas uma aproximação d^* para d, tem-se então que computar $f(d^*)$. Se $f(d^*)$ for próximo de $f(d)$ então o problema é dito ser *bem condicionado*, se $f(d^*)$ for numericamente muito *distante* de $f(d)$, o problema matemático é chamado de *mal condicionado*. Os conceitos de proximidade numérica são estabelecidos com base em informações sobre o problema, e determinada com base na teoria de análise matemática da perturbação.

Um algoritmo para determinar $f(d)$ é dito *numericamente estável* se ele não introduzir mais sensibilidade à perturbação além do que o sistema já possui. Seja f^* um algoritmo usado para implementar ou aproximar f. Então f^* é estável se para todo $d \in \mathcal{D}$ existir $d^* \in \mathcal{D}$ próximo a d tal que $f(d^*)$ é próximo de $f^*(d)$. Se o algoritmo é estável então $f^*(d)$ se aproxima de $f(d^*)$. Se o problema é *bem condicionado* então $f(d^*)$ se aproxima de $f(d)$, logo, nos melhores casos $f^*(d)$ se aproxima de $f(d)$.

De acordo com Klema e Laub [28], não se pode esperar que um algoritmo estável resolva um problema *mal condicionado* com mais precisão do que os dados propiciam,

mas um algoritmo instável poderá produzir soluções pobres mesmo para problemas *bem condicionados.*

A *SVD* é uma ferramenta atrativa para a solução de problemas em análise de sistemas lineares e na solução de problemas de convergência paramétrica na modelagem de processos atacados por ruídos, e na redução de ordem de modelos, em identificação de sistemas.

O teorema principal da teoria de *SVD* pode ser encontrado em Klema e Laub [28] e em Kailath [29], e será apresentado a seguir.

Teorema 1 *Para* $\mathbf{A} \in \Re_r^{m\times n}$, *existem matrizes ortogonais* $\mathbf{U} \in \Re^{m\times m}$ *e* $\mathbf{V} \in \Re^{n\times n}$, *tais que*

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \tag{3.14}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathbf{S} & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{0}_m \end{pmatrix}, \tag{3.15}$$

e

$$\mathbf{S} = diag\left(\sigma_1, \cdots, \sigma_r\right) = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_r \end{bmatrix}$$

com os σ_i, $i = 1, \cdots, r$ *representando os valores singulares, dispostos em ordem decrescente,*

$$\sigma_1 \geq \cdots \geq \sigma_r \geq 0. \tag{3.16}$$

Na equação (3.15) os $\mathbf{0}_m$'s representam matrizes nulas tal que $\boldsymbol{\Sigma}$ possui dimensão $m \times n$ e $\mathbf{S}$ possui dimensão $r \times r$, Pode-se também definir $\mathbf{0}_m$ com uma matriz identidade $\mathbf{I}$ de dimensão apropriada,

$$\mathbf{0}_m = \mathbf{0}\mathbf{I}.$$

Prova: Desde que, $\mathbf{A}^T\mathbf{A} \geq 0$ tem-se que,

$$\sigma\left(\mathbf{A}^T\mathbf{A}\right) \subseteq [0, +\infty),$$

em que denotou-se $\sigma\left(\mathbf{A}^T\mathbf{A}\right)$ por $\{\sigma_i^2, \quad i = 1, \cdots, n\}$. Pode-se organizar os valores singulares de modo que,

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_n. \tag{3.17}$$

Seja $\mathbf{S} = diag\left(\sigma_1, \cdots, \sigma_r\right)$, tem-se que de (3.14),

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\boldsymbol{\Sigma}^T\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T.$$

Como $\mathbf{U}^T = \mathbf{U}^{-1}$ e visto que $\mathbf{\Sigma}^T = \mathbf{\Sigma}$, pois $\mathbf{S}$ é uma matriz diagonal, então,

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T. \tag{3.18}$$

Seja $\mathbf{V}_1$ um conjunto de autovetores ortonormais correspondentes a $\mathbf{S}$, isto é,

$$\begin{aligned} \mathbf{V}_1 &= (\mathbf{v}_1, \cdots, \mathbf{v}_r), \\ \mathbf{V}_2 &= (\mathbf{v}_{r+1}, \cdots, \mathbf{v}_n), \\ \mathbf{V} &= \left[\begin{array}{c|c} \mathbf{V}_1 & \mathbf{V}_2 \end{array} \right]. \end{aligned} \tag{3.19}$$

De (3.15) e (3.19), pode-se então escrever (3.18) como,

$$\mathbf{A}^T\mathbf{A} = \left[\begin{array}{c|c} \mathbf{V}_1 & \mathbf{V}_2 \end{array} \right] \begin{pmatrix} \mathbf{S}^2 & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{0}_m \end{pmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \\ \mathbf{V}_2^T \end{bmatrix}. \tag{3.20}$$

Tomando só a parte não nula do produto (3.20), $\mathbf{V}_1$ e $\mathbf{S}$, tem-se que,

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}_1\mathbf{S}^2\mathbf{V}_1^T, \tag{3.21}$$

e multiplicando por $\mathbf{V}_1$ pela direita,

$$\mathbf{A}^T\mathbf{A}\mathbf{V}_1 = \mathbf{V}_1\mathbf{S}^2\mathbf{V}_1^T\mathbf{V}_1, \tag{3.22}$$

em que $\mathbf{V}_1$ é ortogonal então,

$$\mathbf{A}^T\mathbf{A}\mathbf{V}_1 = \mathbf{V}_1\mathbf{S}^2, \tag{3.23}$$

multiplicando ambos os lados da equação (3.23) por V_1^T, agora pela esquerda, tem-se,

$$\mathbf{V}_1^T\mathbf{A}^T\mathbf{A}\mathbf{V}_1 = \mathbf{V}_1^T\mathbf{V}_1\mathbf{S}^2, \tag{3.24}$$

$$\mathbf{V}_1^T\mathbf{A}^T\mathbf{A}\mathbf{V}_1 = \mathbf{S}^2, \tag{3.25}$$

multiplicando agora por $\mathbf{S}^{-1}$ tem-se,

$$\mathbf{S}^{-1}\mathbf{V}_1^T\mathbf{A}^T\mathbf{A}\mathbf{V}_1\mathbf{S}^{-1} = \mathbf{S}^{-1}\mathbf{S}^2\mathbf{S}^{-1}, \tag{3.26}$$

$$\mathbf{S}^{-1}\mathbf{V}_1^T\mathbf{A}^T\mathbf{A}\mathbf{V}_1\mathbf{S}^{-1} = \mathbf{I}. \tag{3.27}$$

Do mesmo modo, como os autovalores correspondentes de $\mathbf{V}_2$ são todos nulos ($\sigma_i = 0, i = r+1, \cdots, n$), tem-se de (3.20) que,

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}_2\mathbf{0}_m\mathbf{V}_2^T, \tag{3.28}$$

multiplicando por V_2 pela direita e depois pela esquerda tem-se que,

$$\mathbf{A}^T\mathbf{A}\mathbf{V}_2 = \mathbf{V}_2\mathbf{0}_m, \tag{3.29}$$

$$\mathbf{V}_2^T\mathbf{A}^T\mathbf{A}\mathbf{V}_2 = \mathbf{V}_2^T\mathbf{V}_2\mathbf{0}_m, \tag{3.30}$$

então, $\mathbf{V}_2^T\mathbf{A}^T\mathbf{A}\mathbf{V}_2 = \mathbf{0}_m$, e portanto implica que, $\mathbf{A}\mathbf{V}_2 = \mathbf{0}_m$.
Fazendo-se $\mathbf{U}_1 = \mathbf{A}\mathbf{V}_1\mathbf{S}^{-1}$, então a partir da equação (3.27) tem-se $\mathbf{U}_1^T\mathbf{U}_1 = \mathbf{I}$.
Escolhe-se qualquer $\mathbf{U}_2$ tal que $\mathbf{U} = \left[\begin{array}{c|c} \mathbf{U}_1 & \mathbf{U}_2 \end{array}\right]$, em que $\mathbf{U}_1$ e $\mathbf{U}_2$ sejam ortogonais.
Então da equação (3.27),

$$\begin{aligned}\mathbf{U}^T\mathbf{A}\mathbf{V} &= \begin{pmatrix} \mathbf{U}_1^T\mathbf{A}\mathbf{V}_1 & \mathbf{U}_1^T\mathbf{A}\mathbf{V}_2 \\ \mathbf{U}_2^T\mathbf{A}\mathbf{V}_1 & \mathbf{U}_2^T\mathbf{A}\mathbf{V}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{S}^{-1}\mathbf{V}_1^T\mathbf{A}^T\mathbf{A}\mathbf{V}_1 & \mathbf{S}^{-1}\mathbf{V}_1^T\mathbf{A}^T\mathbf{A}\mathbf{V}_2 \\ \mathbf{U}_2^T\mathbf{A}\mathbf{V}_1\mathbf{S}^{-1}\mathbf{S} & \mathbf{U}_2^T\mathbf{A}\mathbf{V}_2 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{S} & \mathbf{0}_m \\ \mathbf{U}_2^T\mathbf{U}_1\mathbf{S} & \mathbf{0}_m \end{pmatrix} = \begin{pmatrix} \mathbf{S} & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{0}_m \end{pmatrix} = \mathbf{\Sigma}, \end{aligned} \tag{3.31}$$

pois $\mathbf{U}_1$ e $\mathbf{U}_2$ são ortogonais $\mathbf{U}_2^T\mathbf{U}_1 = \mathbf{0}_m$, então, $\mathbf{U}\mathbf{U}^T\mathbf{A}\mathbf{V}\mathbf{V}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, com $\mathbf{U}$ e $\mathbf{V}$ vetores ortogonais, logo

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{3.32}$$

como desejado. □

A decomposição correspondente da equação (3.32) é conhecida como Decomposição em Valores Singulares e os números $\sigma_1, \cdots, \sigma_r$ junto com $\sigma_{r+1} = 0, \cdots, \sigma_n = 0$ são chamados valores singulares de $\mathbf{A}$ e eles são a raiz quadrada positiva dos autovalores, que são não-negativos, de $\mathbf{A}^T\mathbf{A}$, visto que de (3.23) $\mathbf{A}^T\mathbf{A}\mathbf{V}_1 = \mathbf{V}_1\mathbf{S}^2$.

As colunas de $\mathbf{U}$ são chamadas de *vetores singulares à esquerda* de $\mathbf{A}$, e são os autovetores ortonormais de $\mathbf{A}\mathbf{A}^T$, enquanto que as colunas de $\mathbf{V}$ são chamadas de *vetores singulares à direita* de $\mathbf{A}$, e são os autovetores ortonormais de $\mathbf{A}^T\mathbf{A}$.

De modo análogo, a matriz $\mathbf{A}^T$ tem m valores singulares pois $\mathbf{A}^T \in \Re^{n\times m}$, as raízes quadradas positivas dos autovalores de $\mathbf{A}\mathbf{A}^T$. Os r valores singulares não nulos de $\mathbf{A}$ e $\mathbf{A}^T$ são é claro, os mesmos, pois se a equação (3.32) é verdadeira, então,

$$(\mathbf{A})^T = \left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right)^T = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T, \tag{3.33}$$

pois $\mathbf{\Sigma}$ é diagonal, logo $\mathbf{\Sigma}^T = \mathbf{\Sigma}$. A escolha de $\mathbf{A}^T\mathbf{A}$ ao invés de $\mathbf{A}\mathbf{A}^T$ na definição dos valores singulares é arbitrária. Somente os valores singulares potencialmente não-nulos serão de algum interesse real [28].

A matriz $\mathbf{A}$ também pode ser escrita em termos de *vetores singulares* como segue:

$$\mathbf{A} = \sum_{k=1}^{r} \boldsymbol{\sigma}_k \mathbf{u}_k \mathbf{v}_k^T, \tag{3.34}$$

em que r é o posto de $\mathbf{A}$. Se $\mathbf{A}^{m\times n}$ tem n valores singulares, então $\mathbf{A}^T$ tem m valores singulares.

É claro, a partir da definição, que o número de valores singulares não nulos de $\mathbf{A}$ determinam seu posto.

Descreve-se a seguir, algumas propriedades básicas de *SVD*,

- Se $\mathbf{A} \in \Re^{m\times n}$ com valores singulares $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ então

$$\|\mathbf{A}\|_2 = \sigma_1, \tag{3.35}$$

e

$$\|\mathbf{A}\|_F = \left(\sigma_1^2 + \cdots + \sigma_n^2\right)^{\frac{1}{2}}. \tag{3.36}$$

Também se $\mathbf{A}$ é inversível, então $\sigma_n > 0$ e

$$\left\|\mathbf{A}^{-1}\right\|_2 = \frac{1}{\sigma_n}. \tag{3.37}$$

- Se $\mathbf{A} \in \Re^{n\times n}$ é normal (i.e., $\mathbf{A}\mathbf{A}^T = \mathbf{A}^T\mathbf{A}$) com valores singulares $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, então $\sigma_k = |\lambda_k|$, em que $\lambda_k \in \sigma(\mathbf{A})$; $\quad k = 1, 2, \cdots, n$. Em particular, os valores singulares e autovalores de $\mathbf{A} \geq 0$ são idênticos.

- Se $\mathbf{A} \in \Re^{m\times n}$ com valores singulares $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ tem-se,

$$\sigma_k = \min_{d(\mathcal{S})=n-k+1} \max_{\substack{\mathbf{x}\in\mathcal{S} \\ \mathbf{x}\neq 0}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}, \qquad k = 1, \cdots, n. \tag{3.38}$$

em que $\mathcal{S}$ é um subespaço de $\Re^n$ e $d(\mathcal{S})$ é a dimensão de $\mathcal{S}$.

A propriedade apresentada na equação (3.38), conhecida como *minimax*, pode ser usada para mostrar a natureza *bem condicionada* de valores singulares.

- Especificamente se $\mathbf{A} \in \Re^{m\times n}$ com valores singulares $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, e se $\mathbf{A} + \mathbf{E}$ possui valores singulares $\tau_1 \geq \cdots \geq \tau_n \geq 0$, então

$$|\sigma_k - \tau_k| \leq \|\mathbf{E}\|_2, \qquad k = 1, \cdots, n. \tag{3.39}$$

A *SVD* é adequada em análises de matrizes quadradas que admitem inversas, todavia a sua maior utilidade está na análise de matrizes não quadradas e com possíveis deficiências de posto que aparecem por exemplo em problemas de mínimos quadrados lineares [28]. Uma característica que faz *SVD* tão atrativa em aritmética finita é que valores singulares são números *bem condicionados*, como foi visto pela propriedade *minimax*, equação (3.38).

A *SVD* revela muitas informações sobre a estrutura da matriz. Seja A com *SVD* dada pelo Teorema 1, a relação entre r e os valores singulares é

$$\sigma_1 \geq \cdots \geq \sigma_r \geq \sigma_{r+1} = \cdots = \sigma_p = 0. \tag{3.40}$$

Então,

$$\rho(\mathbf{A}) = r, \tag{3.41}$$

$$\mathcal{N}(\mathbf{A}) = span\{v_{r+1}, \cdots, v_n\}, \tag{3.42}$$

$$\mathcal{R}(\mathbf{A}) = span\{u_1, \cdots, u_r\}. \tag{3.43}$$

Teorema 2 *Seja a SVD de* $\mathbf{A} \in \Re^{m\times n}$ *dada pelo Teorema 1. Se* $k < r = \rho(\mathbf{A})$ *e*

$$\mathbf{A}_k = \sum_{i=1}^{k} \boldsymbol{\sigma}_i \mathbf{u}_i \mathbf{v}_i^T, \tag{3.44}$$

então,

$$\min_{\rho(\mathbf{B})=k} \|\mathbf{A} - \mathbf{B}\|_2 = \|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}, \tag{3.45}$$

em que, $\mathbf{B} \in \Re_k^{m\times n}$.

Prova: Desde que $\mathbf{U}^T\mathbf{A}_k\mathbf{V} = diag(\sigma_1, \cdots, \sigma_k, 0, \cdots, 0)$ segue que $\rho(\mathbf{A}_k) = k$ e que,

$$\mathbf{U}^T(\mathbf{A} - \mathbf{A}_k)\mathbf{V} = diag(0, 0, \cdots 0, \sigma_{k+1}, \cdots, \sigma_p), \tag{3.46}$$

e então $\|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}$.

Supondo-se $\rho(\mathbf{B}) = k$ para algum $\mathbf{B} \in \Re^{m\times n}$. Pode-se encontrar vetores ortonormais $\mathbf{x}_1, \cdots, \mathbf{x}_{n-k}$ tais que $\mathcal{N}(\mathbf{B}) = span\{\mathbf{x}_1, \cdots, \mathbf{x}_{n-k}\}$, como visto na equação (3.42).

O argumento da dimensão mostra que,

$$span\{\mathbf{x}_1, \cdots, \mathbf{x}_{n-k}\} \cap span\{\mathbf{v}_1, \cdots, \mathbf{v}_{k+1}\} \neq \{0\}.$$

Deste modo, tomando $\mathbf{z}$ um vetor unitário $\|\mathbf{z}\|_2 = 1$, tal que,

$$\mathbf{B}\mathbf{z} = 0 \quad \text{e} \quad \mathbf{A}_\mathbf{z} = \sum_{i=1}^{k+1} \sigma_i \left(\mathbf{v}_i^T\mathbf{z}\right) \mathbf{u}_i,$$

tem-se,

$$\|\mathbf{A}-\mathbf{B}\|_2^2 \geq \|(\mathbf{A}-\mathbf{B})\,\mathbf{z}\|_2^2 = \|\mathbf{A}\mathbf{z}\|_2^2 = \sum_{i=1}^{k+1} \sigma_i^2 \left(\mathbf{v}_i^T \mathbf{z}\right)^2 \geq \sigma_{k+1}^2. \tag{3.47}$$

Completando a prova do teorema. □

O Teorema 2 diz que o menor valor singular de **A** é a distância, norma quadrática, de **A** para o conjunto de todas as matrizes com deficiência de posto.

Finalmente, se $r_\epsilon = \rho(\mathbf{A}, \epsilon)$, em que ϵ é a precisão na operação, então

$$\sigma_1 \geq \cdots \geq \sigma_{r_\epsilon} > \epsilon \geq \sigma_{r_\epsilon+1} \geq \cdots \geq \sigma_p, \qquad p = \min\{m, n\}, \tag{3.48}$$

B será desprezado e o posto k de $\mathbf{A}_k$, será a ordem do modelo reduzido.

3.3 Redução de Ordem Baseada na SVD

Segundo Araki [1], normalmente a modelagem do sistema custa mais que 50% do total do projeto de sistemas de controle. Daí a importância de algoritmos de identificação eficientes e econômicos em termos computacionais.

Uma técnica que combina o método dos mínimos quadrados com a técnica de *realização mínima* de um modelo no espaço de estados e com um método de transformação Z-S, do espaço discreto para continuo foi proposta por Araki [1]. O método da *realização mínima* baseada na SVD consiste na construção de uma matriz Hankel com uma sequência de resposta ao impulso do sistema, selecionam-se os modos controláveis e observáveis da matriz por SVD, e encontra-se uma *realização mínima balanceada* do modelo no espaço de estados [30] [31] [32].

Como pode ser visto na Figura 3.1, a estimação do modelo reduzido, segundo a abordagem proposta por Araki envolve as etapas descritas a seguir.

1. Adquirir os dados de entrada e saída do sistema e filtrá-los por um filtro passa-faixas para melhorar a precisão da identificação, removendo ruídos e distúbios do sinal;

2. Estimar um modelo ARX utilizando o método LS,

$$A(q)\,y(t) = B(q)\,u(t) + e(t). \tag{3.49}$$

em que, $A(q)$ e $B(q)$ são definidos como nas equações (2.9) e (2.10). Deve-se escolher a ordem n_a e n_b do modelo, alta o suficiente para incluir todos os modos dominantes da planta;

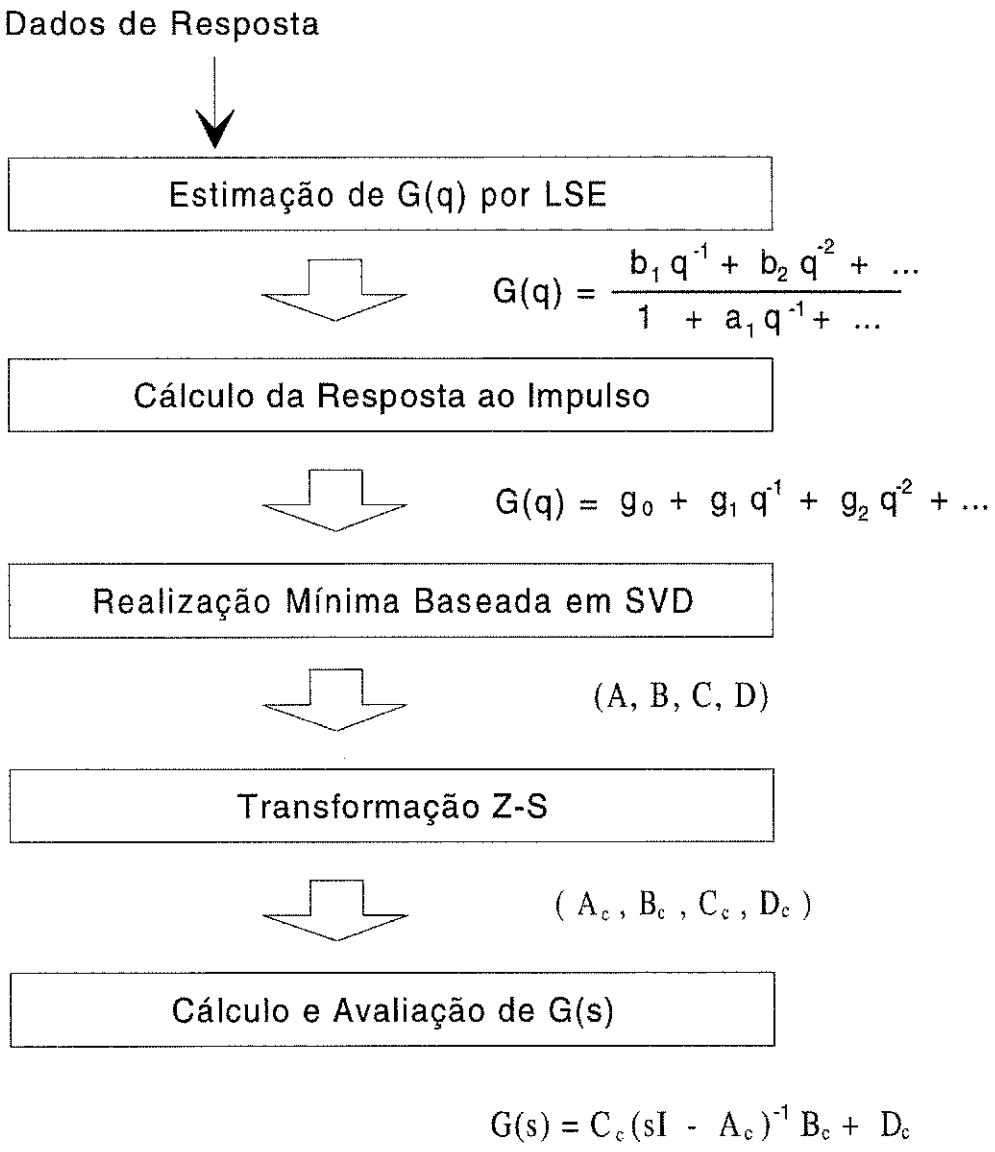

Figura 3.1: Diagrama de blocos do método proposto por Araki

3. Calcular então a resposta ao impulso $\{g_i\}$ do modelo, pelas equações

$$g_0 = \frac{b_0}{a_0}, \tag{3.50}$$

$$g_i = \left(b_i - \sum_{j=1}^{i-1} g_j \, a_{i-j} \right) \frac{1}{a_0}, \qquad i = 1, 2, \cdots, N$$

nas quais, para atender ao algorítmo, N deve ser maior que duas vezes a ordem do modelo estimado, representada aqui por μ, ou seja,

$$N \geq 2\mu.$$

4. Um modelo discreto no espaço de estados,

$$x_{k+1} = Ax_k + Bu_k,$$
$$y_k = Cx_k + Du_k,$$

é obtido pelo método da realização mínima baseada em SVD. O algoritmo é o seguinte, monta-se uma matriz Hankel com a resposta ao impulso encontrada na etapa 3, e calcula-se uma decomposição desta matriz por SVD, chamando esta matriz Hankel de $\mathbf{H}_1$ tem-se,

$$\mathbf{H}_1 = \begin{pmatrix} g_1 & g_2 & \cdots & g_{N-\mu} \\ g_2 & g_3 & \cdots & g_{N-\mu+1} \\ \vdots & \vdots & \ddots & \vdots \\ g_\mu & g_{\mu+1} & \cdots & g_{N-1} \end{pmatrix} = \mathbf{U}\mathbf{\Sigma}\mathbf{V},$$

$$\mathbf{\Sigma} = diag \, [\sigma_1 \; \sigma_2 \; \cdots \; \sigma_n \; \cdots \; \sigma_\mu],$$

com os valores singulares σ_i dispostos em ordem decrescente,

$$(\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \gg \varepsilon \geq \sigma_{\mu+1} \geq \cdots \geq \sigma_\mu),$$

então a ordem mínima do modelo pode ser determinada pelo método apresentado a seguir. Como os valores singulares σ_i possuem informação sobre a ordem do processo, estes são mostrados em um gráfico, tal que a ordem mínima n possa ser determinada visualmente.

Também é possível introduzir uma função de custo $J(m)$, dada por,

$$J(m) = \frac{\left(\sum_{i=0}^{m} \sigma_i^2 \right)}{\left(\sum_{i=0}^{\mu} \sigma_i^2 \right)}, \tag{3.51}$$

em que, $m = 1, 2, \cdots, \mu$ e determinar a ordem n calculando-se,

$$n = \min\{m\}, \qquad \text{tal que} \qquad J(m) > 1 - \varepsilon_0,$$

em que, ε_0 $(0 < \varepsilon_0 \ll 1)$ é um limite. Então, uma *realização mínima* discreta no tempo, do modelo resultante na etapa 3, é obtida pelas seguintes equações :

$$\begin{aligned} \mathbf{A} &= \overline{\mathbf{\Sigma}}^{-1/2}\mathbf{U}_1^T\mathbf{H}_2\mathbf{V}_1^T\overline{\mathbf{\Sigma}}^{-1/2}, \\ \mathbf{B} &= \overline{\mathbf{\Sigma}}^{1/2}\mathbf{V}_1\left[1\ 0\ \cdots 0\right]^T, \\ \mathbf{C} &= \left[1\ 0\ \cdots 0\right]\mathbf{U}_1\overline{\mathbf{\Sigma}}^{1/2}, \\ \mathbf{D} &= g_0. \end{aligned} \tag{3.52}$$

nas quais,

$$\mathbf{U} = [\mathbf{U}_1 \mid \mathbf{U}_2] \quad \text{e} \quad \mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ -- \\ \mathbf{V}_2 \end{bmatrix}.$$

Tais matrizes possuem as seguintes dimensões,

$$\mathbf{U}_{\mu\times\mu},\ \mathbf{U}_{1_{\mu\times n}},\ \mathbf{U}_{2_{\mu\times(\mu-n)}},\ \mathbf{V}_{(N-\mu)\times(N-\mu)},\ \mathbf{V}_{1_{n\times(N-\mu)}},\ \mathbf{V}_{2_{(N-\mu-n)\times(N-\mu)}}. \tag{3.53}$$

nas quais, como foi visto na etapa 3, μ representa a ordem do modelo estimado, N é pelo menos duas vezes maior que μ, e n é a ordem do modelo reduzido; $\overline{\mathbf{\Sigma}}$ é uma matriz diagonal cujos elementos são os valores singulares

$$\overline{\mathbf{\Sigma}} = diag\ [\sigma_1\ \sigma_2\ \cdots\ \sigma_n], \tag{3.54}$$

e $\mathbf{H}_2$ é dado por,

$$\mathbf{H}_2 = \begin{pmatrix} g_2 & g_3 & \cdots & g_{N-\mu+1} \\ g_3 & g_4 & \cdots & g_{N-\mu+2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{\mu+1} & g_{\mu+2} & \cdots & g_N \end{pmatrix}. \tag{3.55}$$

5. Um modelo contínuo no espaço de estados

$$\begin{aligned} \frac{d\,\mathbf{x}(t)}{dt} &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \\ y(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t), \end{aligned} \tag{3.56}$$

é obtido a partir do modelo discreto, a partir de transformações ortogonais T, pelas seguintes equações,

$$\begin{aligned} \overline{\mathbf{A}} &= T^{-1}\mathbf{A}T = diag\ [\nu_1\ \nu_2\ \cdots\ \nu_n], \\ \overline{\mathbf{B}} &= T^{-1}\mathbf{B}, \\ \overline{\mathbf{C}} &= \mathbf{C}T, \end{aligned} \tag{3.57}$$

em que os ν_i representam os autovalores de $\overline{\mathbf{A}}$, depois são calculados certos η_i como segue,

$$\eta_i = \frac{\alpha_i + \beta_i}{\tau}; \qquad i = 1, 2, \cdots, n \tag{3.58}$$

em que, $\nu_i = e^{\alpha_i + j\beta_i}$, e τ é o período de amostragem. Então, um modelo contínuo no espaço de estados ($\mathbf{A}_c$, $\mathbf{B}_c$, $\mathbf{C}_c$, $\mathbf{D}_c$) é obtido por,

$$\begin{aligned} \mathbf{A}_c &= diag\ [\eta_1\ \eta_2\ \cdots\ \eta_n], \\ \mathbf{B}_c &= \left(\overline{\mathbf{A}} - I\right)^{-1} \mathbf{A}_c \overline{\mathbf{B}}, \\ \mathbf{C}_c &= \overline{\mathbf{C}}, \\ \mathbf{D}_c &= \mathbf{D}. \end{aligned} \tag{3.59}$$

6. A função de transferência $G(s)$ é obtida como segue,

$$G(s) = \mathbf{C}_c \left(s\mathbf{I} - \mathbf{A}_c\right)^{-1} \mathbf{B}_c. \tag{3.60}$$

Características desta técnica [1],

- Uma ordem apropriada para a planta é calculada por SVD, considerando os aspectos de controlabilidade e observabilidade.
- Um modelo no espaço de estados é obtido por identificação com dados de resposta curta da planta, em seguida o modelo é transformado para uma função de transferência contínua no tempo.
- Devido à filtragem inicial, o método é robusto contra ruídos e distúrbios nos dados de resposta da planta.

Neste trabalho foram implementadas algumas destas etapas, os passos de 1 a 4. No pacote de rotinas implementada os modelos estimados são apresentados para o usuário na forma discreta, e deste modo, os modelos reduzidos com este método também são apresentados na mesma forma. Os resultados da aplicação desta técnica são apresentados no capítulo 6. Esta técnica foi incluída no pacote de rotinas *ISAC*.

3.4 Identificação Recursiva Baseada em SVD.

Um algoritmo de estimação com atualização da matriz de covariância baseada em decomposição em valores singulares, foi apresentado por Zhang em [15].

A motivação para tal algoritmo é que a equação de atualização de $\mathbf{P}(t)$, no algoritmo mínimos quadrados recursivo (RLS), é sensível a erros de arredondamento e não há

garantia que $\mathbf{P}(t)$ seja sempre *positiva e simétrica definida,* principalmente nos casos em que a excitação é pouco persistente. A sensibilidade a erros de arredondamento faz com que o algoritmo RLS não seja tão robusto quando implementado em computadores, com precisão finita.

Zhang [15] propõe a atualização da matriz de covariância $\mathbf{P}(t)$ baseada em decomposição em valores singulares. Seja a equação de atualização de $\mathbf{P}(t)$ no método RLS,

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \mathbf{K}(t)\left[y(t) - \boldsymbol{\varphi}^T(t)\,\hat{\boldsymbol{\theta}}(t-1)\right], \tag{3.61}$$

$$\mathbf{P}(t) = \left[\mathbf{P}(t-1) - \mathbf{P}(t-1)\,\boldsymbol{\varphi}(t)\,M(t)^{-1}\,\boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)\right]/\lambda(t), \tag{3.62}$$

$$M(t) = \boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)\,\boldsymbol{\varphi}(t) + \lambda(t), \tag{3.63}$$

$$\mathbf{K}(t) = \mathbf{P}(t-1)\,\boldsymbol{\varphi}(t)\;/\;M(t), \tag{3.64}$$

em que, $\lambda(t)$ representa o fator de esquecimento, $\mathbf{P}(t)$ é a matriz de covariância, $\boldsymbol{\varphi}(t)$ é o vetor de regressão e $\mathbf{K}(t)$ é chamada de matriz de ganho.

Substituindo (3.63) em (3.62), nota-se que a atualização da matriz de covariância $\mathbf{P}(t)$ é dada por,

$$\mathbf{P}(t) = \left[\mathbf{P}(t-1) - \mathbf{P}(t-1)\,\boldsymbol{\varphi}(t)\left(\boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)\,\boldsymbol{\varphi}(t) + \lambda(t)\right)^{-1}\boldsymbol{\varphi}^T(t)\,\mathbf{P}(t-1)\right]/\lambda(t). \tag{3.65}$$

Utilizando o lema de inversão matricial [9] [15], tem-se,

$$(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{DA}^{-1}\mathbf{B} + \mathbf{C}^{-1}\right)^{-1}\mathbf{DA}^{-1}. \tag{3.66}$$

Logo, a equação (3.65) pode ser escrita como,

$$\mathbf{P}^{-1}(t) = \mathbf{P}^{-1}(t-1) + \boldsymbol{\varphi}(t)\,\lambda^{-1}(t)\,\boldsymbol{\varphi}^T(t), \tag{3.67}$$

Se $\mathbf{P}(t)$ é uma matriz definida positiva e simétrica, pode-se então decompô-la como,

$$\mathbf{P}^{-1}(t) = \left[\mathbf{Q}(t)\,\mathbf{Q}^T(t)\right]^{-1} \tag{3.68}$$

Com a decomposição da equação (3.68) e utilizando a SVD de $\mathbf{Q}(t)$, encontra-se

$$\begin{aligned}
\mathbf{P}^{-1}(t) &= \left[\mathbf{Q}(t)\,\mathbf{Q}^T(t)\right]^{-1} \\
&= \left[\mathbf{U}(t)\,\boldsymbol{\Sigma}^T(t)\,\mathbf{V}^T(t)\,\mathbf{V}(t)\,\boldsymbol{\Sigma}(t)\,\mathbf{U}^T(t)\right]^{-1} \\
&= \left[\mathbf{U}(t)\,\boldsymbol{\Sigma}^2(t)\,\mathbf{U}^T(t)\right]^{-1},
\end{aligned}$$

$$\mathbf{P}^{-1}(t-1) = \left[\mathbf{U}(t-1)\,\mathbf{\Sigma}^2(t-1)\,\mathbf{U}^T(t-1)\right]^{-1},$$

visto que se $\mathbf{V}(t)$ é ortogonal, $\mathbf{V}^T(t) = \mathbf{V}(t)^{-1}$e se $\mathbf{\Sigma}(t)$ é diagonal, $\mathbf{\Sigma}^T(t) = \mathbf{\Sigma}(t)$.

E assim, a equação (3.67) pode ser escrita como,

$$\begin{aligned}
[\mathbf{U}(t)\,\mathbf{\Sigma}^2(t)\,\mathbf{U}^T(t)]^{-1} &= \left[\mathbf{U}(t-1)\,\mathbf{\Sigma}^2(t-1)\,\mathbf{U}^T(t-1)\right]^{-1} + \boldsymbol{\varphi}(t)\,\lambda^{-1}(t)\,\boldsymbol{\varphi}^T(t) \\
&= \left[\mathbf{U}(t-1)\,\mathbf{\Sigma}^2(t-1)\,\mathbf{U}^T(t-1)\right]^{-1} + \left[\mathbf{U}^T(t-1)\right]^{-1} \times \\
&\quad \mathbf{U}^T(t-1)\,\boldsymbol{\varphi}(t)\,\lambda^{-1}(t)\,\boldsymbol{\varphi}^T(t)\,\mathbf{U}(t-1)\,\mathbf{U}^{-1}(t-1) \\
&= \left[\mathbf{U}^T(t-1)\right]^{-1} \times \\
&\quad \left[\mathbf{\Sigma}^{-2}(t-1) + \mathbf{U}^T(t-1)\,\boldsymbol{\varphi}(t)\,\lambda^{-1}(t)\,\boldsymbol{\varphi}^T(t)\,\mathbf{U}(t-1)\right] \times \\
&\quad \mathbf{U}^{-1}(t-1). \qquad (3.69)
\end{aligned}$$

Fazendo-se $L^2(t) = \lambda^{-1}(t)$.

Considerando a matriz $(2n+1)$ por $2n$,

$$\mathbf{\Lambda} = \begin{bmatrix} L(t)\,\boldsymbol{\varphi}^T(t)\,\mathbf{U}(t-1) \\ \mathbf{\Sigma}^{-1}(t-1) \end{bmatrix},$$

e computando sua SVD,

$$\mathbf{\Lambda} = \overline{\mathbf{U}}(t-1) \begin{bmatrix} \overline{\mathbf{\Sigma}}(t-1) \\ 0 \end{bmatrix} \left[\overline{\mathbf{V}}(t-1)\right]^T.$$

Multiplicando cada lado pela esquerda, por sua transposta, leva a,

$$\begin{aligned}
\mathbf{\Lambda}^T\mathbf{\Lambda} &= \mathbf{\Sigma}^{-2}(t-1) + \mathbf{U}^T(t-1)\,\boldsymbol{\varphi}(t)\,L(t)\,L^T(t)\,\boldsymbol{\varphi}^T(t)\,\mathbf{U}(t-1) \\
&= \overline{\mathbf{V}}(t-1)\left[\overline{\mathbf{\Sigma}}(t-1)\right]^2\left[\overline{\mathbf{V}}(t-1)\right]^T.
\end{aligned}$$

Deste modo pode-se escrever a equação (3.69) como,

$$\begin{aligned}
[\mathbf{U}^T(t)]^{-1}\mathbf{\Sigma}^{-2}(t)\,\mathbf{U}^{-1}(t) &= \left[\mathbf{U}^T(t-1)\right]^{-1}\overline{\mathbf{V}}(t-1)\left[\overline{\mathbf{\Sigma}}(t-1)\right]^2\left[\overline{\mathbf{V}}(t-1)\right]^T\mathbf{U}^{-1}(t-1) \\
&= \left[\left(\mathbf{U}(t-1)\,\overline{\mathbf{V}}(t-1)\right)^T\right]^{-1} \times \\
&\quad \left[\overline{\mathbf{\Sigma}}(t-1)\right]^2\left[\mathbf{U}(t-1)\,\overline{\mathbf{V}}(t-1)\right]^{-1}, \qquad (3.70)
\end{aligned}$$

pois, $\overline{\mathbf{V}}$ é ortogonal, então $\left[\overline{\mathbf{V}}(t-1)\right]^T = \left[\overline{\mathbf{V}}(t-1)\right]^{-1}$.

Comparando-se os dois lados da equação tem-se,

$$\mathbf{U}(t) = \mathbf{U}(t-1)\,\overline{\mathbf{V}}(t-1) \qquad \text{e} \qquad \mathbf{\Sigma}(t) = \left[\overline{\mathbf{\Sigma}}(t-1)\right]^{-1}, \qquad (3.71)$$

e a matriz de ganho pode ser escrita como,

$$\mathbf{K}(t) = \frac{\mathbf{U}(t)\,\mathbf{\Sigma}^2(t)\,\mathbf{U}^T(t)\,\boldsymbol{\varphi}(t)}{\lambda(t)}. \tag{3.72}$$

Assim o algoritmo para a realização da técnica de Zhang é constituido pelos seguintes passos,

1. Monta-se o vetor de regressão como no método RLS

$$\boldsymbol{\varphi}(t) = \left[\; -y(t-1) \;\; \cdots \;\; -y(t-n_a) \;\; u(t-1) \;\; \cdots \;\; u(t-n_b) \;\right]^T. \tag{3.73}$$

2. Calcula-se L e $\mathbf{\Lambda}$, utilizando matrizes identidade de dimensões apropriadas como valores iniciais para $\mathbf{U}$ e $\mathbf{\Sigma}$,

$$L(t) = \sqrt{\frac{1}{\lambda(t)}}, \tag{3.74}$$

$$\mathbf{\Lambda} = \begin{bmatrix} L(t)\,\boldsymbol{\varphi}^T(t)\,\mathbf{U}(t-1) \\ \mathbf{\Sigma}^{-1}(t-1) \end{bmatrix}. \tag{3.75}$$

3. Calcula-se o SVD de $\mathbf{\Lambda}$

$$[\overline{\mathbf{U}}, \overline{\mathbf{\Sigma}}, \overline{\mathbf{V}}] = SVD(\mathbf{\Lambda}). \tag{3.76}$$

4. Calcula-se $\mathbf{U}$ e $\mathbf{\Sigma}$ por (3.71) e $\mathbf{K}(t)$ por (3.72).

5. O vetor de parâmetros $\boldsymbol{\theta}_z$ é dado por,

$$\boldsymbol{\theta}_z(t) = \boldsymbol{\theta}_z(t-1) + \mathbf{K}(t)\left(y(t) - \boldsymbol{\varphi}^T(t)\,\boldsymbol{\theta}_z(t-1)\right). \tag{3.77}$$

Depois retorna-se para o passo 1, e assim por diante até incluir o último dado disponível da planta, pois este é um método recursivo. Os resultados da implementação deste algoritmo para identificação em malha aberta e numa implementação adaptativa, serão apresentados no capítulo 6.

3.5 Conclusões

Os fundamentos da técnica de decomposição em valores singulares foram apresentados neste capítulo. Foram vistos também alguns métodos de redução de ordem do modelo estimado e uma técnica de estimação que utiliza a SVD.

Uma vez obtido um modelo para o processo, pode-se projetar um dispositivo de controle que possa conduzir o sistema para um comportamento adequado. O método de projeto de controladores adotado para este trabalho será apresentado no próximo capítulo.

Capítulo 4

Projeto de Controladores

Com uma descrição da planta na forma paramétrica pode-se então projetar um controlador para conduzir o sistema para o comportamento desejado. No presente capítulo, uma técnica de projeto de controladores PID pelo método de alocação de pólos é apresentada. Uma discussão sobre a representação de modelos contínuos e discretos, também é apresentada.

4.1 Projeto de PID por Alocação de Pólos

Para a utilização de controladores do tipo PID, algumas técnicas disponíveis são: cancelamento de pólos, técnica de alocação de pólos e pólo dominantes. Descreve-se nesta secção apenas a técnica de alocação de pólos utilizando PID, para sistemas de segunda ordem.

Este método de projeto consiste em colocar um bloco de controle na malha do sistema de tal forma que os pólos em malha fechada do sistema processo-controlador estejam localizados em posições especificadas pelo projetista [33]. Estas posições devem ser especificadas de modo a propiciar o comportamento desejado para o sistema em malha fechada.

Seja um modelo de segunda ordem da planta dado pela seguinte função de transferência no domínio da frequência s,

$$G_p(s) = \frac{K_p}{(1+sT_1)(1+sT_2)}. \tag{4.1}$$

Como o controlador PID possui a função de transferência,

$$G_c(s) = \frac{K(1+sT_i+s^2T_iT_d)}{sT_i}, \tag{4.2}$$

a equação característica do sistema em malha fechada pode ser escrita como,

$$s^3 + s^2\left[\frac{1}{T_1} + \frac{1}{T_2} + \frac{K_p K T_d}{T_1 T_2}\right] + s\left[\frac{1}{T_1 T_2} + \frac{K_p K}{T_1 T_2}\right] + \frac{K_p K}{T_i T_1 T_2} = 0. \tag{4.3}$$

E sejam as especificações do comportamento desejado para malha fechada dadas por, ω a frequência natural, ζ o fator de amortecimento, e α um pólo real, com a seguinte equação característica de malha fechada,

$$(s + \alpha\omega)\left(s^2 + 2\zeta\omega s + \omega^2\right). \tag{4.4}$$

Os parâmetros do controlador PID podem então ser calculados comparando-se os coeficientes das equações (4.3) e (4.4),

$$K = \frac{T_1 T_2 \omega^2 (1 + 2\zeta\alpha) - 1}{K_p}, \tag{4.5}$$

$$T_i = \frac{T_1 T_2 \omega^2 (1 + 2\zeta\alpha) - 1}{T_1 T_2 \alpha \omega^3}, \tag{4.6}$$

$$T_d = \frac{T_1 T_2 \omega (1 + 2\zeta) - T_1 - T_2}{T_1 T_2 \omega^2 (1 + 2\zeta\alpha) - 1}, \tag{4.7}$$

em que, K é o ganho proporcional, T_i o ganho integrativo e T_d o ganho derivativo do controlador PID projetado.

No pacote de rotinas implementadas, a especificação do comportamento desejado deve ser definida no domínio continuo no tempo. Esta forma foi adotada por ser uma representação de mais fácil compreensão e manipulação pelo usuário. A seguir apresenta-se uma discussão sobre a representação de sistemas contínuos por modelos discretos e como pode ser obtido um sistema contínuo, partindo-se de uma representação do sistema no dominio discreto.

4.2 Representações de Modelos Contínuos e Discretos

A obtenção de um sistema discreto no tempo equivalente a um sistema contínuo é chamada de *amostragem de um sistema contínuo*. Seja o sistema contínuo,

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{4.8}$$

Dado o estado x em um instante de amostragem t_k, pode-se encontrar a relação entre as variáveis do sistema nos intervalos de amostragem. O estado, em algum tempo futuro t, pode ser determinado resolvendo-se o sistema (4.8), encontra-se então,

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_k)}\mathbf{x}(t_k) + \int_{t_k}^{t} e^{\mathbf{A}(t-s')}\mathbf{B}\mathbf{u}(s')\,ds'$$

O estado no próximo instante de amostragem t_{k+1} é então dado por,

$$\begin{aligned}
\mathbf{x}(t_{k+1}) &= e^{\mathbf{A}(t_{k+1}-t_k)}\mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} e^{\mathbf{A}(t_{k+1}-s')}\mathbf{B}\mathbf{u}(s')\,ds' \\
&= e^{\mathbf{A}(t_{k+1}-t_k)}\mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} e^{\mathbf{A}(t_{k+1}-s')}ds'\,\mathbf{B}\mathbf{u}(t_k) \\
&= \mathbf{\Phi}(t_{k+1},t_k)\,\mathbf{x}(t_k) + \mathbf{\Gamma}(t_{k+1},t_k)\,\mathbf{u}(t_k)
\end{aligned}$$

em que foi considerado, no segundo termo, que $\mathbf{u}$ é constante entre os instantes de amostragem.O vetor de estados no tempo t_{k+1} é uma função linear de $\mathbf{x}(t_k)$ e $\mathbf{u}(t_k)$.

O sistema de equações do modelo, mais conhecido como amostragem *zero-order-hold* (*ZOH*) do sistema (4.8) é então dado por,

$$\begin{aligned}
\mathbf{x}(t_{k+1}) &= \mathbf{\Phi}(t_{k+1},t_k)\,\mathbf{x}(t_k) + \mathbf{\Gamma}(t_{k+1},t_k)\,\mathbf{u}(t_k) \\
\mathbf{y}(t_k) &= \mathbf{C}\mathbf{x}(t_k) + \mathbf{D}\mathbf{u}(t_k)
\end{aligned} \tag{4.9}$$

em que,

$$\begin{aligned}
\mathbf{\Phi}(t_{k+1},t_k) &= e^{\mathbf{A}(t_{k+1}-t_k)} \\
\mathbf{\Gamma}(t_{k+1},t_k) &= \int_0^{t_{k+1}-t_k} e^{\mathbf{A}s}ds\,\mathbf{B}
\end{aligned}$$

Para um tempo de amostragem com período h,

$$t_k = kh$$

o modelo (4.9) pode ser simplificado para,

$$\begin{aligned}
\mathbf{x}(kh+h) &= \mathbf{\Phi}\mathbf{x}(kh) + \mathbf{\Gamma}\mathbf{u}(kh) \\
\mathbf{y}(kh) &= \mathbf{C}\mathbf{x}(kh) + \mathbf{D}\mathbf{u}(kh)
\end{aligned} \tag{4.10}$$

em que,

$$\begin{aligned}
\mathbf{\Phi} &= e^{\mathbf{A}h} \\
\mathbf{\Gamma} &= \int_0^h e^{\mathbf{A}s}ds\,\mathbf{B}
\end{aligned} \tag{4.11}$$

Das equação (4.11) segue que,

$$\begin{aligned} \frac{d}{dt}\mathbf{\Phi}(t) &= \mathbf{A}\mathbf{\Phi}(t) = \mathbf{\Phi}(t)\,\mathbf{A} \\ \frac{d}{dt}\mathbf{\Gamma}(t) &= \mathbf{\Phi}(t)\,\mathbf{B} \end{aligned}$$

As matrizes $\mathbf{\Phi}$ e $\mathbf{\Gamma}$ satisfazem as equações,

$$\frac{d}{dt}\begin{pmatrix} \mathbf{\Phi}(t) & \mathbf{\Gamma}(t) \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \begin{pmatrix} \mathbf{\Phi}(t) & \mathbf{\Gamma}(t) \\ \mathbf{0} & \mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$$

na qual, $\mathbf{I}$ é uma matriz identidade com a mesma dimensão que o número de entradas do sistema, e $\mathbf{0}$ é uma matriz nula com dimensão idêntica a $\mathbf{I}$. As matrizes $\mathbf{\Phi}$ e $\mathbf{\Gamma}$, para um período de amostragem h, podem então ser obtidas a partir da matriz dc blocos,

$$\begin{pmatrix} \mathbf{\Phi}(t) & \mathbf{\Gamma}(t) \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \exp\left\{\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} h\right\} \tag{4.12}$$

Para calcular $\mathbf{\Phi}$ e $\mathbf{\Gamma}$ podem ser utilizados diversos métodos, entre eles, a expansão em séries da exponencial matricial, a transformada de Laplace, o teorema de Cayley-Hamilton ou através da transformação para a forma de Jordan.

A obtenção de um sistema contínuo a partir de um sistema equivalente discreto no tempo pode também ser feita,com base na equação (4.12), segue que,

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \frac{1}{h}\ln\begin{pmatrix} \mathbf{\Phi}(t) & \mathbf{\Gamma}(t) \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{4.13}$$

na qual, $\ln(\cdot)$ é a função logarítmica matricial. Logo o sistema contínuo no tempo pode ser obtido a partir do sistema discreto equivalente, calculando-se o logarítmo de uma matriz de blocos [34].

A forma apresentada acima é também utilizada pelo MATLAB, além de outras [35] [36], para a conversão de sistemas discretos para contínuos, assim como, a conversão de sistemas contínuos para sistemas discretos. Desta forma, para o projeto do controlador, o modelo estimado, que é apresentado para o usuário na sua forma discreta, é convertido para um modelo contínuo utilizando-se a técnica ZOH (zero-order-hold) disponivel em forma de função no MATLAB.

4.3 Conclusões

A técnica de projeto de controladores apresentada neste capítulo foi incluída no pacote de rotinas *ISAC,* como também implementada para fins comparativos entre dois algoritmos de identificação, o método dos mínimos quadrados e o método de estimação com

atualização da matriz de covariância por SVD, abordado na seção 3.4. Os resultados, gráficos e conclusões desta comparação, serão apresentados no capítulo 6.

Foi visto que a especificação do comportamento desejado para o projeto do controlador PID pelo método apresentado torna-se mais fácil com a representação no domínio do tempo. Uma outra opção para o projeto do controlador seria partir da representação discreta do modelo estimado e calcular o controlador baseado em uma especificação discreta do comportamento desejado para a malha fechada.

No próximo capítulo será apresentado o pacote de rotinas que foi implementado com base nas técnicas e métodos apresentados nesta dissertação até o momento. As características principais, assim como os recursos disponíveis e suas formas de acesso serão apresentados. As ferramentas de auxílio ao usuário, que facilitaram à execução de experimentos para modelagem de processos, serão apresentadas.

Capítulo 5

O Sistema ISAC

As técnicas apresentadas nos capítulos anteriores foram implementadas, testadas e integradas em um pacote de rotinas, que é apresentado no presente capítulo. Foi desenvolvido um ambiente para identificação e projeto de controladores, para o ambiente *MATLAB*. São vistas as características principais deste pacote. São apresentadas ainda, as janelas principais do programa, e como o usuário pode editar o sinal de excitação e definir a função de transferência para um experimento com simulação do processo.

Os métodos de identificação que estão disponíveis para o usuário são relatados. É apresentada a tela de identificação do ISAC. As ferramentas de análise em frequência do ISAC também estão descritas neste capítulo. Apresentam-se os procedimentos que devem ser seguidos para que seja feita a redução de ordem do modelo estimado utilizando *SVD*. Para ilustrar a apresentação das janelas principais do pacote, são utilizados os resultados da aplicação do *software* a um protótipo de um secador de grãos industrial, montado no LIEC (Laboratório de Instrumentação e Controle) [37], como também a aplicação a plantas simuladas pelo próprio pacote.

5.1 Descrição Geral do Programa

O ISAC - Identificação de Sistemas Auxiliada por Computador, é um pacote de rotinas com interface gráfica amigável que permite a realização de experimentos para a identificação de plantas como também para o projeto de controladores. O programa foi desenvolvido utilizando-se os recursos de interface gráfica *GUI* (*Graphical User Interface*) do *MATLAB for Windows versão 4.2C* [35], o que permite uma comunicação simples com o usuário, facilitando a navegação pelo programa. O usuário poderá executar facilmente os experimentos de identificação sem ter que utilizar a programação com a linguagem por comandos do *MATLAB*. Deste modo, o pacote fornece conforto ao

usuário de identificação de sistemas com pouca habilidade na linguagem do *MATLAB*. A estrutura básica do ISAC pode ser vista na Figura 5.1.

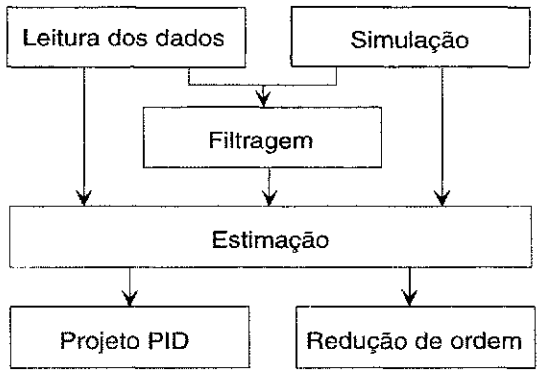

Figura 5.1: Estrutura básica do programa ISAC.

A estrutura básica da Figura 5.1 pode ainda ser apresentada na forma de ítens, explicitando as principais características do pacote de rotinas,

1. Ambiente voltado para o estudo de algorítmos para identificação de sistemas e modelagem de processos, com interface gráfica (*GUI*) que permite a escolha do método de identificação e edição dos sinais de excitação do sistema.
2. Rotinas para filtragem dos dados do sistema em estudo, redução de modelos, cálculo de pólos e zeros do sistema, análise em frequência, diagrama de Bode, diagrama de Nyquist, resposta do modelo estimado ao sinal de excitação aplicado no processo real durante o experimento.
3. Rotinas para visualização de sinais e dados do sistema (sinais de entrada e saída, parâmetros estimados). Armazenamento dos dados do experimento.
4. Possui métodos de identificação paramétricos não-recursivos, e redução da ordem dos modelos estimados utilizando *SVD*.
5. Pode ser feita a estimação paramétrica em um conjunto de dados no formato *MATLAB* ou em simulações de plantas editadas no próprio ISAC.
6. O sinal de excitação e a função de transferência podem ser editados e modificados pelo usuário.
7. Possui rotina para projeto de controladores PID, baseada no método de alocação de pólos.
8. A validação do modelo estimado pode ser feita com base em uma função de custo ou pela resposta temporal do modelo.

O ISAC limita-se a experimentos para modelagem matemática de processos de uma entrada e uma saída, também conhecidos como *SISO*. Este pacote de rotinas é composto por programas na linguagem do *MATLAB*. Tais programas estão interligados como mostra a Figura 5.2.

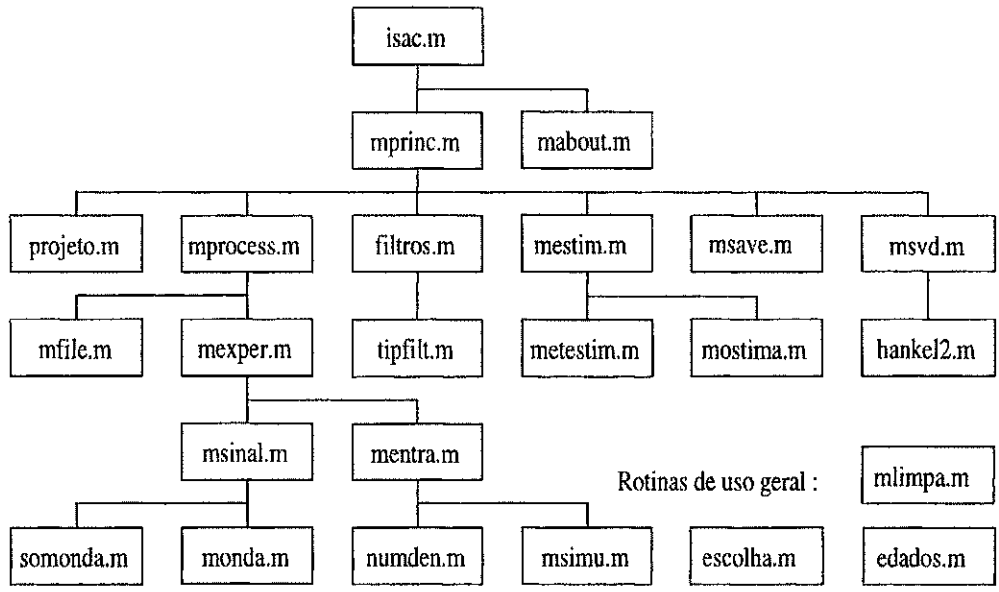

Figura 5.2: Fluxograma do programa ISAC.

O diagrama apresentado na Figura 5.2 foi elaborado com base no fluxo de execução do pacote de rotinas. Para ter acesso aos recursos do ISAC o usuário deve executar primeiro o programa chamado *isac.m*. A execução dos demais programas do pacote não é feita diretamente pelo usuário, mas pelos próprios programas do pacote. Entretanto, a escolha do programa a ser executado depende da função solicitada pelo usuário. Assim, seguindo a hierarquia do diagrama pode-se perceber que os próximos programas a ser executados serão o *mprinc.m* e o *mabout.m*. A seguir poderá ser executado um dos programas da terceira linha do diagrama, ou seja, *projeto.m*, *mprocess.m*, *filtros.m*, *mestim.m*, *morder.m*, *msvd.m*, ou *msave.m*, e assim por diante. A execução dos programas do diagrama da Figura 5.2 é feita de modo bidirecional e sequencial, por exemplo, o item *mprocess.m* pode executar um dos dois programas que estão imediatamente abaixo, *mfile.m* ou *mexper.m*, ou retornar para o programa *mprinc.m* que está logo acima. Uma descrição sucinta da função destes programas é apresentada a seguir,

1. ISAC.M - Este é o programa que deve ser inicialmente executado com a tarefa de abrir a janela principal e permitir o acesso às funções e recursos do pacote ISAC. O programa monta um menu de barras para permitir este acesso ao usuário.

Ao acionar este programa o usuário também terá ao seu dispor uma janela de opções com botões do tipo "push" do MATLAB. Somente este programa pode ser executado diretamente pelo usuário, os demais são chamados automaticamente de acordo com as funções solicitadas pelo usuário.

2. MABOUT.M - Ao ser executado abre uma janela informativa sobre o ISAC. Poderá ser utilizada posteriormente para a implementação do help *on-line* do programa.

3. MPRINC.M - Tem a função de criar a janela principal do menu com botões que permitem o acesso aos recursos do ISAC.

4. MSAVE.M - Chamado pelo programa *mprinc.m,* abre uma janela para armazenamento dos dados de entrada e saída em arquivo de dados.

5. MSVD.M - Programa que abre uma janela com botões que dá acesso às funções de redução de ordem do modelo estimado, utilizando SVD. Permite também a redução automática ou visual do modelo estimado.

6. MESTIM.M - Este programa abre uma janela que permite escolha das opções e métodos para a estimação da planta. A ordem inicial do modelo pode ser escolhida neste programa.

7. METESTIM.M - Programa com a rotina de que permite a implementação do método de estimação escolhido pelo usuário. Os métodos disponíveis são: método dos mínimos quadrados e método das variáveis instrumentais.

8. MOSTIMA.M - Apresenta os parametros do modelo estimado. Permite também que sejam traçados os diagramas de Bode e Nyquist do modelo, calculados os pólos do modelo e traçada a resposta do sistema ao sinal de excitação utilizado. A rotina de validação do modelo está implementada neste programa.

9. MPROCESS.M - Ao ser executado, abre uma janela com botões que permite que seja escolhida a origem dos dados que serão utilizados nos experimentos de identificação. As opções disponíveis são de leitura de arquivo de dados ou de simulação dos dados da planta.

10. PROJETO.M - Programa que permite o projeto de controladores PID pelo método de alocação de pólos, e utilizando o modelo estimado.

11. FILTROS.M - Ao ser executado, abre uma janela para filtragem dos dados. Os filtros disponíveis são do tipo Butterworth, Chebychev, Bessel ou Elíptico e pode ter a caracteristica de resposta passa-baixa, passa-alta, passa-faixa ou rejeita-faixa de acordo com a escolha do usuário.

12. TIPFILT.M - Programa que contém as rotinas de implementação do filtro selecionado pelo usuário na janela do programa FILTROS.M.

13. MFILE.M- Ao ser executado, abre uma janela para abertura de arquivos com dados amostrados da planta.

14. MEXPER.M - Ao ser executado, abre uma janela com botões que dão acesso aos programas MSINAL.M e MENTRA. Nestes o usuário poderá editar o sinal de excitação ou a função de transferência da planta simulada.

15. MSINAL.M - Ao ser executado, abre uma janela para permitir a construção de um sinal de excitação que será utilizado para experimento de identificação em uma simulação de um processo. Permite a edição de sinais com formato básico, quais sejam onda quadrada, dente-de serra, senoidal e degrau. Este programa permite a modificação de parâmetros do sinal, tais como, tempo de amostragem, ponto inicial, tempo final, entrada da amplitude, período do sinal. Cria botões para traçar ou apagar gráficos. Permite ainda, somar curvas de diferentes formas e armazenar em uma variável para uso posterior.

16. MONDA.M - Programa com as rotinas de implementação da curva editada pelo usuário.

17. SOMONDA.M - Este programa permite que mais de uma forma de onda seja somada para gerar um sinal de excitação com formato especificado pelo usuário.

18. MENTRA.M - Permite edição da função de transferência da planta que será simulada e utilizada para o experimento de identificação.

19. NUMDEN.M - Programa que gerencia a entrada de dados para edição da funçâo de transferência da planta que será simulada. Permite somente a entrada de números, e possui funcionamento idêntico ao do programa EDADOS.M, só que edita vetores ao invés de números.

20. MSIMU.M - Utiliza o numerador e denominador editados no programa MENTRA.M para simular a planta. Gera ruído aleatório e insere no sistema para tenha uma maior aproximação de um processo real. O usuário pode modificar

este programa para mudar a dinâmica do ruído que ataca o sistema na simulação. O ruído adicionado na saída da planta simulada pode ter sua amplitude modificada na janela MENTRA.M.

21. HANKEL2.M - Programa com rotina que monta a matriz Hankel que é utilizada na técnica de redução de ordem baseada em decomposição por valores singulares.

22. EDADOS.M - Rotina de uso genérico pelos demais programas do pacote ISAC, que permite a entrada de valores numéricos nas rotinas do ISAC, impedindo a entrada acidental de letras, já que isto provocaria uma suspensão anormal da execução do programa. Números no formato exponencial do *MATLAB* são aceitos. Exemplo: $(1e-1)$, que representa o valor $(0,1)$; ou $(3.1e2)$, para representar o valor $(310,0)$.

23. MLIMPA.M - Rotina de uso genérico pelos demais programas do pacote ISAC, que apaga as janelas que estiverem abertas, exceto a janela principal.

24. ESCOLHA.M - Rotina que gera os menu com botões para acesso aos recursos principais do ISAC. Esta rotina é utilizada por diversos programas do pacote.

5.2 A Janela Principal do ISAC

Na janela principal permite o acesso aos recursos disponíveis no ISAC. Nesta janela o usuário dispõe de um *menu* de barras, e de um *menu* de botões, para acessar os recursos e funções do programa. Na Figura 5.3 é apresentada a janela principal do ISAC.

Os dados, entrada e saída do sistema, que serão utilizados para os cálculos de identificação podem ser obtidos basicamente de duas formas, quais sejam,

1. Lendo-se arquivos de dados amostrados do processo, no formato *MATLAB*, ou

2. Executando-se a simulação de um processo.

Na Figura 5.4 é apresentada a janela na qual o usuário deverá informar o nome e a localização do arquivo de dados amostrados que será carregado no espaço de trabalho do *MATLAB*, para a realização do experimento. O pacote de rotinas possui uma biblioteca de arquivos de dados amostrados de um protótipo de secador de grãos industrial. Tais arquivos podem ser carregados pelo usuário para a execução das rotinas de identificação de sistemas, e estão armazenados com o nome de *cba1.mat, cba2.mat, ... ,cba6.mat.*

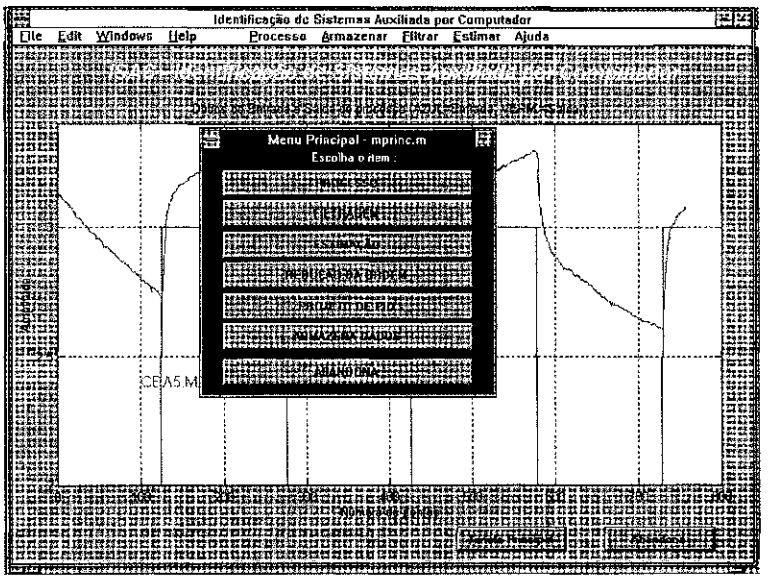

Figura 5.3: Janela principal do programa ISAC.

Caso o usuário deseje criar seus próprios arquivos de dados, deverá fazê-lo seguindo o padrão adotado pelo ISAC. Este padrão estabelece que o arquivo de dados deve conter apenas o sinal de entrada e o sinal de saída amostrados do processo. Tais sinais devem estar dispostos em duas colunas, em que os dados de entrada devem ser denominados "u" e os dados de saída como "y". Os arquivos devem seguir o padrão de arquivos com formato *MATLAB* [36] [35].

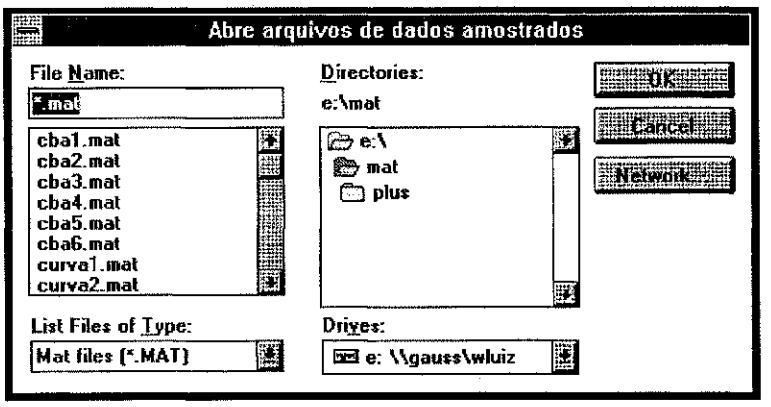

Figura 5.4: Janela para carregar dados amostrados.

No caso dos dados de experimentos com sistemas reais, a planta pode ser um sistema simples como por exemplo, um filtro ativo com o sinal de saída convenientemente amplificado, ou protótipos de processos. Nos testes que serão apresentados ao longo deste capítulo foi utilizado um protótipo de um secador de grãos industrial. Quaisquer outros processos *SISO* podem ser utilizados para a obtenção de dados. Além disto,

o pacote possui arquivos de dados amostrados de um protótipo de secador de grãos industrial [37], no formato *MATLAB*, e podem ser utilizados como exemplo no ambiente ISAC. Para o caso dos dados serem gerados por simulação, o usuário deverá definir a função de transferência do processo e a forma de onda do sinal de excitação para o experimento. Assim, tem-se bastante versatilidade para a obtenção de dados para o experimento de identificação.

O menu de barras da janela principal torna-se bastante útil para os usuários que desejam uma maior velocidade no acesso às funções do ISAC. Neste caso as janelas de menu, que são mais lentas, não são ativadas. O código escrito para implementar o menu de barras está implementado no programa *isac.m* e pode ser visto no *Apêndice A*, no final da presente dissertação.

5.3 Edição da Forma de Onda

Quando realizando identificação com processo simulado, o ISAC permite que os sinais de excitação do processo sob teste sejam editados. A edição dos sinais de excitação para a planta pode ser feita pelo usuário na janela de edição da forma de onda, implementada com os programas *msinal.m*, *monda.m* e *somonda.m*. Podem ser utilizadas formas de onda básicas: onda quadrada, dente-de serra, senóide ou degrau. Na Figura 5.5, é mostrada a janela de edição do sinal de excitação.

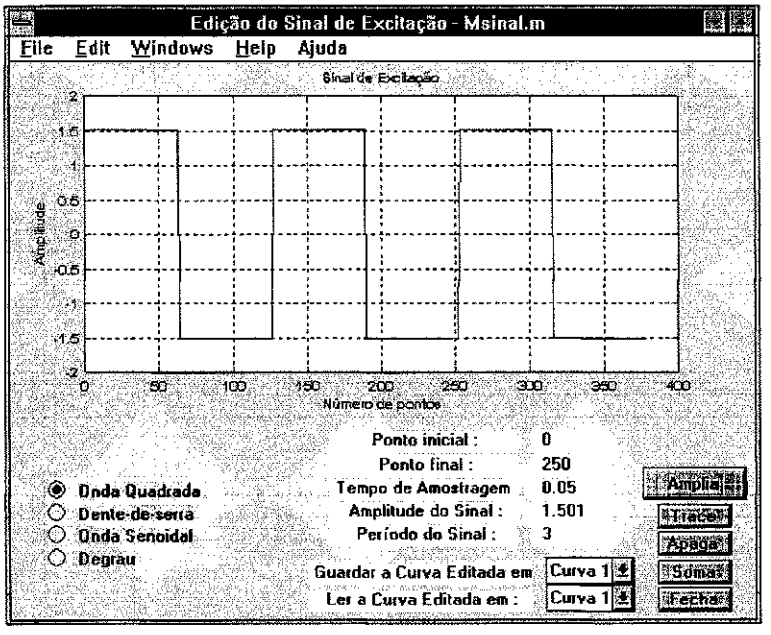

Figura 5.5: Janela para edição do sinal de excitação.

As rotinas para criar as formas de onda foram implementadas no programa *monda.m*

e são apresentadas no *Apêndice A.*

Podem ainda, ser modificados pelo usuário, o período do sinal, a amplitude, o tempo de amostragem, e os pontos inicial e final da curva. Na janela de edição do sinal de excitação pode-se ainda, combinar segmentos das formas de onda básicas para criar sinais mais elaborados. As rotinas que executam a soma de segmentos de sinais estão implementadas no programa *somonda.m.*

A curva pode ser ampliada com o botão *Amplia,* para observação de detalhes do gráfico do sinal na tela. Uma vez ativada a função *Amplia,* o usuário deve especificar, com o *mouse*, a área do sinal que deseja ampliar, na Figura 5.6 pode ser visto um exemplo de marcação de área a ser ampliada, entre os pontos 240 e 380 do gráfico. O botão esquerdo do *mouse* é utilizado para ampliar, enquanto que o botão direito deve ser utilizado para retornar a escala do gráfico à situação anterior. Para finalizar a função de ampliação deve-se utilizar o botão *Congela.* A rotina de ampliação foi implementada em mais de um programa do pacote e pode ser vista no final desta dissertação.

Pode-se também guardar as curvas e informações sobre os sinais editados em um arquivo de dados ou ler de um arquivo, sinais para excitação. As informações que podem ser armazenadas são cada um dos pontos da curva, o ponto inicial, ponto final, tempo de amostragem, amplitude e período do sinal.

Para a entrada dos parâmetros das curvas, a solução utilizada foi criar um programa que pode ser chamado a cada vez que o ISAC necessitar de entrada de dados, este programa foi chamado de *edados.m.*

Nota-se que na rotina há restrições propositais para a inserção de letras, pois estas não teriam uma interpretação numérica pelas funções do *MATLAB.* Tais restrições foram inseridas para proteção do pacote de rotinas contra erros de execução, o que causaria uma parada não desejada na execução do programa, e consequentemente a necessidade de reinicializar o programa novamente.

No programa *msinal.m* pode ser visto um exemplo de utilização do programa *edados.m.*

5.4 Edição da Função de Transferência

A função de transferência da planta a ser simulada pode ser editada de forma simples através da interface amigável do ISAC, Figura 5.6. O usuário deve inserir os coeficientes da função de transferência definindo o numerador e denominador na forma de vetores com coeficientes em ordem decrescente. A função de transferência é editada no domínio

Z. Por exemplo, a função de transferência

$$G(q) = \frac{0,1q^{-1} + 0,05q^{-2}}{1 - 1,5q^{-1} + 0,7q^{-2}}, \tag{5.1}$$

deverá ser editada como apresentado na Figura 5.6.

Na mesma janela pode ser observada a resposta da planta editada ao sinal de excitação criado na janela da Figura 5.5. O botão *Amplia* também está presente nesta tela, porém é apresentado com o nome "Congela" porque foi ativado para a ampliação de parte do gráfico. A área demarcada com um retângulo, entre os pontos 240 e 380 é que será ampliada.

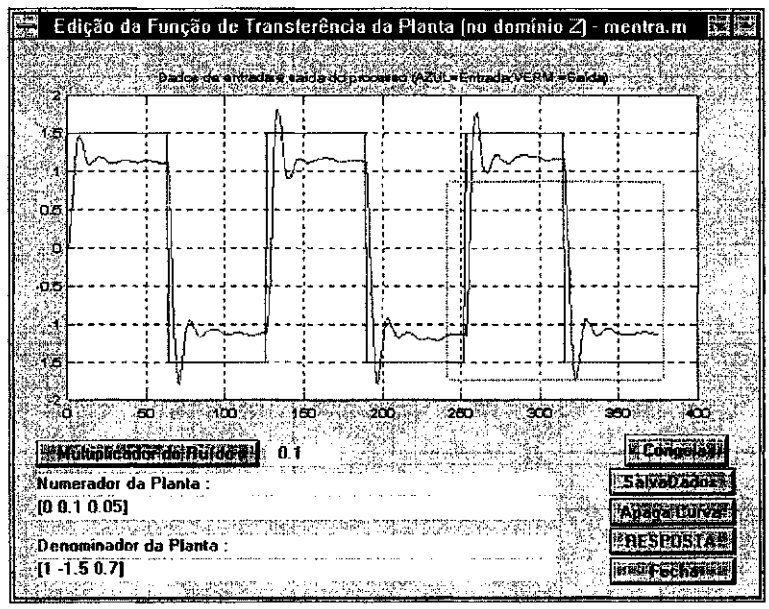

Figura 5.6: Janela de edição da função de transferência a ser simulada.

As rotinas que montam a janela de edição da função de transferência estão implementadas no programa *mentra.m* e são apresentadas ao final da dissertação.

A simulação das plantas editadas são feitas com base no modelo ARMAX, com o ruído adicionado a saída do sistema. Na janela de edição da função de transferência o usuário pode modificar a amplitude do ruído, alterando o valor do campo "multiplicador do ruído". Para modificar a dinâmica do ruído o usuário poderá editar o programa *msimu.m,* alterando o modelo do ruído ou simplesmente alterando a função de transferência do filtro do ruído, presente na rotina.

5.5 Filtragem dos Dados

Antes de aplicar os algorítmos de identificação, pode ser feita uma filtragem dos dados, para eliminar perturbações e melhorar a autocorrelação dos dados. O tratamento

deverá eliminar anormalidades do conjunto de dados, tais como pontos fora da curva e alguma classe de ruídos não desejados. Para o tratamento dos dados pode-se implementar filtros com métodos estatísticos. Para isto estão disponíveis os filtros do tipo Butterworth, Chebychev, Bessel ou Elíptico. A ordem do filtro, frequências de corte, nível de ripple e *condições iniciais*, podem ser modificadas pelo usuário. O diagrama de Bode do filtro aplicado também pode ser visualizado. Para filtrar os sinais utiliza-se a janela mostrada na Figura 5.7.

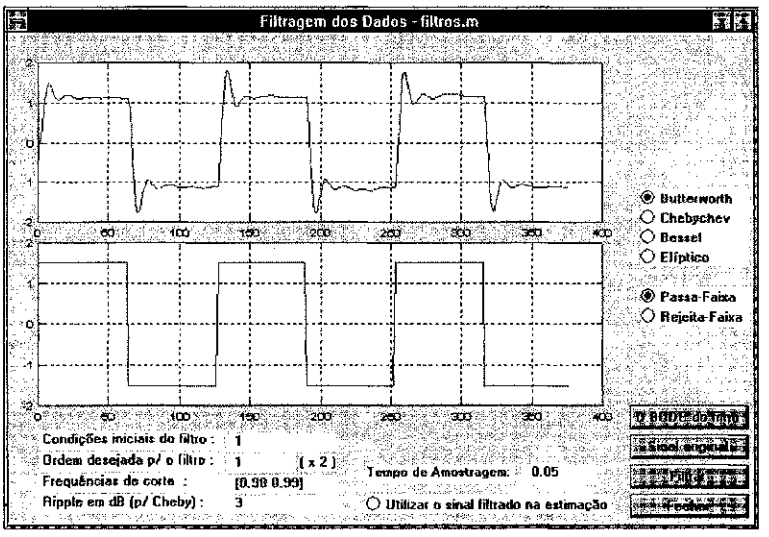

Figura 5.7: Janela para filtragem dos dados.

Os filtros foram implementados com base em funções já existentes no *MATLAB*. Uma característica especial destas funções é que a ordem do filtro depende também do tamanho do vetor de especificação da frequência de corte. Na janela de filtragem do ISAC, se o tamanho do vetor que determina a frequência de corte, especificada no campo denominado *Frequências de corte*, for maior que 1 então o filtro terá a ordem multiplicada por um fator de 2. Por outro lado, se o tamanho deste vetor de frequência for igual a 1, então o filtro terá a ordem especificada pelo campo: *Ordem desejada p/ o filtro.*

Quando o vetor com as frequências de corte for maior que dois ($[a\ b]$), deverá ser digitado pelo usuário com colchetes delimitadores no mesmo padrão numérico que é utilizado pelo *MATLAB*. Os filtros foram implementados no programa *filtros.m*, e para a escolha do tipo de filtro foram implementadas as rotinas do programa *tipfilt.m*.

5.6 Métodos de Estimação e Tipo de Sistema

Para a construção do modelo, o ISAC utiliza-se de dois métodos de identificação que serão listados adiante. Estes métodos utilizam-se dos dados de entrada e saída para criar um modelo e comparar a saída deste modelo com a saída real do processo, por meio de uma função quadrática. Deste modo o algorítmo ajusta o modelo gradativamente até obter um que seja o mais idêntico possível ao modelo real.

Com os dados da planta disponíveis, o usuário poderá escolher qual o método de estimação que irá utilizar. As opções disponíveis são,

1. Método dos mínimos quadrados, e
2. Método das variáveis instrumentais.

As rotinas para estimação foram implementadas pelo método não-recursivo de estimação do modelo e estão presentes no programa *metestim.m.*

Os arquivos de dados previamente adquiridos para a execução de experimentos de identificação, devem obedecer ao formato padrão utilizado pelo *MATLAB.* Este formato é de fácil compreensão e pode ser encontrado nos manuais de referência do *MATLAB* [35] [36]. O experimento de identificação pode ser feito com uma planta real ou com um processo simulado pelo ISAC. No último caso, a função de transferência no domínio discreto pode ser editada em uma janela do programa.

Um sistema pode ser representado por diversos modelos, dos mais simples aos mais complexos e detalhados. O usuário poderá escolher a ordem do modelo que deseja encontrar. A maior ordem que o usuário pode utilizar no programa é 14, porém modelos de plantas com ordem nesta faixa não tem aplicações práticas, e portanto dificilmente será necessário ultrapassar este limite. A janela mostrada na Figura 5.8 deve ser utilizada para definir o método de estimação, a ordem desejada para o modelo e obter o modelo estimado, pode-se ainda utilizar ferramentas de análise em frequência para investigar o modelo obtido. A validação do modelo encontrado poderá ser feita com base nos gráficos de resposta do sistema ou com a utilização do erro médio quadrático entre o dado de saída e a saída do modelo estimado, fornecido pela tela de estimação do ISAC. Em ambos os casos, a decisão sobre o modelo que será utilizado é do usuário.

Obtido o modelo pode-se dispor das ferramentas clássicas de análise em frequência; diagrama de Bode, diagrama de Nyquist, cálculo de pólos e zeros do sistema, ou obter a resposta do sistema ao sinal de entrada que foi utilizado na estimação. O programa executa rotinas de redução de ordem do modelo baseadas em *SVD.*

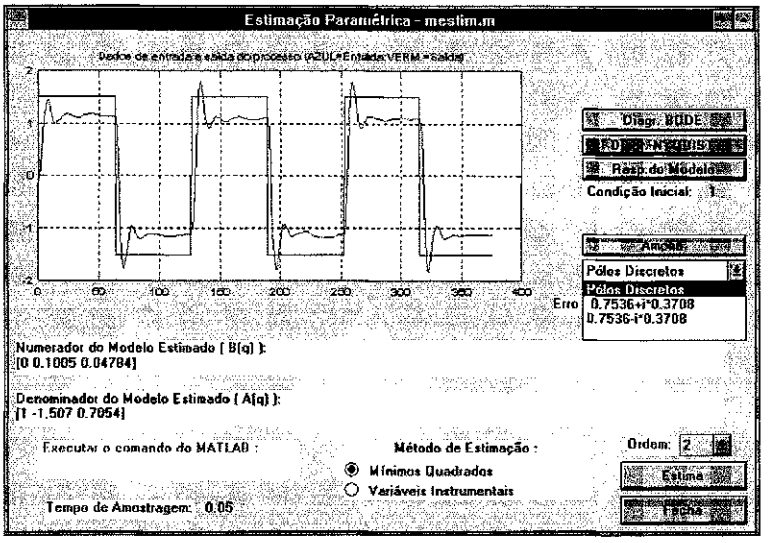

Figura 5.8: Janela para identificação do processo.

5.7 Ordem do Modelo versus Projeto do Controlador

A definição do que seria um bom modelo matemático para uma planta depende muito das necessidades do projeto no qual será aplicado. Para as aplicações em geral, são desejados modelos de ordem reduzida ao invés de modelos mais complexos. A utilização de modelos simples permite vantagens em relação aos controladores projetados, pois estes também serão mais simples. Assim, a ordem do modelo está muito ligada a satisfação das necessidades básicas do projeto para o qual está orientado o experimento de identificação. Muitas vezes pensa-se que um bom modelo é aquele que se aproxima mais do processo real. Para controle isto nem sempre é verdadeiro. Um bom modelo, do ponto de vista de um projetista de controladores, é aquele que atende com simplicidade às necessidades básicas para o projcto. Nesta área, simplicidade e economia são palavras bastante correlacionadas.

O ISAC permite a estimação com a ordem da planta inicialmente maior que a desejada, como uma estimação mais grosseira, um ponto de partida. Em seguida, com o uso da janela apresentada na Figura 5.9, pode ser feita a redução da ordem do modelo utilizando-se de técnicas de Decomposição em Valores Singulares. A redução da ordem é feita utilizando-se a técnica de Decomposição em Valores Singulares, baseada na resposta ao impulso do sistema modelado, como apresentado no capítulo 3.

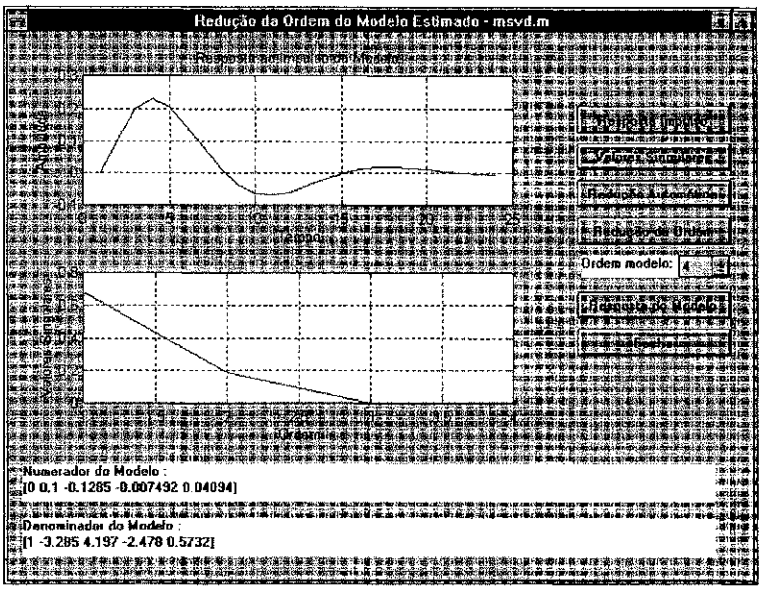

Figura 5.9: Janela para redução da ordem de uma planta estimada para modelo de quarta ordem.

5.7.1 Redução da Ordem do Modelo por SVD

O usuário do ISAC pode modificar a ordem inicial do modelo a ser estimado, para aquela que desejar. Após a estimação de um modelo com a ordem designada, a redução da ordem poderá ser feita utilizando-se a técnica de Decomposição em Valores Singulares [1], baseada na resposta ao impulso do sistema modelado. O programa monta uma matriz Hankel com os valores da resposta ao impulso do modelo estimado e decompõe em valores singulares. A redução do modelo pode ser obtida automaticamente ou visualmente pelo usuário. Através de uma função de custo com parâmetros previamente definidos pelo usuário pode-se obter automaticamente a ordem do modelo. A análise gráfica da curva com os valores singulares do modelo permite também que o usuário obtenha, empiricamente, a ordem reduzida do modelo estimado.

Para ilustrar a operação da janela de redução da ordem do modelo é apresentada a seguir uma sequência de procedimentos que foram utilizados para encontrar o modelo de um processo. Inicialmente tem-se que gerar os dados de um sistema linear. O sistema utilizado possui a seguinte função de transferência no domínio discreto,

$$\frac{B(q)}{A(q)} = \frac{0,1q^{-1} + 0,05q^{-2}}{1 - 1,5q^{-1} + 0,7q^{-2}}$$

Na Figura 5.10 é apresentada a tela do ISAC que foi utilizada para a simulação do processo.

Após a simulação da planta foi feita uma estimação de modelo para quarta ordem, e em seguida executou-se as rotinas de redução de ordem. Na Figura 5.9, é apresentada

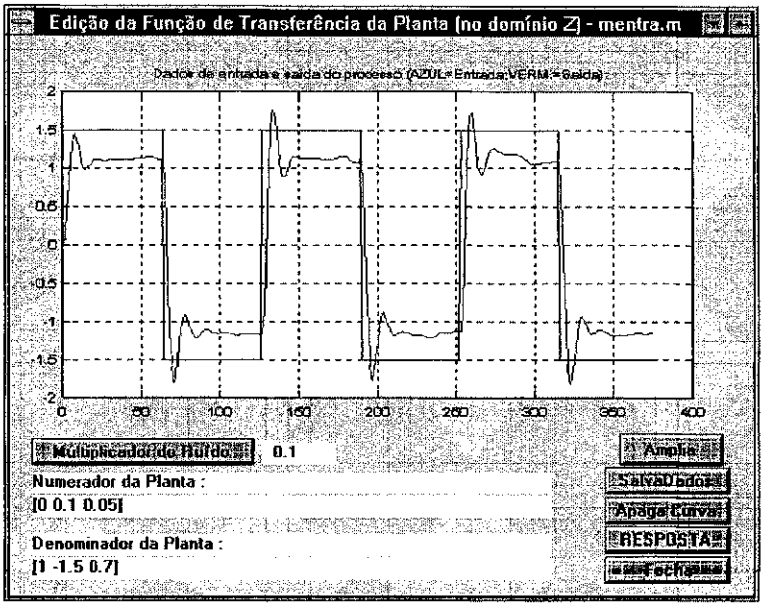

Figura 5.10: Janela para simulação de planta.

a tela inicial executada após a estimação, com o modelo estimado antes da redução. A resposta temporal deste modelo pode ser vista na Figura 5.11,

Para reduzir a ordem do modelo pode ser utilizado o modo automático, no qual o usuário não precisa informar qual a ordem desejada para o modelo reduzido, ou o modo em que o usuário decide qual será esta ordem. O modo de redução automática utiliza a função de custo da equação (3.51). Nas Figuras 5.12 e 5.13, são mostradas a janela com o modelo reduzido usando-se o botão automático e a tela com resposta do modelo reduzido, respectivamente. Como pode ser visto, a ordem do modelo estimado foi reduzida para segunda ordem, e este valor é apresentado para o usuário, juntamente com a função de transferência do modelo reduzido.

Caso seja desejada uma redução para um modelo de outra ordem o usuário deve selecionar a ordem e acionar o botão "Redução da Ordem", na janela. A ordem desejada deve ser menor que a ordem do modelo estimado, caso isto não seja obedecido, o ISAC informa o erro ao usuário. Na Figura 5.14, é apresentada a tela com o modelo reduzido para primeira ordem. O usuário pode escolher a ordem desejada, que vai de 1 a 14, como pode ser visto na barra amarela mostrada na Figura. A resposta temporal do modelo reduzido para primeira ordem é mostrada na Figura 5.15. Como pode ser visto, para o modelo de primeira ordem o sistema perdeu parte da sua dinâmica, fato que pode ser concluido observando-se que a resposta apresenta-se menos oscilatória que a resposta do processo. A tela apresenta o sinal de excitação utilizado, a resposta do processo e a resposta do modelo reduzido.

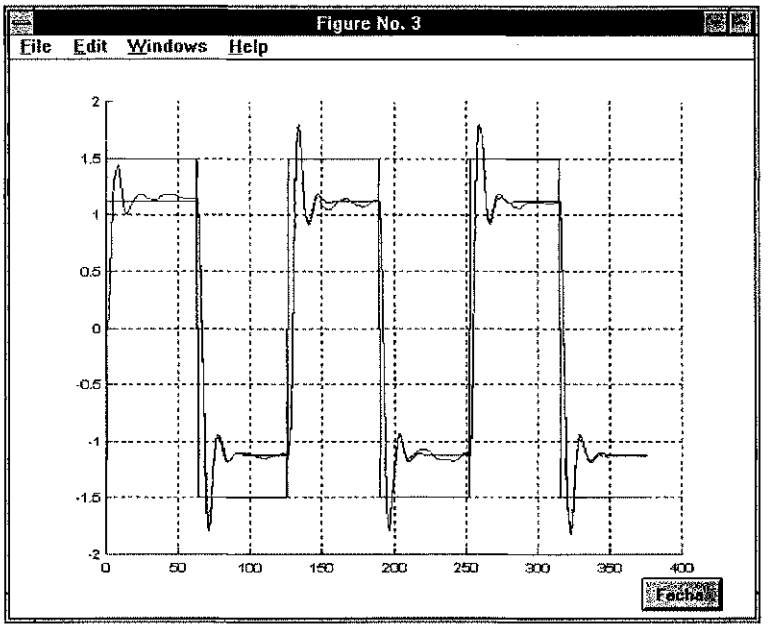

Figura 5.11: Resposta temporal do modelo estimado.

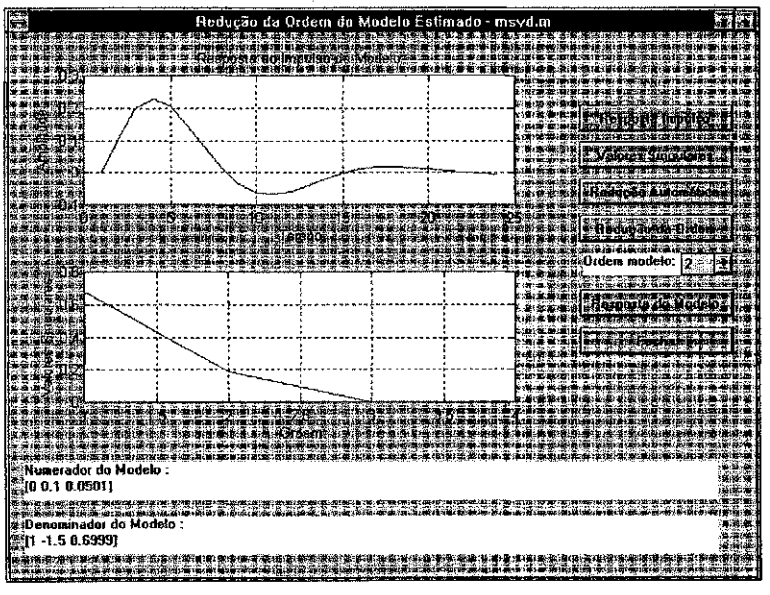

Figura 5.12: Janela com a redução automática da ordem do modelo estimado.

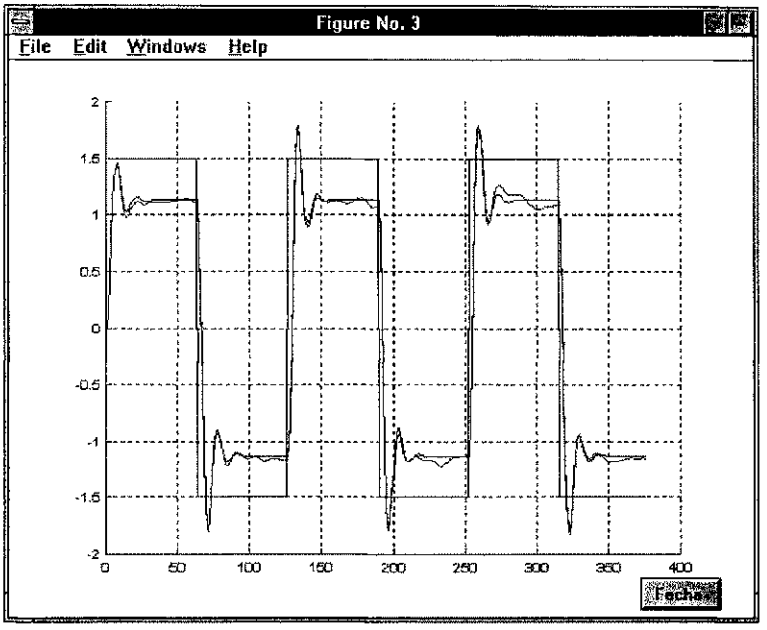

Figura 5.13: Resposta temporal do modelo reduzido.

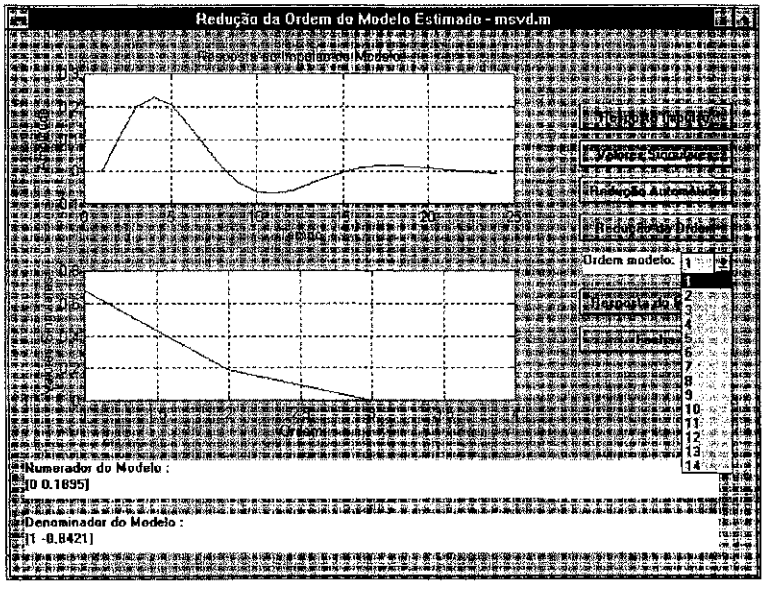

Figura 5.14: Janela com a redução da ordem do modelo, informando-se o valor da ordem.

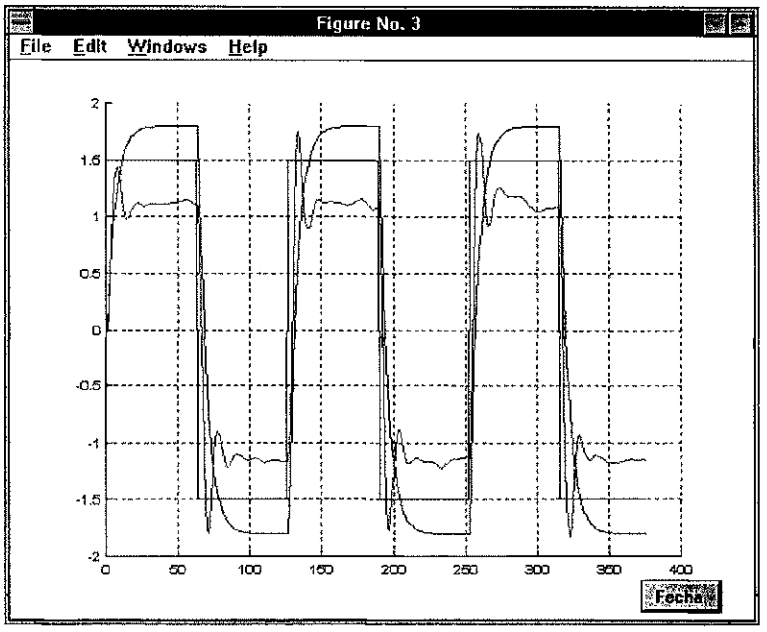

Figura 5.15: Resposta temporal do modelo reduzido para primeira ordem.

Para a redução de ordem foi criado o programa *msvd.m.*

5.7.2 Projeto de Controladores

Com o modelo estimado do processo pode ser feito o projeto de um controlador PID. A rotina utiliza-se da técnica de alocação de pólos descritas no capítulo 3. Para a utilização deste método é necessário que seja obtido com antecedência, um modelo de segunda ordem com a janela de estimação, ou com a janela de redução da ordem do modelo estimado.

O usuário pode definir as características desejadas para o sistema em malha fechada através das especificações dadas por, ω a frequência natural, ζ o fator de amortecimento, e α um pólo real, com a seguinte equação característica de malha fechada,

$$(s + \alpha\omega)\left(s^2 + 2\zeta\omega s + \omega^2\right) \tag{5.2}$$

A função de transferência é dada por,

$$F.T._{desejada} = \frac{\omega^3\alpha}{(s + \alpha\omega)\left(s^2 + 2\zeta\omega s + \omega^2\right)}$$

Os parâmetros do controlador são então ser calculados pelas equações (4.5), (4.6) e (4.7).

Na Figura 5.16 é apresentada a janela para o projeto de controladores PID.O sinal de referência padrão que foi utilizado varia de 1 a -1. O botão *Amplia* também

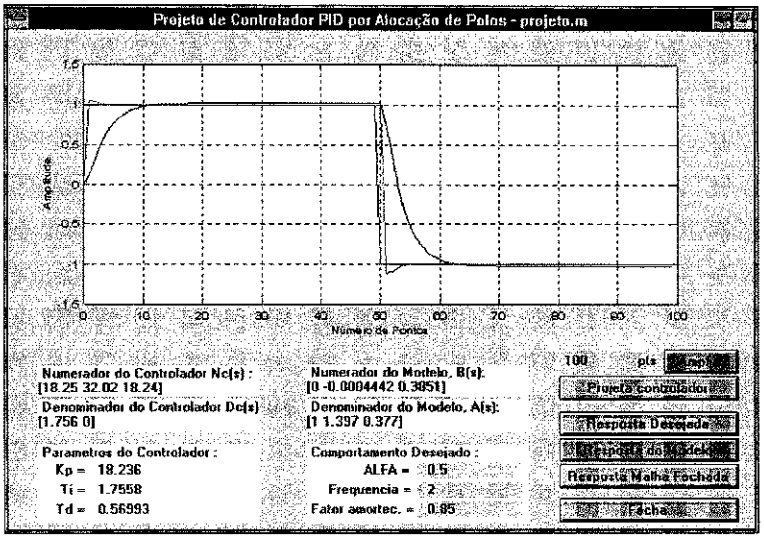

Figura 5.16: Janela para projeto de controladores.

está presente nesta tela. Da Figura pode-se também ver que os valores utilizados para definir as características desejadas para o sistema em malha fechada foram os seguintes, $\omega = 2,0$ para a frequência natural, $\zeta = 0,85$ para o fator de amortecimento, e $\alpha = 0,5$ para um pólo real. O numerador e denominador estimados da planta, no domínio contínuo, são dados respectivamente por,

$$-0,4442s \times 10^{-3} + 0,3851$$

e

$$s^2 + 1,397s + 0,377$$

Os ganhos proporcional, integral e derivativo do controlador PID obtido são,

$$\begin{aligned} K_p &= 18,236 \\ T_i &= 1,7558 \\ T_d &= 0,5699 \end{aligned}$$

O programa *projeto.m* possui as rotinas para o projeto de controladores implementadas neste trabalho.

5.8 Análise da Resposta em Frequência

O ISAC dispõe de rotinas para análise da resposta em frequência. O usuário pode obter os diagramas de Bode e Nyquist, a resposta ao impulso, os pólos e zeros do modelo. Nas Figuras 5.17 e 5.18 podem ser vistos os diagramas de Bode e Nyquist

de um modelo estimado. Os pólos e zeros do sistema podem ser vistos na janela de estimação do ISAC.

O usuário dispõe da resposta, no domínio do tempo, do modelo obtido. O mesmo sinal que foi utilizado para excitação da planta pode ser injetado no modelo estimado. O usuário pode ainda fornecer condições iniciais para a obtenção da resposta do modelo. Deste modo pode-se comparar as respostas real e estimada.

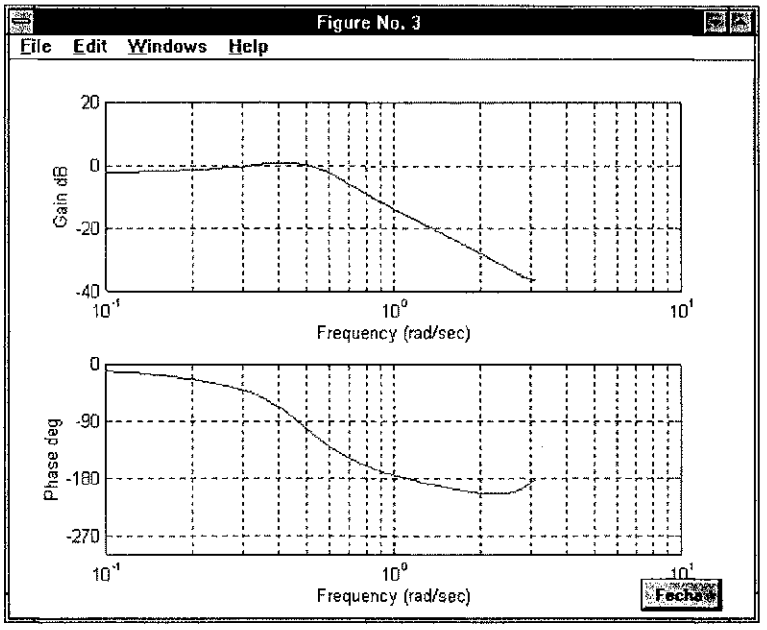

Figura 5.17: Diagrama de Bode de um modelo estimado.

5.9 Conclusão

Apresentou-se um pacote de rotinas que irá facilitar a execução de experimentos para modelagem de processos desconhecidos. A técnica de redução da ordem do modelo através da Decomposição em Valores Singulares permitem que o usuário reduza a ordem do modelo obtido. A facilidade de navegação e a flexibilidade para a realização de experimento com dados provenientes de arquivos, fazem com que o ISAC seja uma ferramenta de grande utilidade. Quaisquer processos com características dinâmicas representam uma aplicação em potencial para o pacote de rotinas ISAC. Optimizações podem ser implementadas no programa, mas o que há atualmente atende aos usuários iniciantes na área. Os métodos de identificação utilizados podem ainda ser estudados mais profundamente e novos métodos podem ser incluídos no pacote.

Outros programas para identificação de sistemas estão disponíveis no mercado, como por exemplo, o *toolbox* de identificação criado por Lennart Ljung [35] [18]. En-

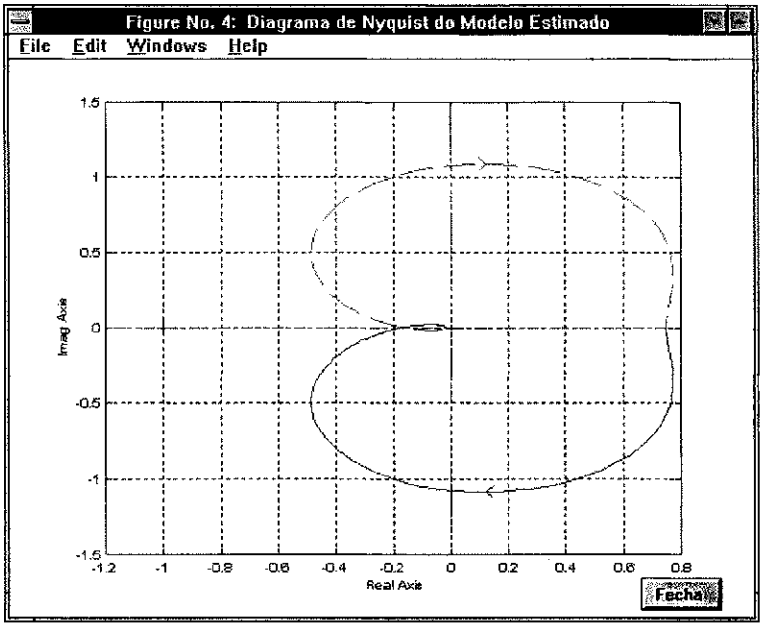

Figura 5.18: Janela para projeto de controladores.

tretanto, a proposta do ISAC que difere dos demais, é desenvolver um ambiente para a implementação de rotinas que utilizam *SVD* na redução da ordem do modelo, e ter a inclusão de rotinas para o projeto de controladores no mesmo pacote. Portanto, a redução de ordem por decomposição em valores singulares e o projeto de controladores são os pontos principais do programa.

O ISAC possui ferramentas de filtragem, gerenciamento dos dados em arquivos, análise de resposta em frequência e pode ser aplicado em quaisquer processos elétricos, mecânicos, químicos, ou outros. Quaisquer processos com características dinâmicas representam uma aplicação em potencial.

O ISAC reduz o esforço do usuário na utilização da linguagem do MATLAB, pois utiliza os recursos gráficos da linguagem. É composto de rotinas compactas que utilizam comandos internos e programas dos *toolboxes* básicos do MATLAB.

Capítulo 6

Simulações e Resultados Experimentais

Neste capítulo são apresentados os resultados de implementações e testes do método de identificação com atualização da matriz de covariância por *SVD*, idealizados por *Zhang* e da aplicação do *ISAC* em um protótipo de um secador de grãos. As rotinas utilizadas para teste dos algorítmos de identificação, implementadas no *MATLAB*, são apresentadas no *Apêndice B*. Foram utilizadas simulações de plantas para a comparação entre os algorítmos implementados.

6.1 Condições Gerais das Simulações

Para comparações entre os métodos apresentados nos capítulos anteriores, estes foram testados em cinco plantas distintas, cada uma com características particulares. Tais plantas e suas características de resposta serão apresentadas a seguir. A planta *P1* foi utilizada por Zhang em [15]; as demais plantas foram inicialmente utilizadas no trabalho de *Isermman* em [38], e posteriormente por Mascarenhas em [33].

Será denominada planta *P1* o processo cuja função de transferência no domínio discreto, é dada pela equação (6.1), utilizado também em [15],

$$G_1(q) = \frac{q^{-1} + 0,5q^{-2}}{1 - 1,5q^{-1} + 0,7q^{-2}} \tag{6.1}$$

a resposta ao degrau desta planta pode ser vista na Figura 6.1.

Os demais processos que foram utilizados nos experimentos são apresentados na Tabela 6.1, no domínio s.

Para a discretização dos sistemas foram utilizados valores distintos para as taxas de amostragem. As taxas de amostragem mais adequadas situam-se entre 1/10 a 1/50 da

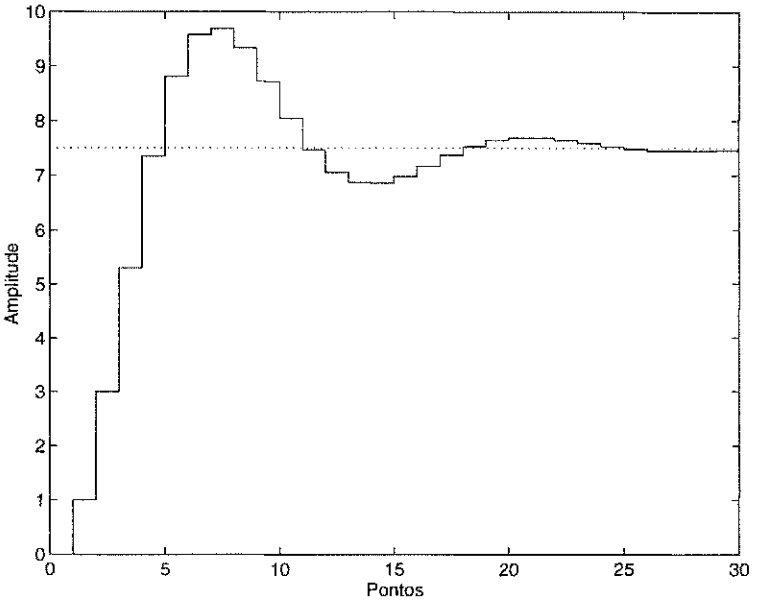

Figura 6.1: Resposta ao degrau unitário da planta P1.

Planta	Função de Transferência	Pólos	Zeros
P2	$G_2(s) = \frac{0,1811}{(s+0,31)(s+0,5843)}$	$s = -0,31$ e $s = -0,5843$	-
P3	$G_3(s) = \frac{3,845}{(s+1)(s+3,845)}$	$s = -1,0$ e $s = -3,845$	-
P4	$G_4(s) = \frac{16}{s^2+4s+16}$	$s = -2 \pm 3,4641i$	-
P5	$G_5(s) = \frac{1-s}{(4s+1)(s+1)}$	$s = -1,0$ e $s = -0,25$	1

Tabela 6.1: Funções de transferência das plantas utilizadas para testes.

	a1	a2	b1	b2	Amostr.(seg)
P2	$-1,901$	$0,903$	$0,0011$	$0,0011$	$T = 0,114$
P3	$-1,918$	$0,919$	$0,562 \times 10^{-3}$	$0,546 \times 10^{-3}$	$T = 0,017$
P4	$-1,859$	$0,875$	$0,0085$	$0,0081$	$T = 0,033$
P5	$-1,919$	$0,920$	$-0,0154$	$0,0165$	$T = 0,066$

Tabela 6.2: Valores dos parâmetros discretos.

menor constante de tempo do sistema [21]. Foi utilizado um tempo de amostragem de 1/15 da menor constante de tempo do sistema. Os valores dos parâmetros discretos, e tempo de amostragem utilizados para cada planta, são apresentados na Tabela 6.2.

As respostas ao degrau unitário das plantas utilizadas podem ser vistas na Figura 6.2. Pode-se observar que as plantas apresentam comportamento bem distintos. A planta *P2* tem a resposta mais lenta do grupo de processos submetidos aos testes com os algorítmos, a planta *P3* possui resposta rápida mas sem sobrepico, a planta *P4* possui resposta com sobrepicos (*overshoot*) e mostra-se como o sistema mais rápido, e a planta *P5*, um sistema de fase não mínima. Deste modo, tentou-se aplicar os testes em uma faixa de processos de dinâmicas bem distintas.

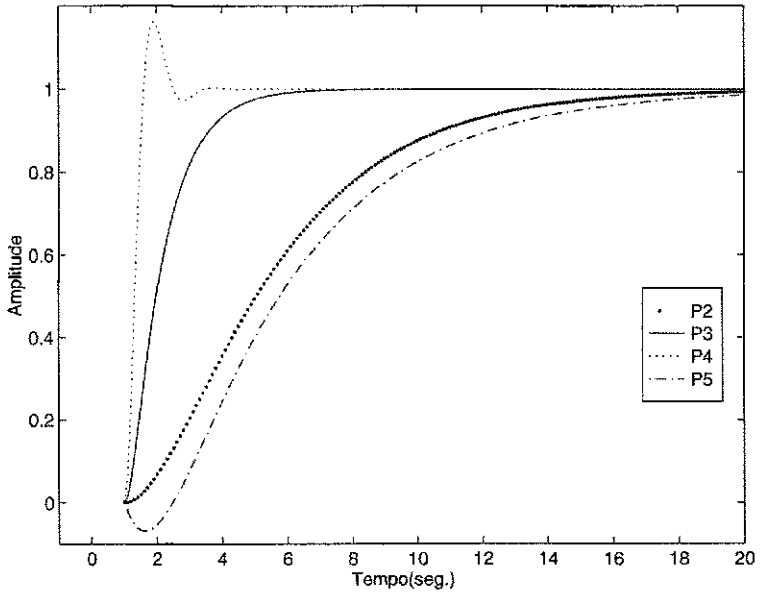

Figura 6.2: Resposta ao degrau unitário das plantas P2, P3, P4 e P5.

Os sinais de excitação utilizados nos experimentos de identificação apresentados neste capítulo foram do tipo, sinal de onda quadrada, seno e um sinal montado a partir de um ruído PRBS, que possuem persistência de ordem diferente, e podem ser vistos na Figura 6.3,

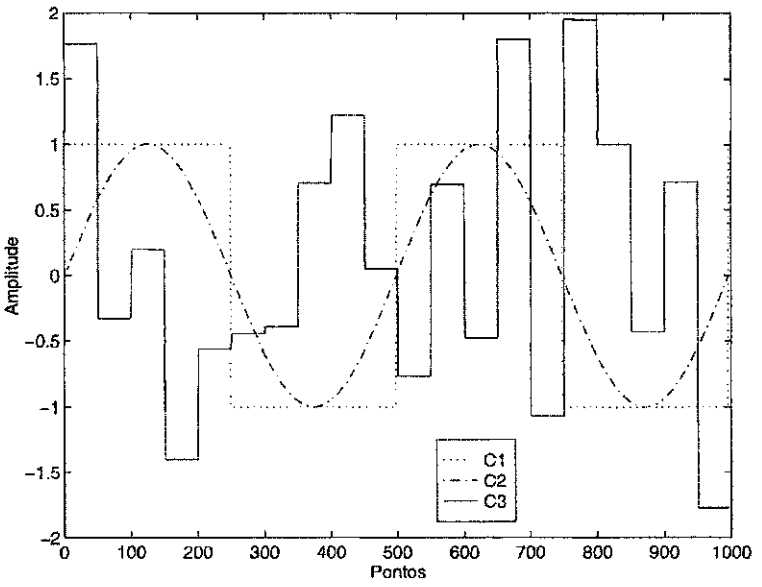

Figura 6.3: Sinais de excitação aplicados nas plantas.

Para simular um ruído de perturbação para o sistema, utilizou-se um modelo ARMAX. Foi gerado um ruído gaussiano, u_v, com média nula e variância 1, implementado com a função *randn* do MATLAB. Em seguida este ruido u_v foi filtrado com a seguinte função de transferência,

$$G_F(q) = \frac{0,3q^{-1} + 0,5q^{-2}}{1 - 0,5q^{-1} + 0,04q^{-2}} \tag{6.2}$$

gerando o sinal u_e, que foi filtrado mais uma vez pelo denominador da função de transferência da planta sob teste, gerando então o sinal y_e que foi somado à saída da planta. Assim, foi utilizado o modelo ARMAX para teste da rotina. Com esta implementação pode-se facilmente mudar para o modelo ARX simplesmente eliminando-se a etapa de filtragem G_F do sinal u_v, apresentada na equação (6.2).

O nível de ruído pode ser observado em termos da taxa de ruído por sinal N/S, calculada com a equação 6.3,

$$N/S = \frac{STD\,(\text{ruído})}{STD\,(\text{saída})} \tag{6.3}$$

na qual, o operador STD representa o desvio padrão do sinal, como foi adotado em [39]. Outra forma de se verificar o nível de ruído é através da relação sinal-ruído, S/N, que é calculada como,

$$S/N = \frac{\text{Potência do sinal}}{\text{Potência do ruído}} \tag{6.4}$$

porém neste trabalho foi adotada a relação N/S por ser diretamente proporcional ao nível de ruído, enquanto que a relação sinal-ruído é inversamente proporcional. Ambas são adequadas para mensurar a quantidade de ruído no sistema.

Uma forma de testar e comparar a convergência de dois ou mais algorítmos de estimação é utilizar os modelos, gerados por cada um dos algoritmos, em aplicações semelhantes. O projeto de controladores baseado em modelos paramétricos é um exemplo de aplicação que pode ser utilizada para este fim. Neste trabalho foi utilizada a técnica de projeto de controladores PID baseada em alocação de pólos para efeitos comparativos. É importante ressaltar que a técnica de projeto foi utilizada como um meio de se verificar a eficácia de um ou outro algoritmo, e sendo assim a técnica de projeto não é objeto principal de estudo neste trabalho. Para o projeto do controlador PID pelo método de alocação de pólos foi utilizada a seguinte característica desejada para a malha fechada,

$$G_{desejado}(s) = \frac{\alpha\omega^3}{(s+\alpha\omega)(s^2+2\omega\zeta s+\omega^2)} \tag{6.5}$$

os valores utilizados para ω, ζ, e α serão apresentados nas seções subsequentes de acordo com o caso.

Outra forma de avaliar o modelo estimado é através de uma função de custo, $V_N\left(\boldsymbol{\theta}, \mathbf{Z}^N\right)$, apresentada no capítulo 2.

6.2 Experimentos com o Método de Zhang

O programa *ZHANGP.M* foi implementado exclusivamente para avaliações do método de Zhang e comparação com outros métodos. Como foi visto no capítulo 3.2, o método proposto por Zhang foi criado para atender principalmente aos casos de identificação com sinal de excitação de baixa persistência e na presença de muito ruído, situação em que o método dos mínimos quadrados recursivo não se mostrou tão estável e preciso, como pode ser visto nos gráficos subsequentes.

Para plantas de segunda ordem e utilizando sinais de excitação com persistência de ordem 2, como o sinal senoidal, e na presença de ruído aditivo com $N/S = 0,8514$, notou-se que a convergência paramétrica foi mais rápida e mais precisa no caso da identificação pelo método com atualização da matriz de covariância por SVD (Zhang), do que no método dos mínimos quadrados recursivo (RLS), como pode ser visto nas Figuras 6.4 e 6.5, as curvas representam o valor real do parâmetro para a planta *P3* , o parâmetro estimado por RLS e o parâmetro estimado pelo método de Zhang, como está indicado na legenda. Na Figura 6.4, percebe-se que os parâmetros estimados pelo método de Zhang confundem-se com os valores reais. Na Figura 6.6, o sinal utilizado na excitação, **excit**, as curvas da saída real da planta com ruído, **real**, ruído aplicado, **ruido**, e da saída dos modelos estimados pelo método dos mínimos quadrados recursivo,

rls, e método de Zhang, **zhang**. Percebe-se ainda na Figura 6.6, e nas demais figuras de resposta temporal que serão apresentadas adiante, que os sinais de saída dos modelos estimados e a saída real estão sobrepostos. Na Figura 6.7, é apresentado o erro de predição para as estimações paramétricas da planta. Na Figura 6.8, são apresentadas a resposta da planta P3 em uma configuração de controle com um controlador do tipo PID projetado pelo método de alocação de pólos e baseando-se no modelos estimados. Foram calculados dois controladores PID com base nos modelos de cada método, para comparar a qualidade dos modelos estimados. São apresentados a saída da planta em malha aberta, **abert**, o sinal de referência para o sistema em malha fechada, **refer**, a saída do sistema com projeto do controlador baseado no modelo estimado por RLS, **rls**, e a saída do sistema com o projeto baseado no modelo estimado pelo que utiliza SVD, **zhang**.

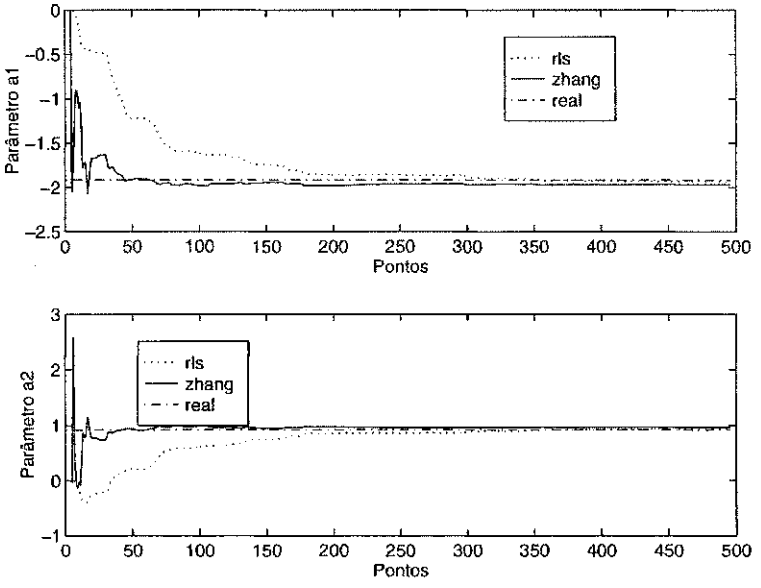

Figura 6.4: Parâmetros a1 e a2 do modelo estimado para a planta P3.

Para o projeto dos controladores por alocação de pólos foram utilizados os seguintes parâmetros,

$$\omega = 2,0, \qquad \zeta = 0,707 \qquad \text{e} \qquad \alpha = 1,0.$$

Pode-se notar que, com o método de Zhang, tais parâmetros sofrem menos influência devido ao ruído. Daí conclui-se que neste caso o método de Zhang é mais robusto e o algoritmo mais estável que o método RLS. Entretanto, o resultado da aplicação dos modelos no projeto do controlador PID, foi idêntica para ambos os casos. O fator de esquecimento utilizado para o método RLS foi de $0,995$ e para o método de Zhang, 1×10^{-8}.

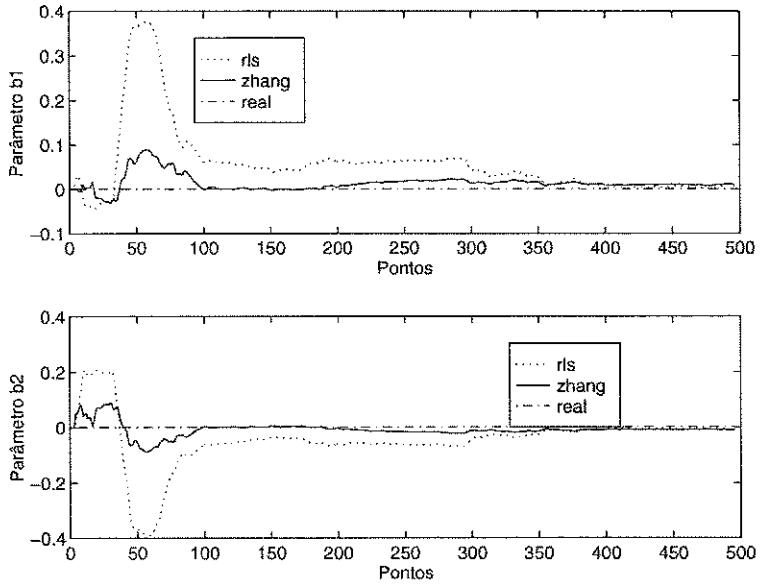

Figura 6.5: Parâmetros b1 e b2 do modelo estimado para a planta P3.

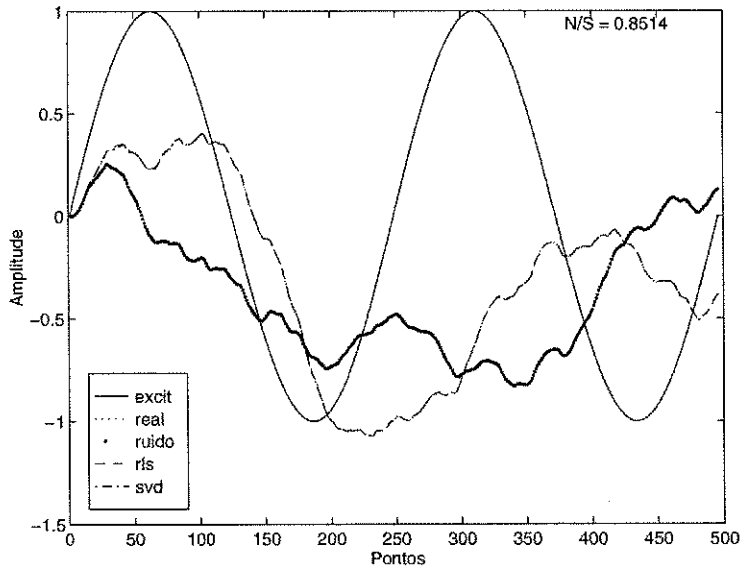

Figura 6.6: Resposta no tempo do modelo estimado para a planta P3.

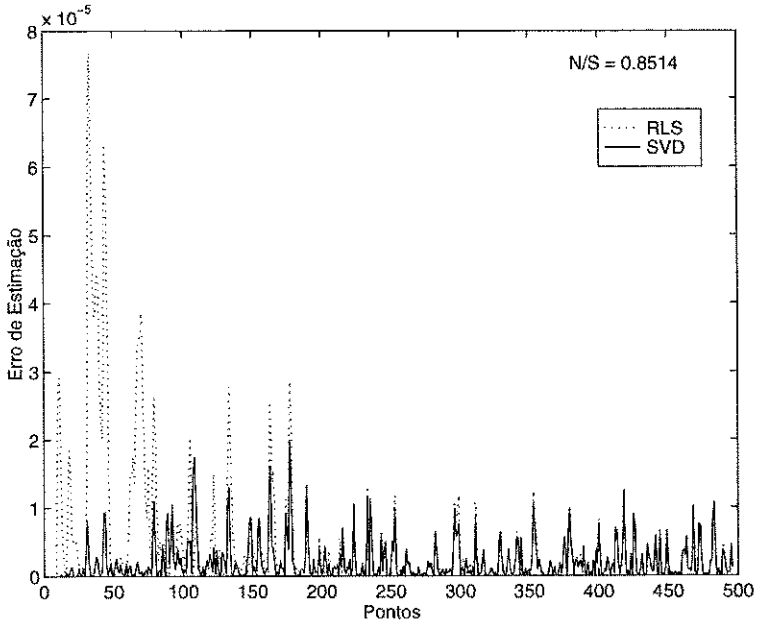

Figura 6.7: Erro de predição para as estimações da planta P3.

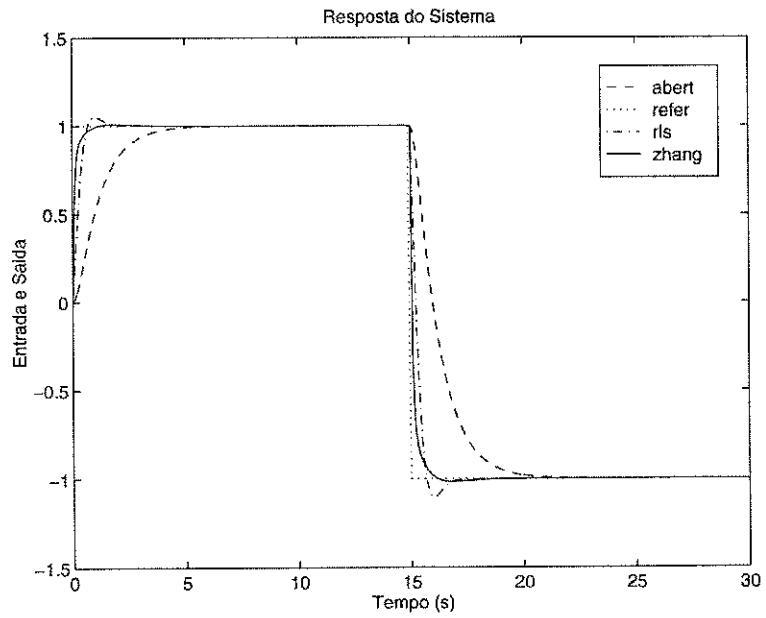

Figura 6.8: Resposta do sistema P3 em malha fechada com PID.

Para este e para os demais experimentos que serão apresentados adiante, foram selecionadas as melhores de várias simulações, cerca de dez para cada planta, modificando-se apenas o fator de esquecimento no algorítmo dos mínimos quadrados (λ_{RLS}) e o fator de esquecimento no algorítmo do método de Zhang (λ_{Zhang}). Foram utilizados diversos valores para o fator de esquecimento, o critério para a modificação destes valores e os valores incrementados ou decrementados, foram baseados nos resultados obtidos nas estimações. Deste modo foram obtidos vários gráficos com valores distintos do fator de esquecimento. Entretanto, não serão apresentados os respectivos gráficos por motivos volumétricos. Este foi o procedimento adotado para encontrar o fator de esquecimento ideal para cada planta.

No método dos mínimos quadrados quando diminue-se o fator de esquecimento aumenta-se a velocidade de convergência dos parâmetros, porém aumentam-se também as variações paramétricas. No método de Zhang quando diminue-se o fator de esquecimento aumenta-se a velocidade de convergência dos parâmetros,entretanto as variações paramétricas não se modificam tanto quanto no RLS.

Para constatar os fatos comentados no parágrafo anterior, foi feita uma nova simulação com a planta *P3*, utilizando-se $\lambda_{RLS} = 0,980$ e com o mesmo valor de λ_{Zhang}, e para um nível de ruído $N/S = 0,6714$. Pode se constatar que, no método RLS, os parâmetros a1 e a2 convergiram mais rapidamente que no experimento anterior, Figura 6.9, porém as variações dos parâmetros b1 e b2 aumentaram, como pode ser visto na Figura 6.10. Pelo método de Zhang os parâmetros convergiram mais rapidamente e com variação paramétrica menor. Os sinais de saída e ruído são apresentados na Figura 6.11 e na Figura 6.12, verifica-se o sistema em malha fechada com um controlador PID, em que o resultado mostrou-se melhor com a utilização do método de Zhang. Na Figura 6.13, é apresentado o erro de predição para as estimações paramétricas da planta.

Para verificar o efeito da persistência da excitação sobre os algorítmos, um novo experimento foi realizado, utilizando-se um sinal de excitação mais persistente que o senoidal. Assim, utilizando uma onda quadrada, para a mesmo experimento com a planta *P3*, e com $\lambda_{RLS} = 0,985$ e $\lambda_{Zhang} = 1 \times 10^{-9}$ e ruído $N/S = 0,6238$, verificou-se que o método de Zhang ainda foi superior ao RLS. Os parâmetros são apresentados nas Figuras 6.14 e 6.15. Os parâmetros mostram-se tendenciosos, e isto ocorre devido ao sinal de excitação que não teve persistência suficiente. Os sinais de saída e ruído são apresentados na Figura 6.16; Na Figura 6.18, é apresentado o erro de predição para as estimações paramétricas da planta. E na Figura 6.17, o sistema em malha fechada, com resultados melhores para Zhang.

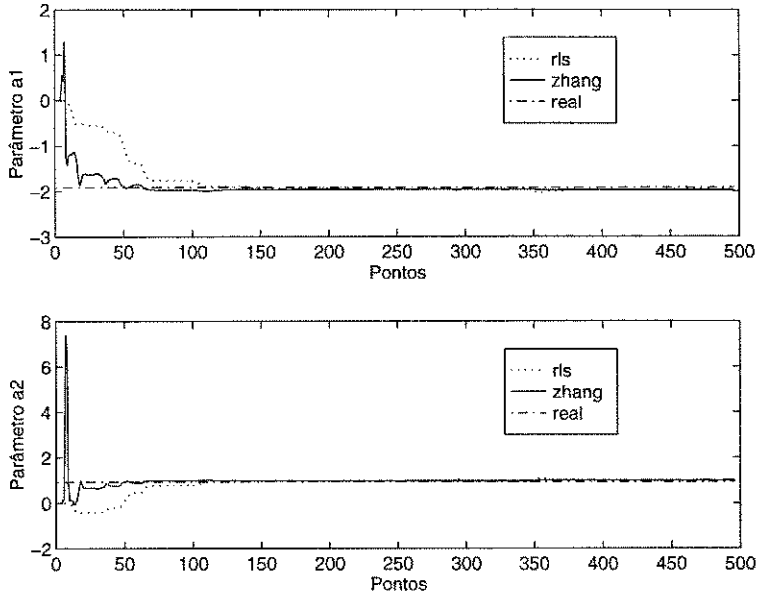

Figura 6.9: Parâmetros a1 e a2 do modelo estimado para a planta P3.

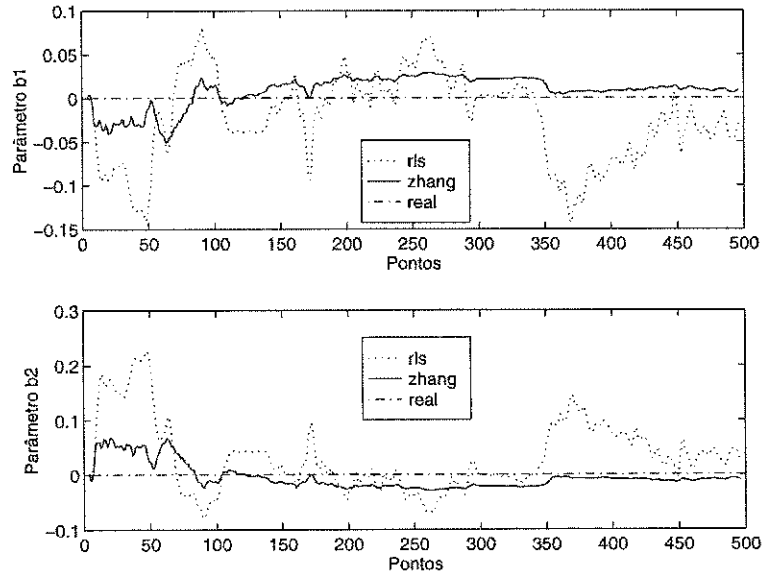

Figura 6.10: Parâmetros b1 e b2 do modelo estimado para a planta P3.

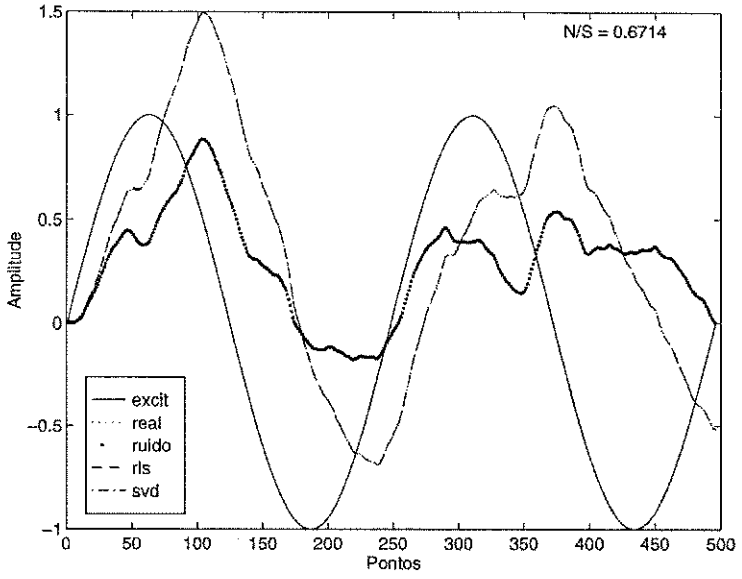

Figura 6.11: Resposta no tempo do modelo estimado para a planta P3.

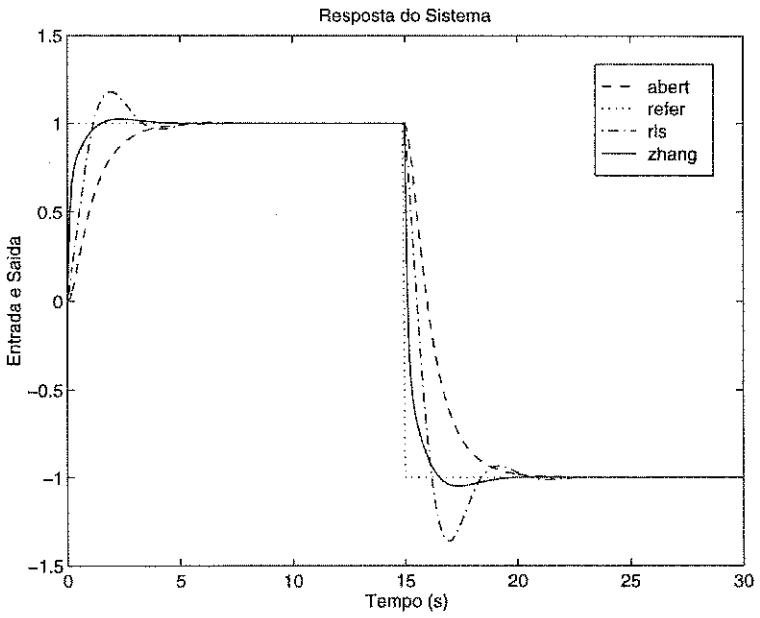

Figura 6.12: Resposta do sistema P3 em malha fechada com PID.

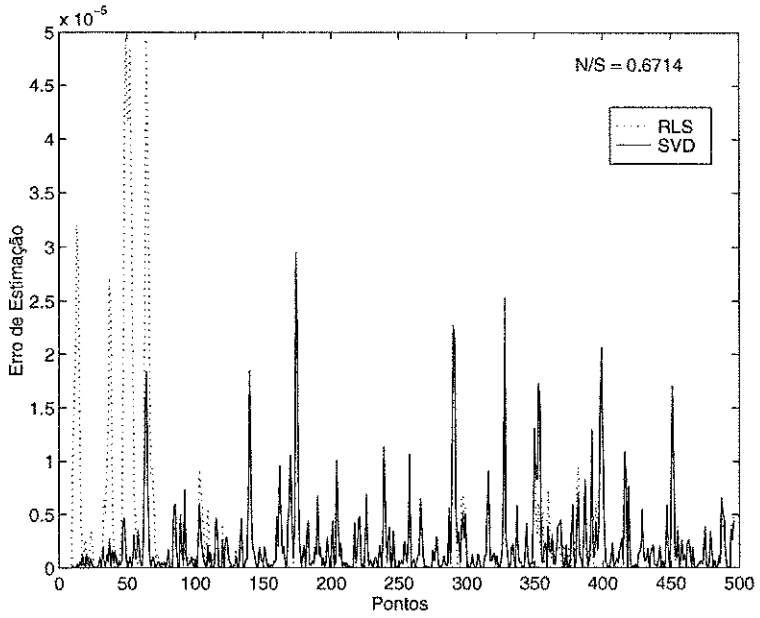

Figura 6.13: Erro de predição para as estimações da planta P3.

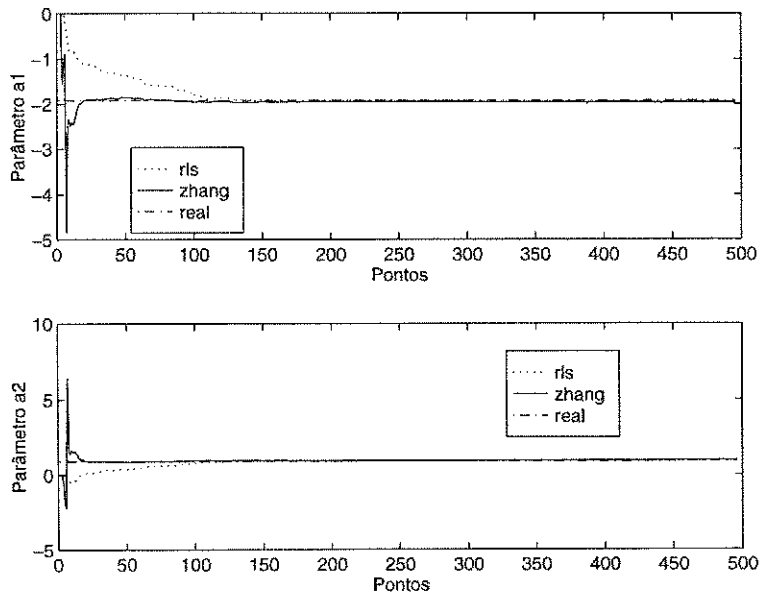

Figura 6.14: Parâmetros a1 e a2 do modelo estimado para a planta P3.

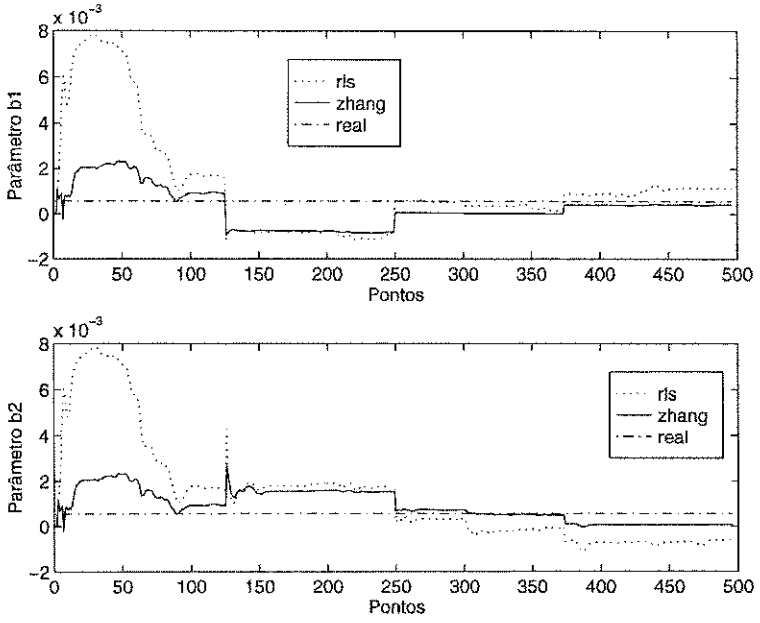

Figura 6.15: Parâmetros b1 e b2 do modelo estimado para a planta P3.

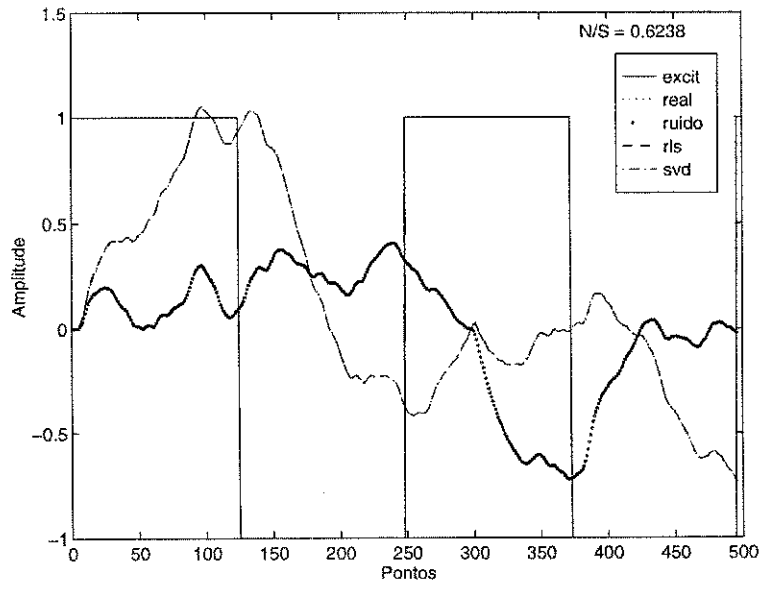

Figura 6.16: Resposta no tempo do modelo estimado para a planta P3.

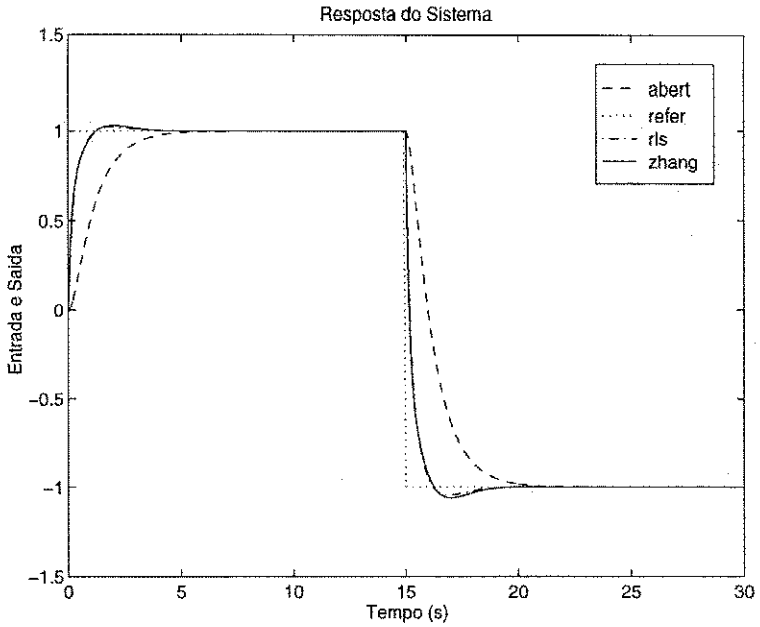

Figura 6.17: Resposta do sistema P3 em malha fechada com PID.

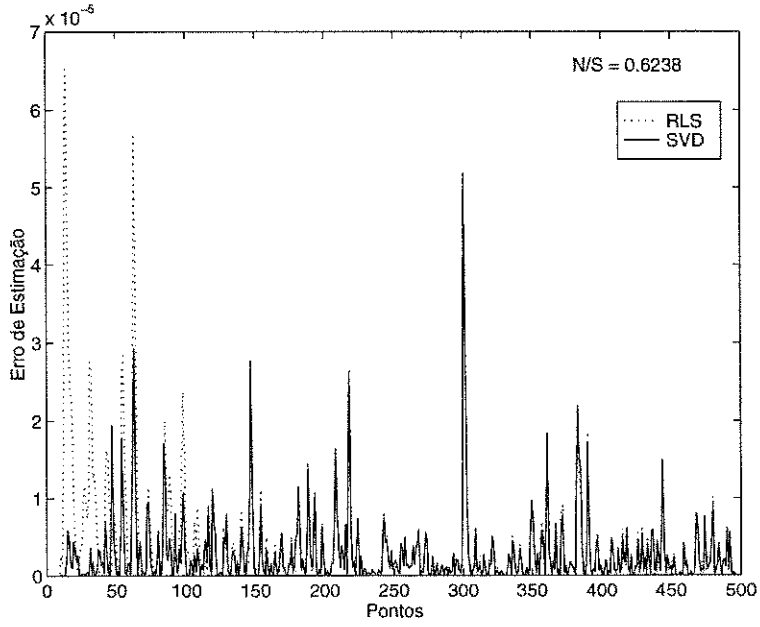

Figura 6.18: Erro de predição para as estimações da planta P3.

Pode-se utilizar o método das variáveis instrumentais para tentar reduzir a polarização dos parâmetros estimados. Assim, com um nível de ruído semelhante, $N/S = 0,6739$, foi feita uma nova simulação. Os parâmetros resultantes são apresentados nas Figuras 6.19 e 6.20. Os parâmetros continuam apresentando polarização, isto indica que o método das variáveis instrumentais, para estas condições, não consegue resolver o problema da polarização. O método baseado em SVD mantem-se como a melhor opção. Os sinais de saída e ruído são apresentados na Figura 6.21. Na Figura 6.22, é apresentado o erro de predição para as estimações paramétricas da planta.

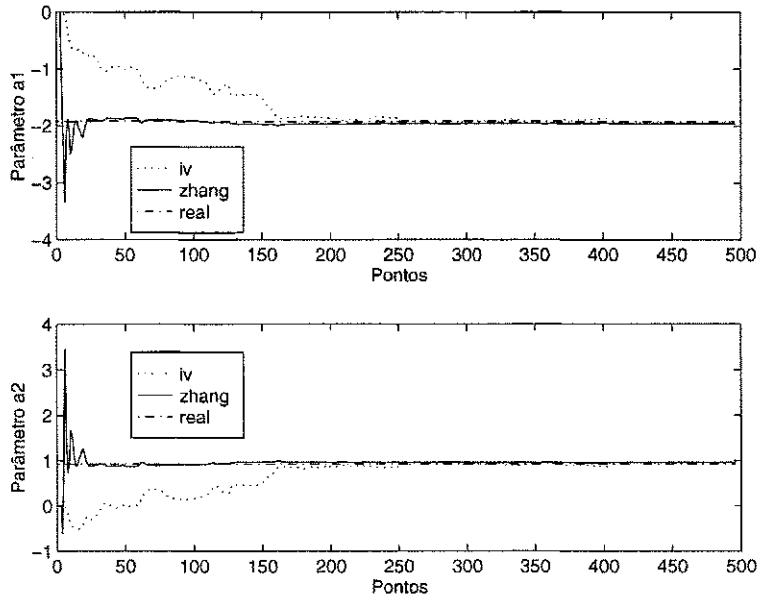

Figura 6.19: Parâmetros a1 e a2 do modelo estimado para a planta P3.

Para a planta *P1*, utilizando-se $\lambda_{RLS} = 0,99$ e $\lambda_{Zhang} = 0,5$, com um sinal de excitação senoidal, e com um nível de ruído $N/S = 0,06437$, podem ser visto nas Figuras 6.23 e 6.24 os gráficos dos parâmetros dos modelos estimados. Percebe-se que para o método Zhang a convergência dos parâmetros é maior e as variações paramétricas são menores. O método dos mínimos quadrados sofre uma grande influência do ruído de perturbação. Os sinais e ruído podem ser visto na Figura 6.25. Na Figura 6.26, é apresentado o erro de predição para as estimações paramétricas da planta.

Como a variação paramétrica para o RLS apresentou-se considerável no experimento anterior, aumentou-se o fator de esquecimento λ_{RLS}. Então, para um $\lambda_{RLS} = 1$, e com um nível de ruído $N/S = 0,07044$, o resultado pode ser observado nas Figuras 6.27, 6.28 e 6.29. A estimação pelo método de Zhang apesar de não ter tido uma boa convergência, ainda mostrou-se melhor. Na Figura 6.30, é apresentado o erro de predição para as estimações paramétricas da planta.

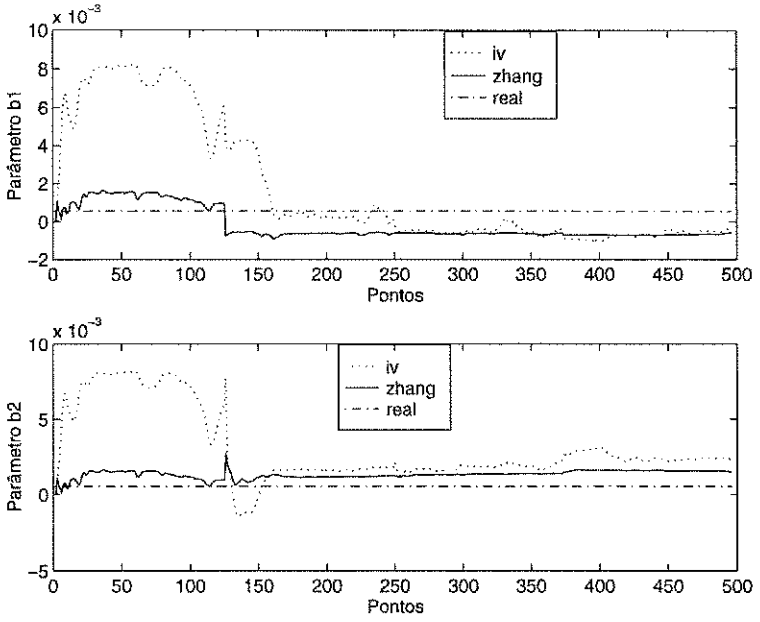

Figura 6.20: Parâmetros b1 e b2 do modelo estimado para a planta P3.

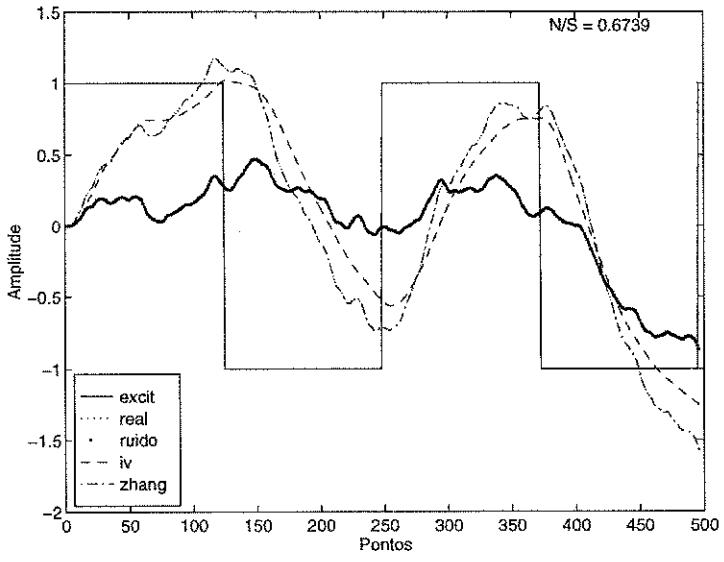

Figura 6.21: Resposta no tempo do modelo estimado para a planta P3.

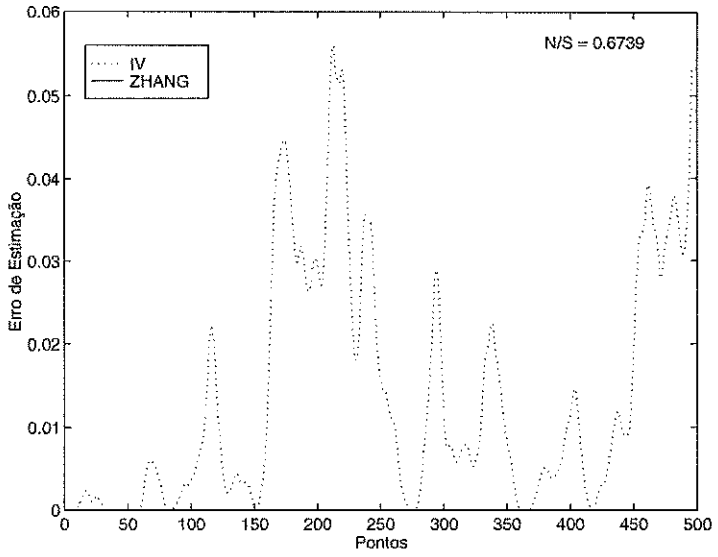

Figura 6.22: Erro de predição para as estimações da planta P3.

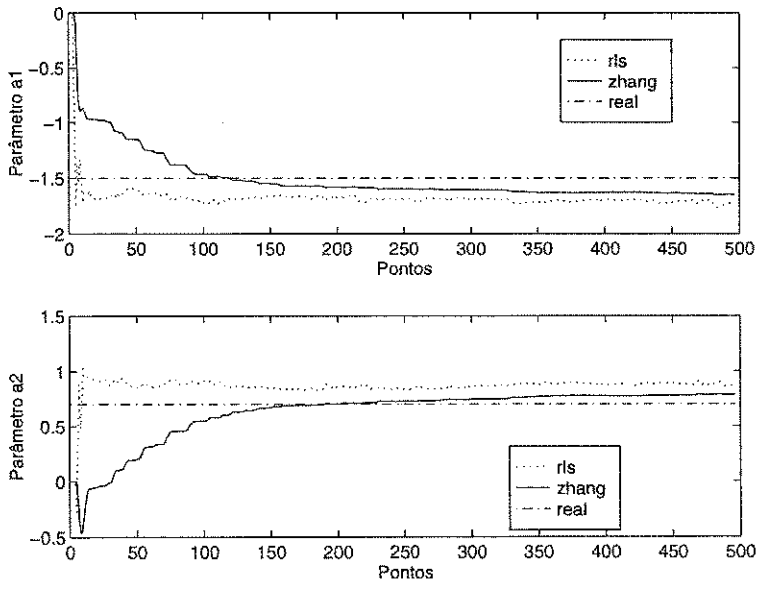

Figura 6.23: Parâmetros a1 e a2 do modelo estimado para a planta P1.

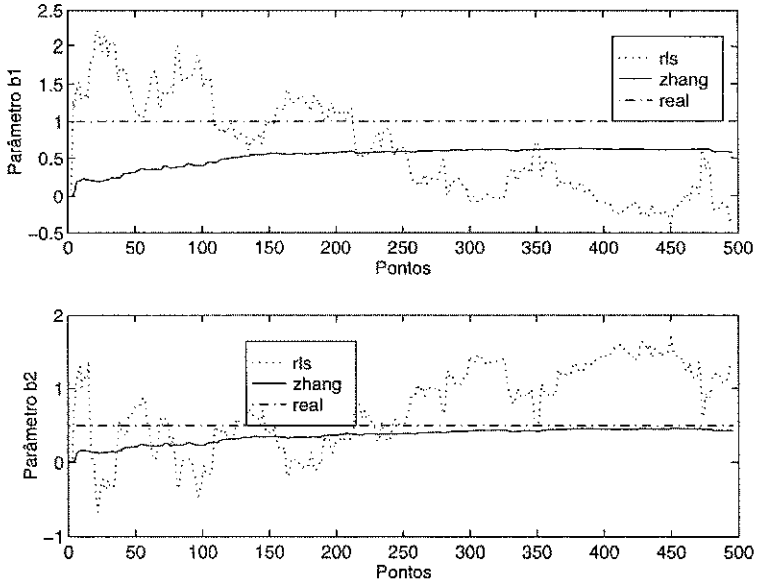

Figura 6.24: Parâmetros b1 e b2 do modelo estimado para a planta P1.

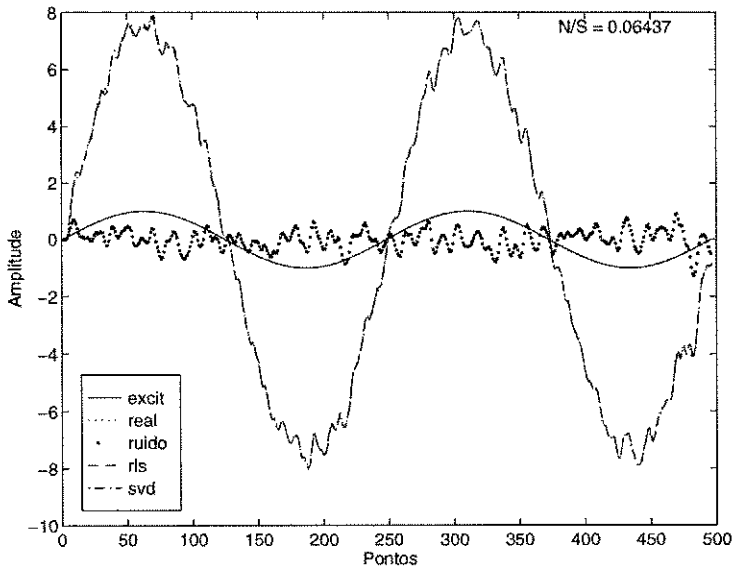

Figura 6.25: Resposta no tempo do modelo estimado para a planta P1.

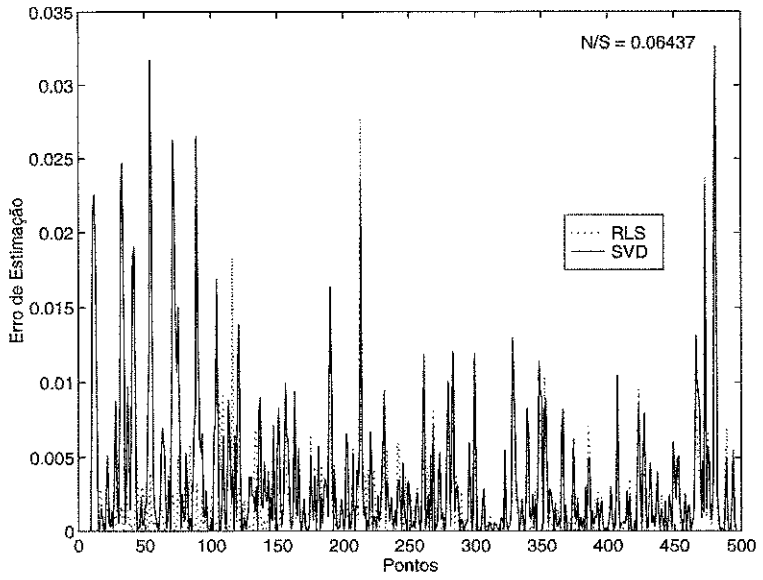

Figura 6.26: Erro de predição para as estimações da planta P1.

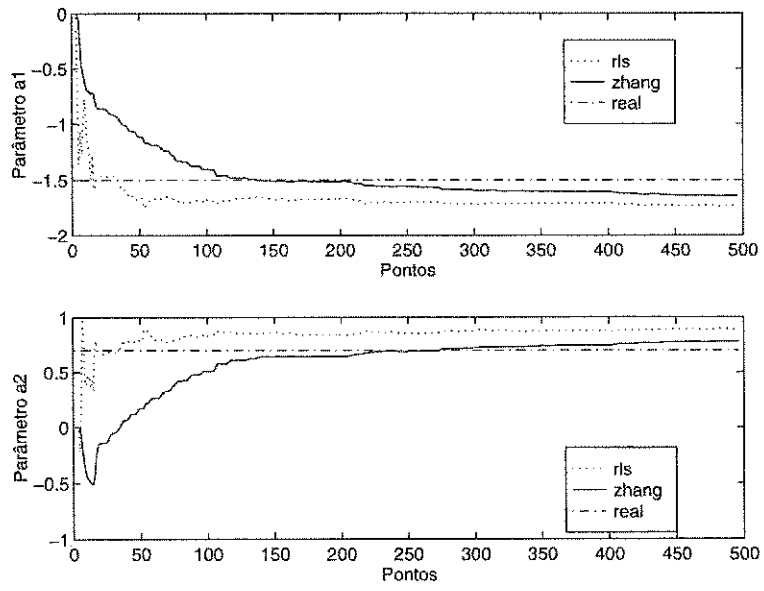

Figura 6.27: Parâmetros a1 e a2 do modelo estimado para a planta P1.

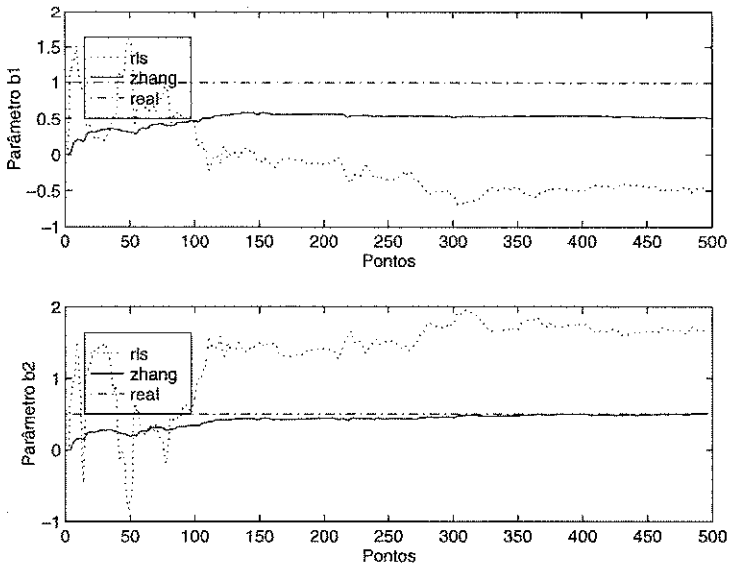

Figura 6.28: Parâmetros b1 e b2 do modelo estimado para a planta P1.

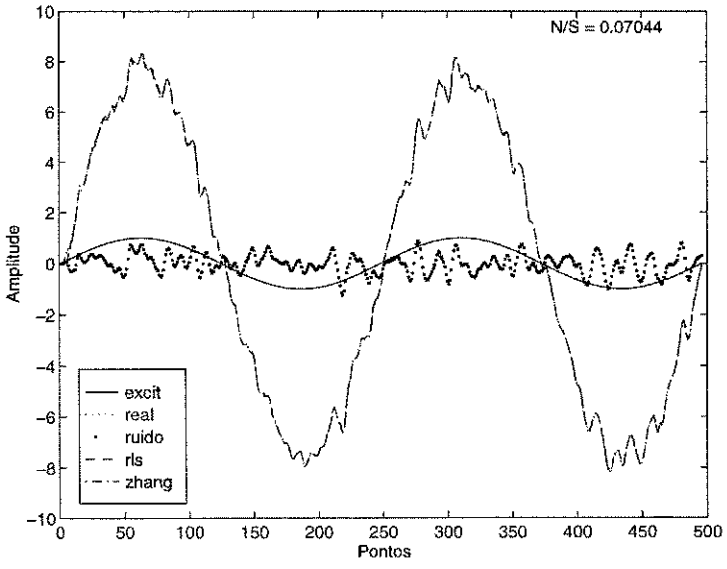

Figura 6.29: Resposta no tempo do modelo estimado para a planta P1.

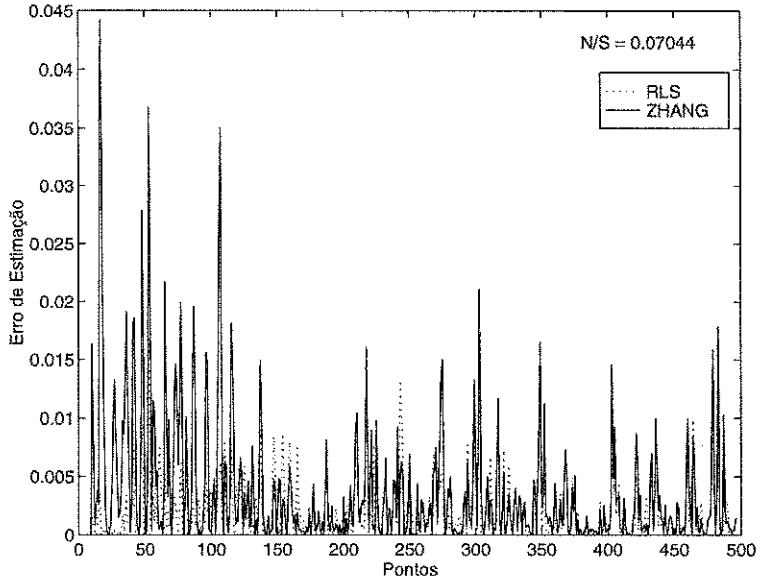

Figura 6.30: Erro de predição para as estimações da planta P1.

Utilizando a planta *P2*, e com $N/S = 0,4814$, os parâmetros estimados pelos dois métodos podem ser observados na Figura 6.31 e 6.32 nos quais foram utilizados fatores de esquecimento $\lambda_{RLS} = 0,99$ e $\lambda_{Zhang} = 1 \times 10^{-8}$, e sinal senoidal como excitação, apresentado na Figura 6.33. Na Figura 6.34, é apresentado o erro de predição para as estimações paramétricas da planta. Os resultados se mostraram melhores para o método de Zhang, como pode ser observado. A convergência paramétrica também foi mais rápida para Zhang.

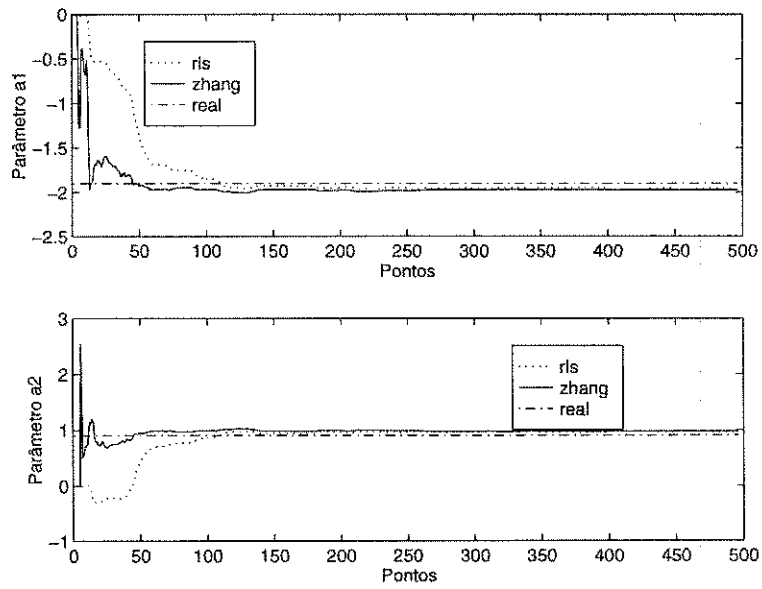

Figura 6.31: Parâmetros a1 e a2 do modelo estimado para a planta P2.

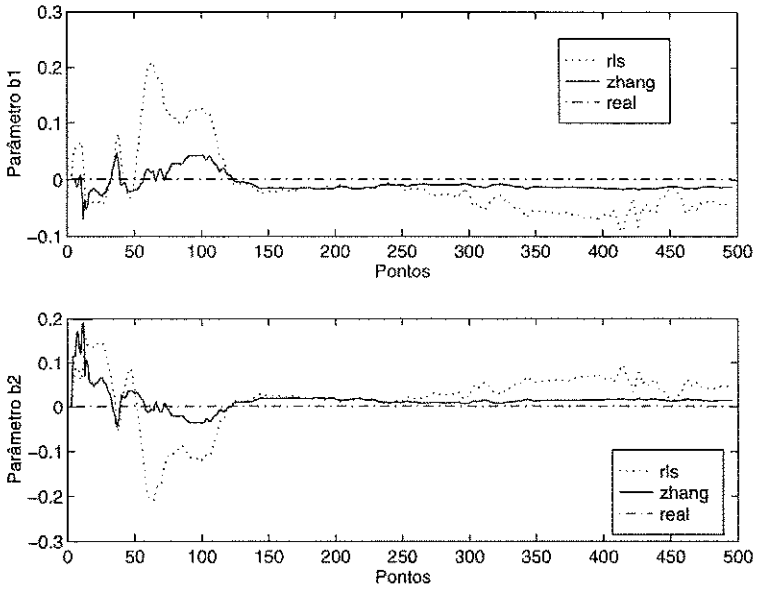

Figura 6.32: Parâmetros b1 e b2 do modelo estimado para a planta P2.

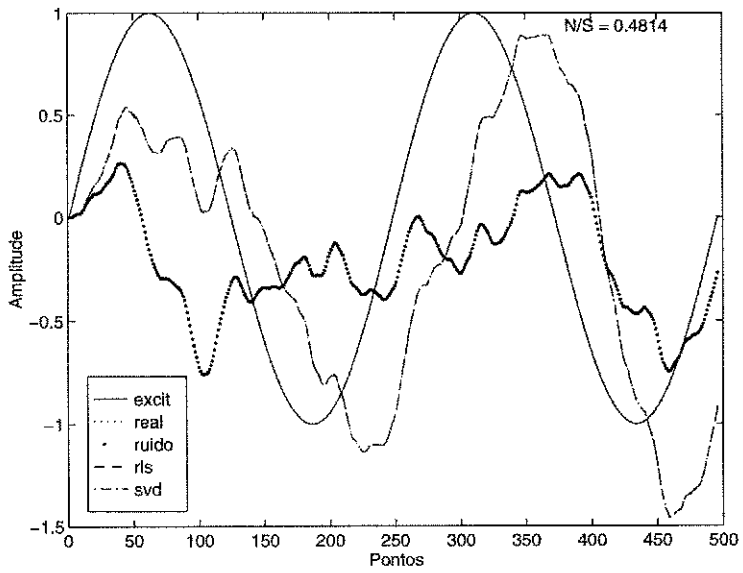

Figura 6.33: Resposta no tempo do modelo estimado para a planta P2.

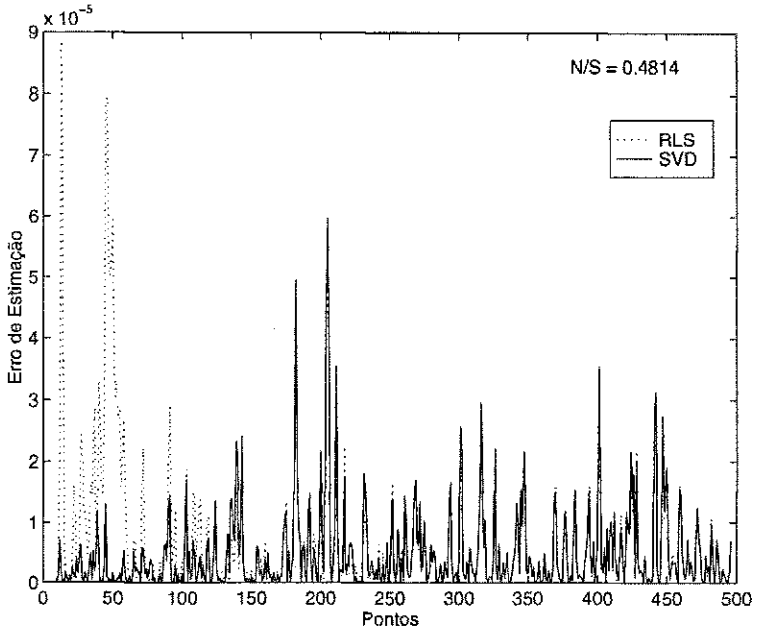

Figura 6.34: Erro de predição para as estimações da planta P1.

Repetido o experimento para a planta *P2*, com fatores de esquecimento $\lambda_{RLS} = 0,99$ e $\lambda_{Zhang} = 1 \times 10^{-8}$, e com nível de ruído $N/S = 0,06493$, o resultado ainda se mostrou melhor para o método de Zhang, pois a convergência é mais rápida e a estabilidade é maior. Figuras 6.35, 6.36 e 6.37. Na Figura 6.38, é apresentado o erro de predição para as estimações paramétricas da planta.

Nas Figuras 6.39 e 6.40, podem ser observados os parâmetros estimados para a planta *P4*, nos quais o método de Zhang também se mostra melhor que o RLS. A convergência para o método RLS não foi tão satisfatória, e a estabilidade também é melhor para o método de Zhang. Os sinais de excitação, ruído e saída são apresentados na Figura 6.41. Na Figura 6.42, é apresentado o erro de predição para as estimações paramétricas da planta. Foram utilizados os fatores de esquecimento, $\lambda_{RLS} = 0,99$ e $\lambda_{Zhang} = 1 \times 10^{-8}$, e o nível de ruído foi de $N/S = 0,0282$.

Para o caso em que utilizou-se a planta *P4*, $\lambda_{RLS} = 0,97$ e $\lambda_{Zhang} = 1 \times 10^{-6}$ e com $N/S = 1,455$. Os resultados para os parâmetros a1 e a2 foram semelhantes, porém para b1 e b2, o método de Zhang se mostrou mais próximo preciso e estável, como pode ser observado nas Figuras 6.43, 6.44 e 6.45. Na Figura 6.46, é apresentado o erro de predição para as estimações paramétricas da planta.

Para um experimento com a planta *P5*, utilizando-se $\lambda_{RLS} = 0,99$ e $\lambda_{Zhang} = 1 \times 10^{-6}$ e com $N/S = 0,7287$, Os gráficos mostram que para o método de Zhang os resultados foram melhores com a convergência mais rápida e fiel, apresentados nas Figuras 6.47, 6.48 e 6.49. Na Figura 6.50, é apresentado o erro de predição para as

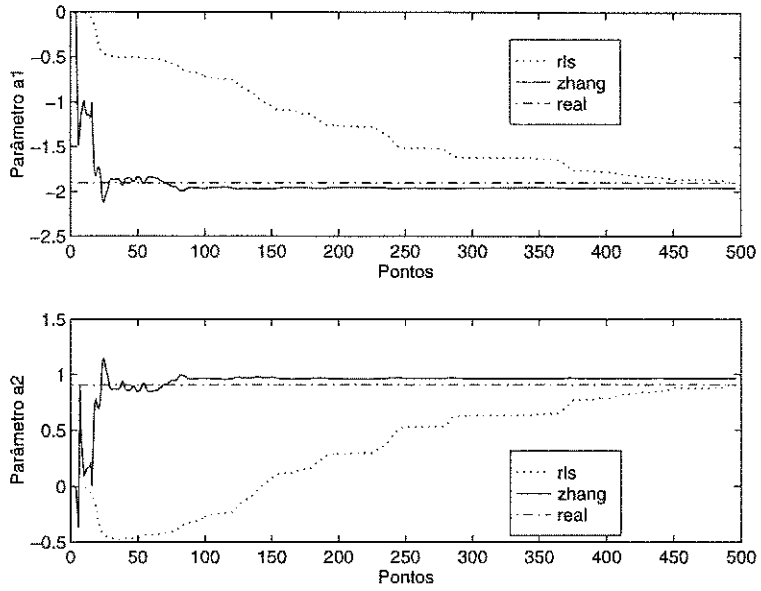

Figura 6.35: Parâmetros a1 e a2 do modelo estimado para a planta P2.

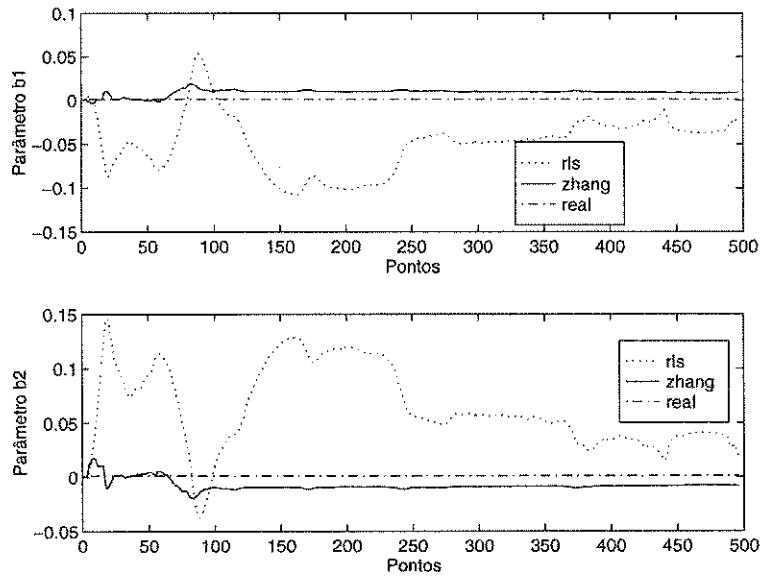

Figura 6.36: Parâmetros b1 e b2 do modelo estimado para a planta P2.

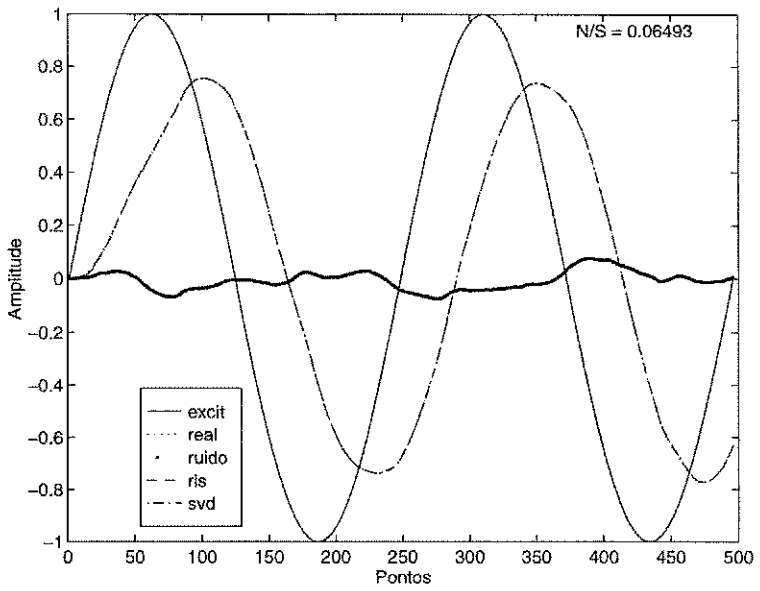

Figura 6.37: Resposta no tempo do modelo estimado para a planta P2.

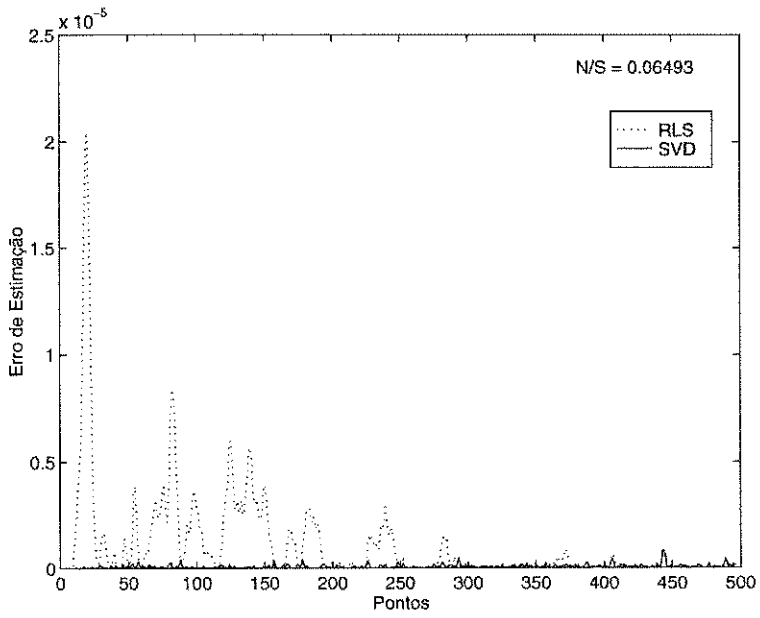

Figura 6.38: Erro de predição para as estimações da planta P1.

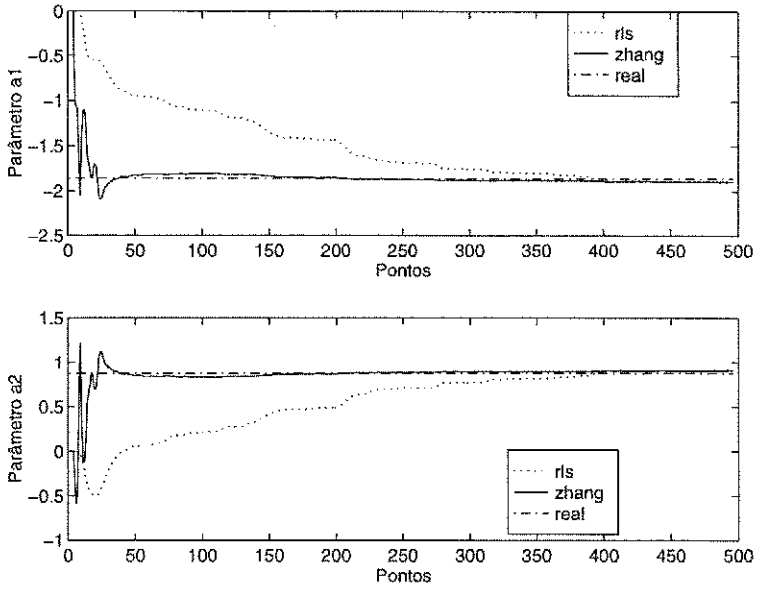

Figura 6.39: Parâmetros a1 e a2 do modelo estimado para a planta P4.

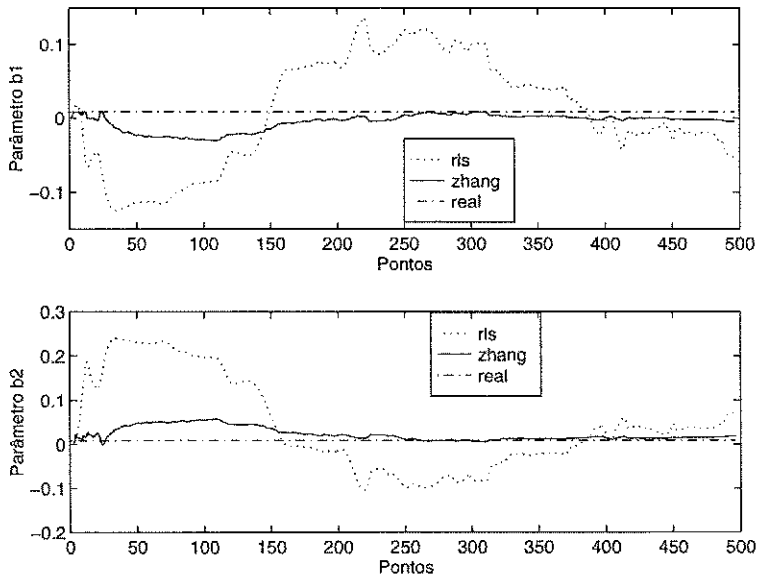

Figura 6.40: Parâmetros b1 e b2 do modelo estimado para a planta P4.

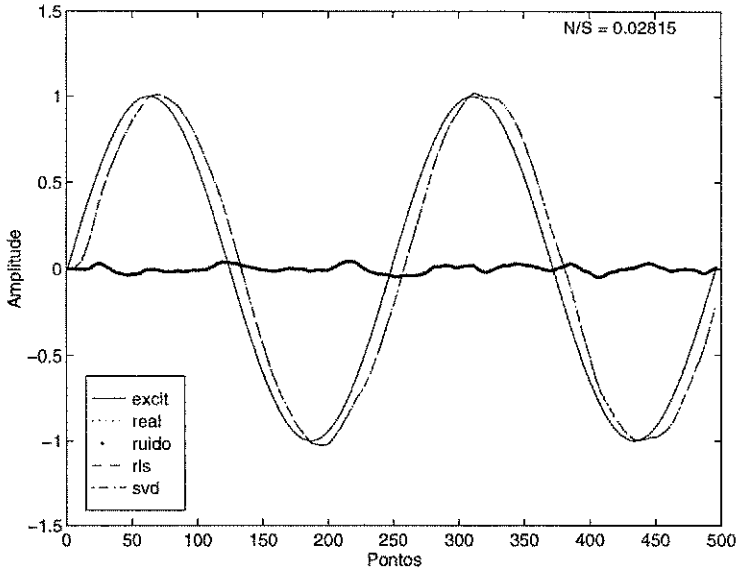

Figura 6.41: Resposta no tempo do modelo estimado para a planta P4.

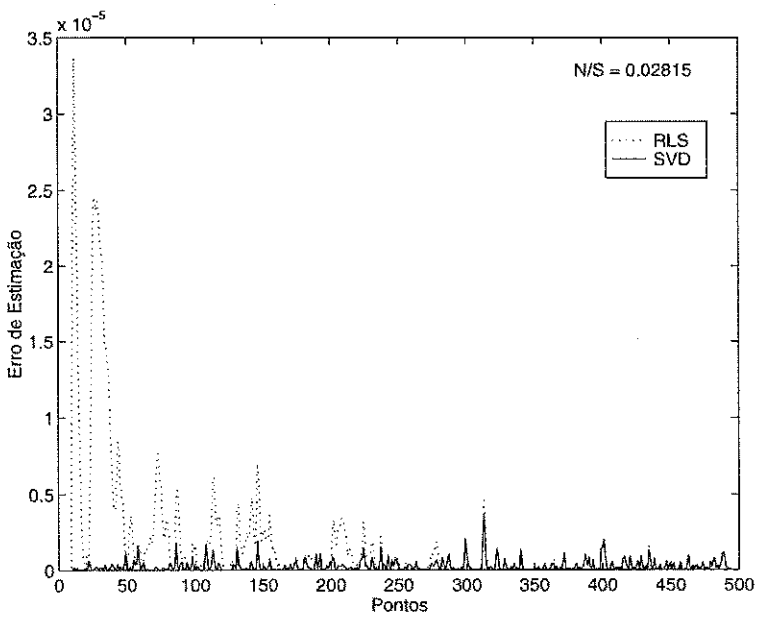

Figura 6.42: Erro de predição para as estimações da planta P1.

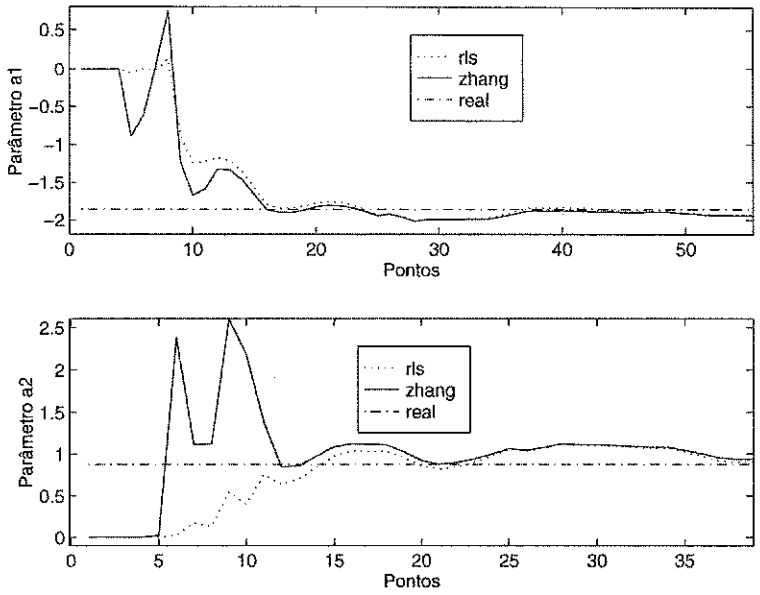

Figura 6.43: Parâmetros a1 e a2 do modelo estimado para a planta P4.

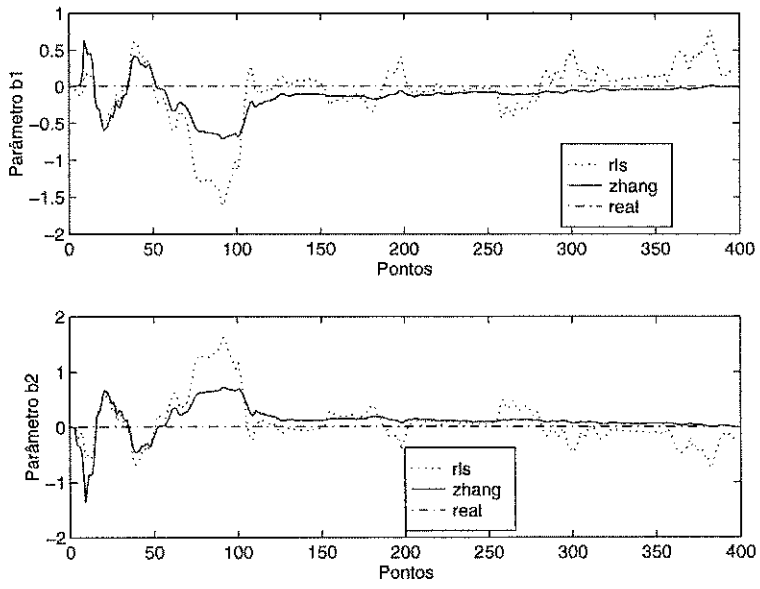

Figura 6.44: Parâmetros b1 e b2 do modelo estimado para a planta P4.

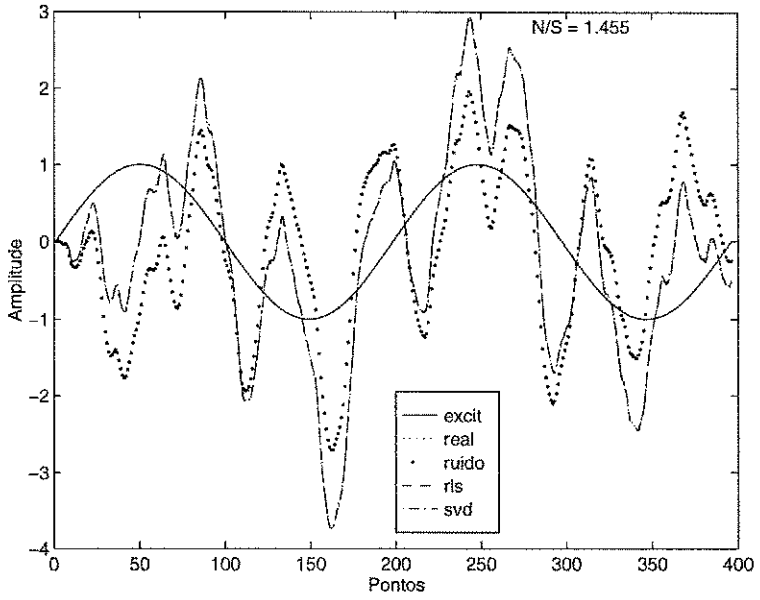

Figura 6.45: Resposta no tempo do modelo estimado para a planta P4.

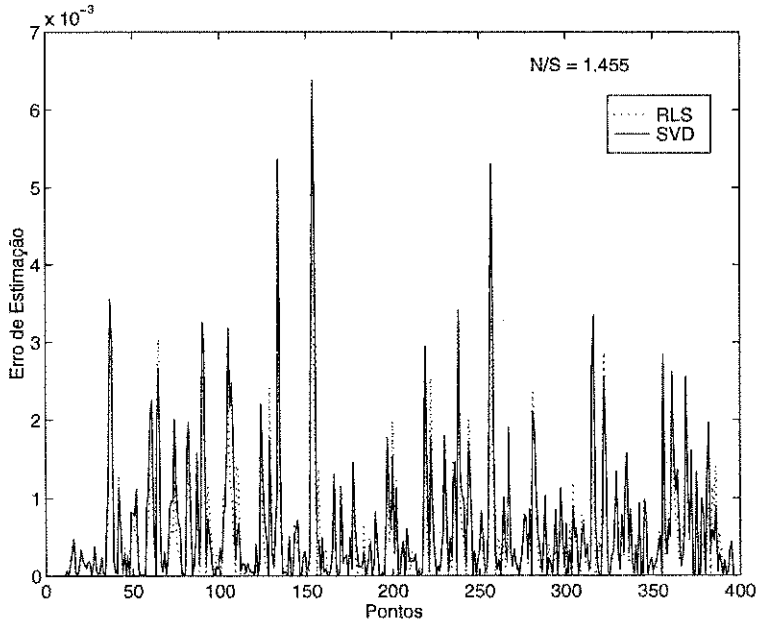

Figura 6.46: Erro de predição para as estimações da planta P1.

estimações paramétricas da planta.

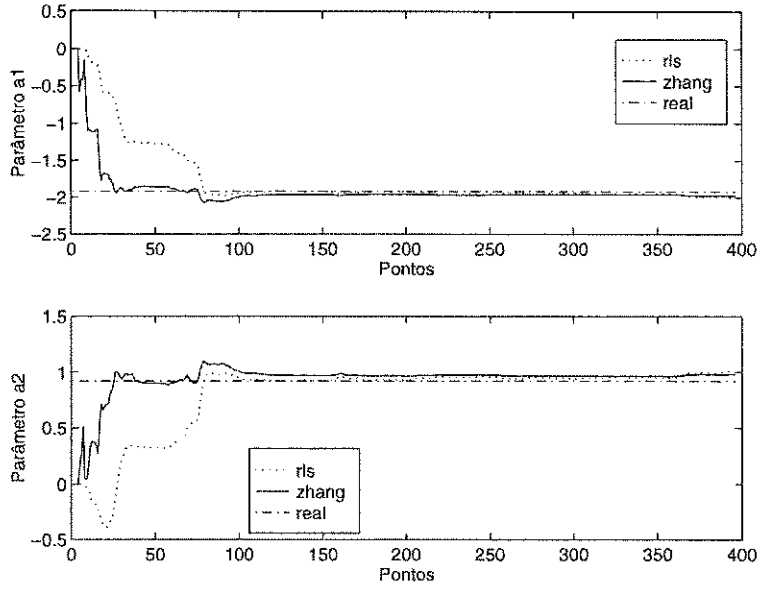

Figura 6.47: Parâmetros a1 e a2 do modelo estimado para a planta P5.

Testando uma excitação montada a partir de um sinal PRBS, para o a planta *P5* nas condições do experimento anterior, foram obtidos resultados que se mostraram semelhantes ao caso anterior, como pode ser observado nas Figuras 6.51, 6.52 e 6.53. Na Figura 6.54, é apresentado o erro de predição para as estimações paramétricas da planta. O nível de ruído foi de $N/S = 0,8789$.

Conclui-se que, para os casos estudados, o método recursivo de Zhang se mostrou superior em estabilidade, velocidade de convergência e precisão dos parâmetros obtidos. A imunidade ao ruído foi maior para o método de Zhang, sendo que o fator de esquecimento mostrou-se mais crítico para o método RLS, mas deve ser cuidadosamente escolhido em ambos casos.

6.3 Aplicação do ISAC a um Processo Real

O ISAC foi utilizado na identificação de um modelo para um processo real composto de um Protótipo do Secador Industrial de Grãos, Figura 6.55. A modelagem foi feita através da realização de um experimento de identificação seguido da redução da ordem do modelo estimado, através da função de custo apresentada em (3.51).

Os dados de entrada aplicados no processo foram gerados a partir de um programa para a Sintonia de Controladores Industriais - SCI, elaborado por Mascarenhas [33].

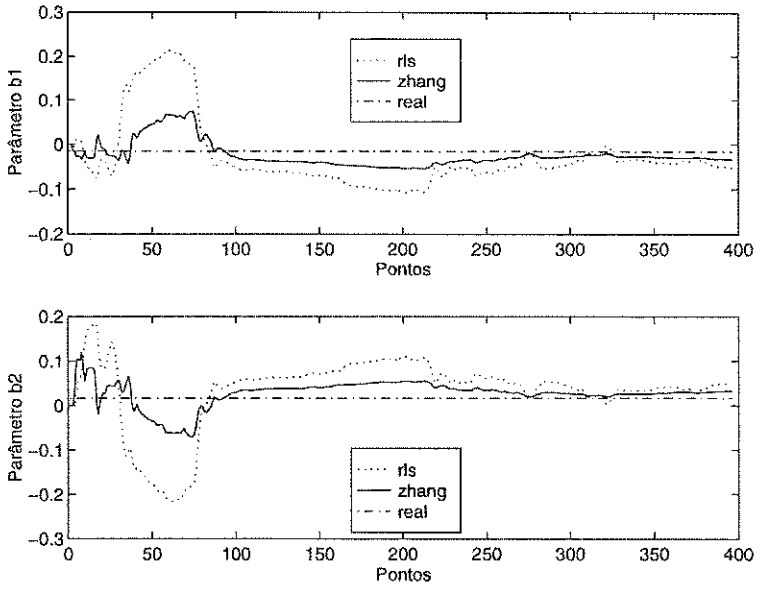

Figura 6.48: Parâmetros b1 e b2 do modelo estimado para a planta P5.

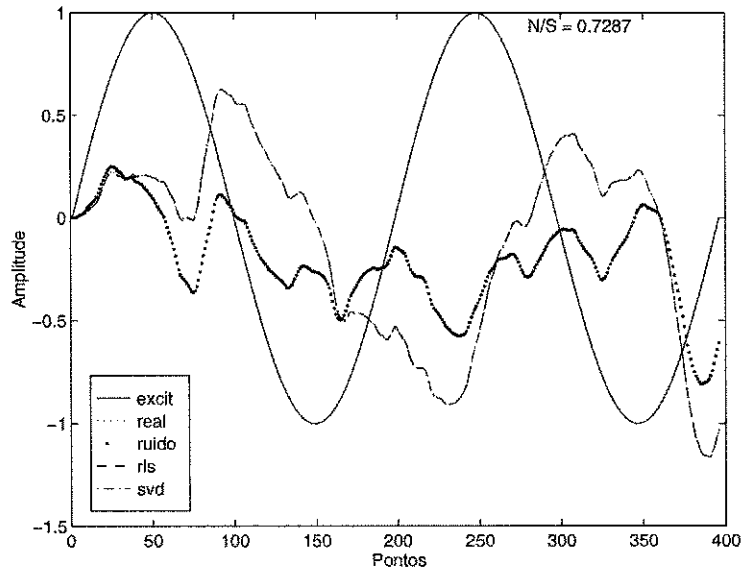

Figura 6.49: Resposta no tempo do modelo estimado para a planta P5.

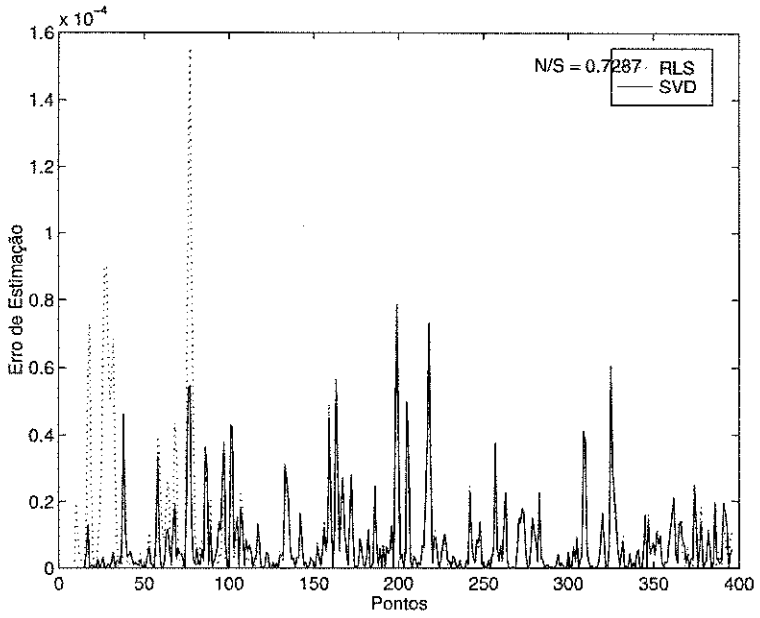

Figura 6.50: Erro de predição para as estimações da planta P1.

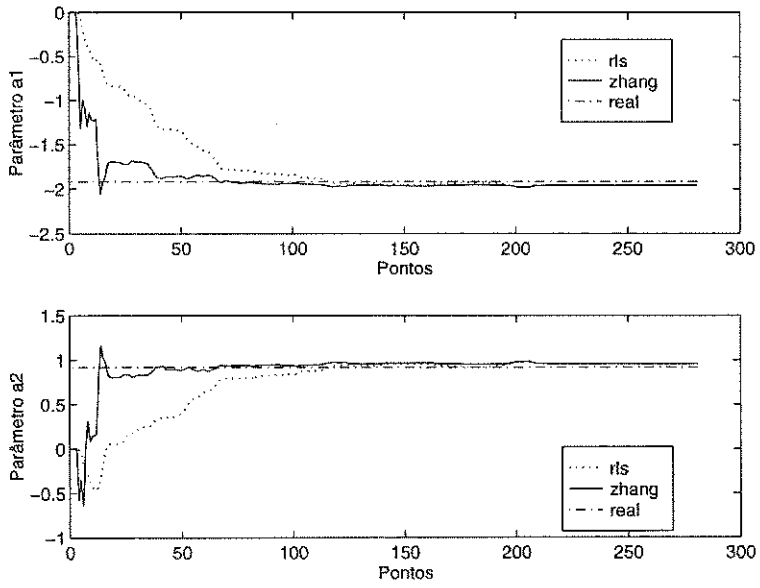

Figura 6.51: Parâmetros a1 e a2 do modelo estimado para a planta P5.

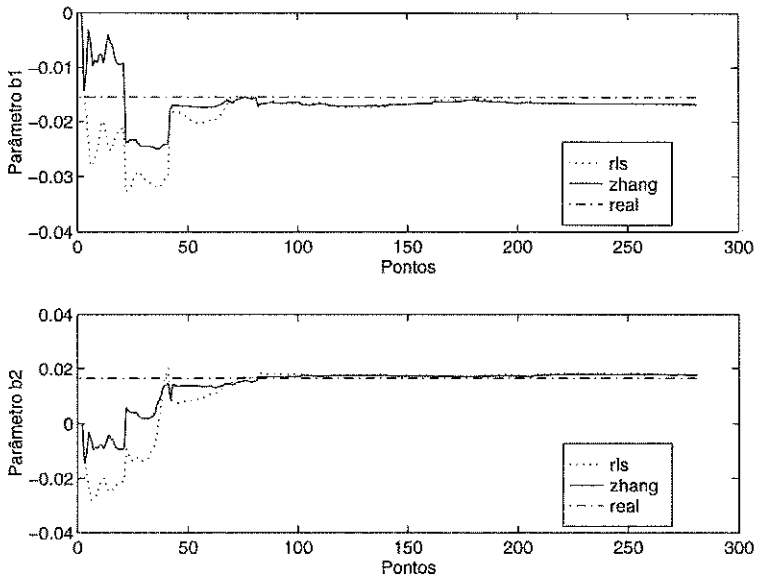

Figura 6.52: Parâmetros b1 e b2 do modelo estimado para a planta P5.

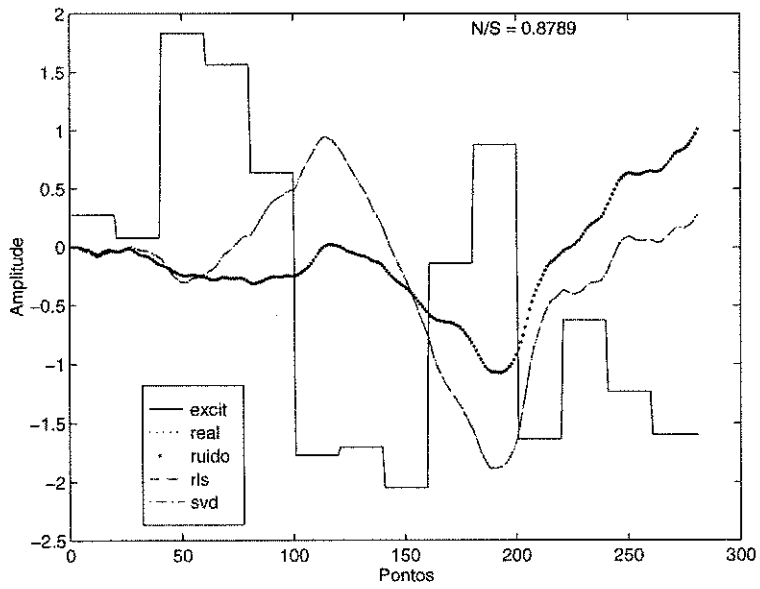

Figura 6.53: Resposta no tempo do modelo estimado para a planta P5.

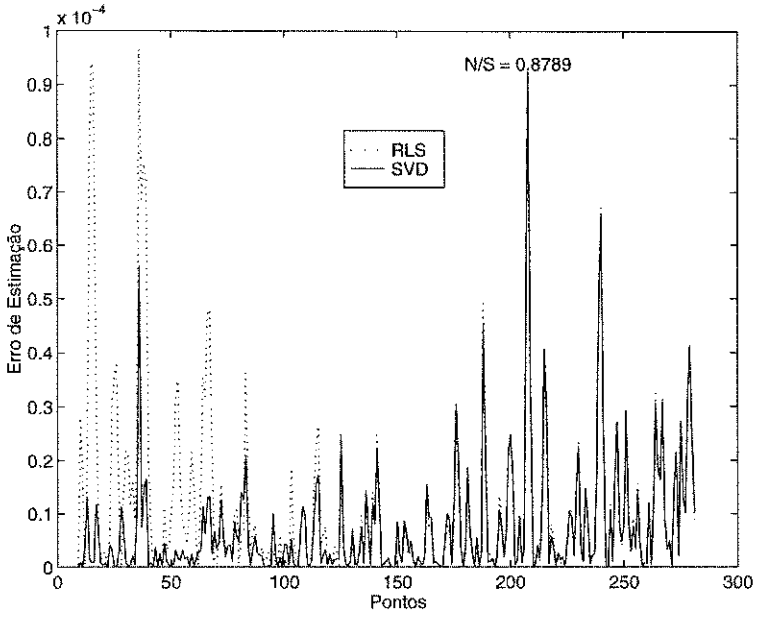

Figura 6.54: Erro de predição para as estimações da planta P1.

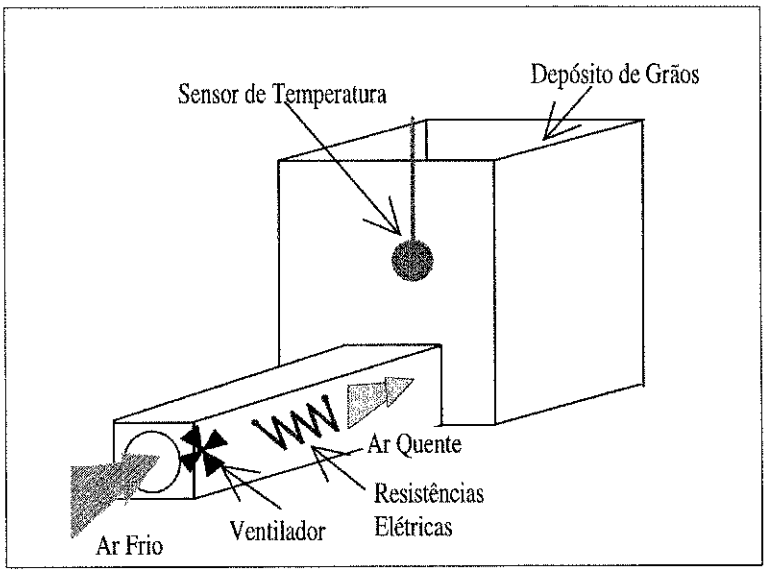

Figura 6.55: Protótipo de secador de grãos industrial.

Na Figura 6.56, é apresentada a tela do ISAC com o modelo estimado para sexta ordem e utilizando o método dos mínimos quadrados não-recursivo.

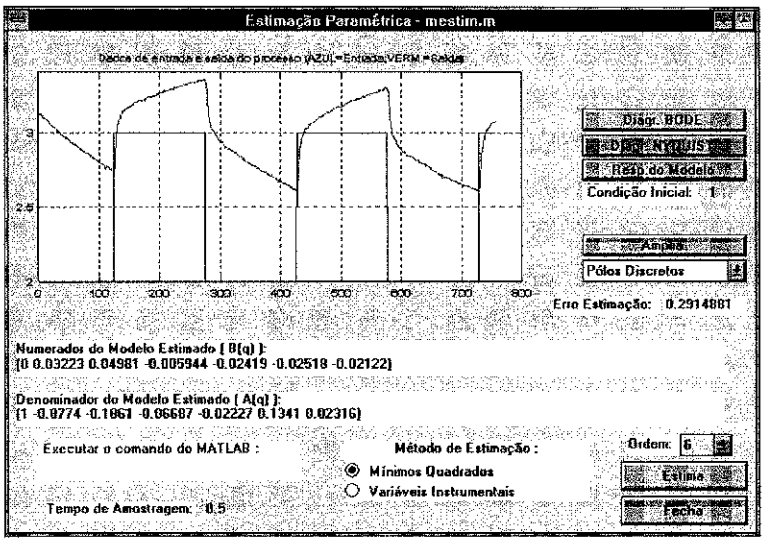

Figura 6.56: Modelo estimado de sexta ordem do processo (dados : cba5.mat).

Pode-se verificar que o modelo obtido na estimação possui a seguinte função de transferência, $F_6(q)$, de sexta ordem,

$$F_6(q) = \frac{0,03223q^{-1} + 0,0498q^{-2} - 0,0059q^{-3} - 0,0242q^{-4} - 0,0252q^{-5} - 0,0212q^{-6}}{1 - 0,8774q^{-1} - 0,1861q^{-2} - 0,0669q^{-3} - 0,0223q^{-4} + 0,1341q^{-5} + 0,2316q^{-6}}. \tag{6.6}$$

Os diagramas de Bode e Nyquist foram obtidos para o modelo estimado, Figuras 6.57 e 6.58, respectivamente.

A resposta deste modelo estimado ao dados de entrada originais do experimento pode ser observada na Figura 6.59, em que condições iniciais foram utilizadas para uma melhor comparação entre os dados originais e a resposta do modelo.

A redução da ordem do modelo foi obtida no ISAC, com a utilização de uma função de custo. A análise gráfica das curvas apresentadas indicam que o modelo do sistema pode ser reduzido para um modelo de terceira ordem. Isto pode ser visto avaliando-se as amplitudes dos valores singulares do gráfico de *valores singulares versus ordem do modelo*, Figura 6.60. Quando a amplitude do valor singular torna-se constante, após ter sofrido uma queda, deve-se tomar a ordem reduzida como sendo a ordem imediatamente anterior, e de amplitude bem maior. Tais procedimentos indicam que esta é uma técnica empírica, que é apresentada em [1]. A função de custo (3.51) também pode ser utilizada para a redução do modelo. Na Figura 6.61, pode ser visto o resultado da redução do modelo com a função de custo, da qual pode ser observada

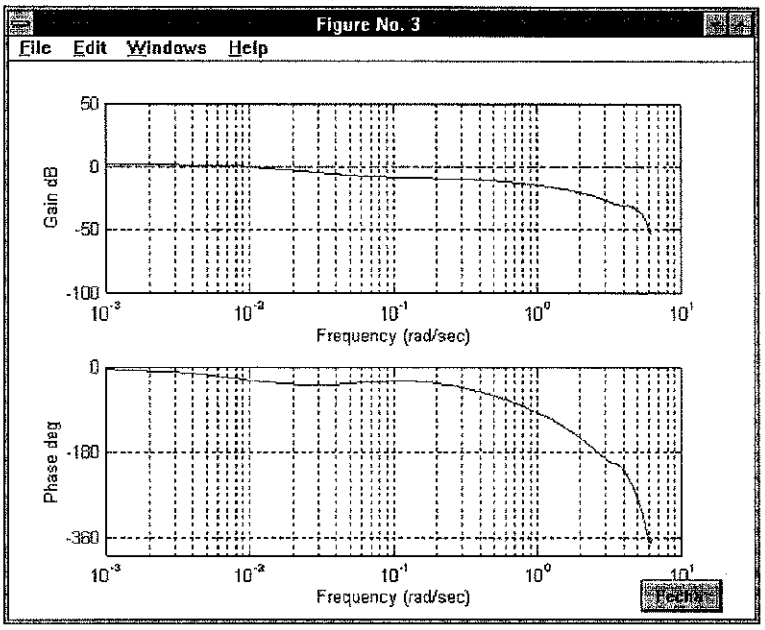

Figura 6.57: Diagrama de Bode do modelo estimado.

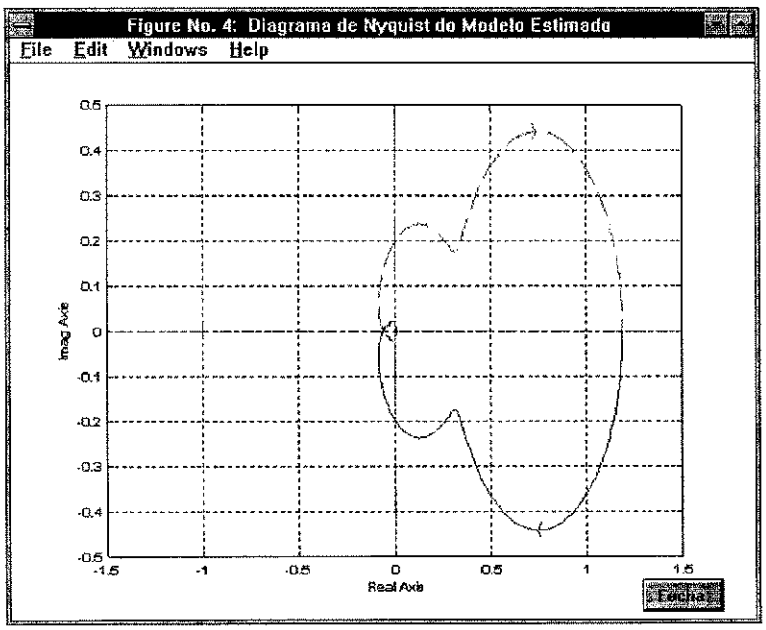

Figura 6.58: Diagrama de Nyquist para o processo estimado.

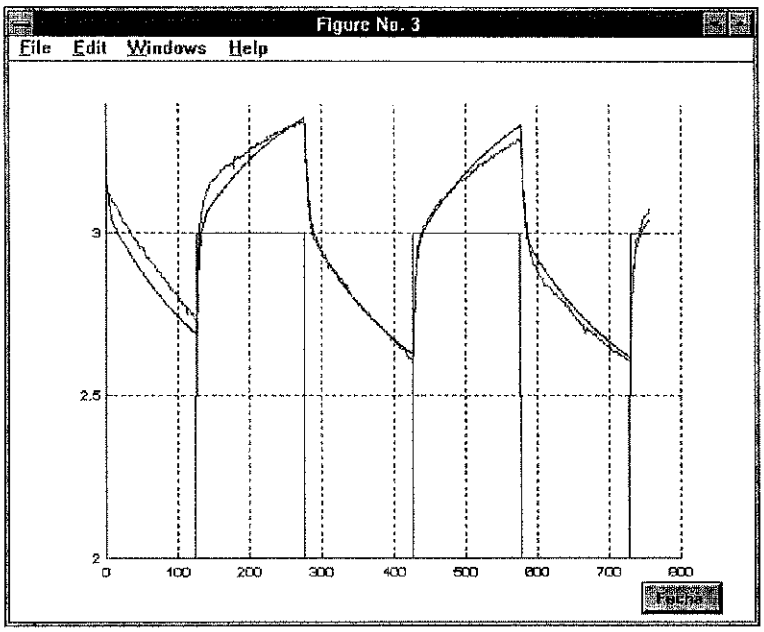

Figura 6.59: Resposta do modelo ao sinal de entrada original, com condições iniciais.

a seguinte função de transferência, $F_3(q)$, de terceira ordem,

$$F_3(q) = \frac{0,03224q^{-1} + 0,03374q^{-2} - 0,01699q^{-3}}{1 - 1,367q^{-1} + 0,6116q^{-2} - 0,126q^{-3}}. \tag{6.7}$$

Após a obtenção do modelo reduzido para terceira ordem, foi realizada uma nova estimação, mas para um modelo de terceira ordem, para comparação entre o modelo reduzido e o modelo estimado diretamente para a ordem mínima da planta. Os resultados obtidos são apresentados nas Figuras 6.62, 6.63, 6.64 e 6.65. Comparando-se os diagramas de Bode e Nyquist e a resposta temporal do modelo de sexta ordem com as mesmas figuras para o modelo de terceira ordem, percebe-se a semelhança no comportamento dos modelos, e portanto, por ter ordem menor o modelo de terceira ordem é uma representação mais adequada para o processo.

Como pode ser observado na Figura 6.62, o modelo estimado para terceira ordem apresentou a seguinte função de transferência,

$$F_3'(q) = \frac{0,03117q^{-1} + 0,03133q^{-2} - 0,05948q^{-3}}{1 - 1,48q^{-1} + 0,2786q^{-2} + 0,204q^{-3}}. \tag{6.8}$$

Estes resultados atestam a eficácia do método de redução de ordem implementado.

6.4 Projeto de Controlador PID com o ISAC

Para exemplificar a aplicação do ISAC para o projeto de controladores PID, baseando-se em um modelo do processo, foram feitos testes com uma planta simulada no próprio

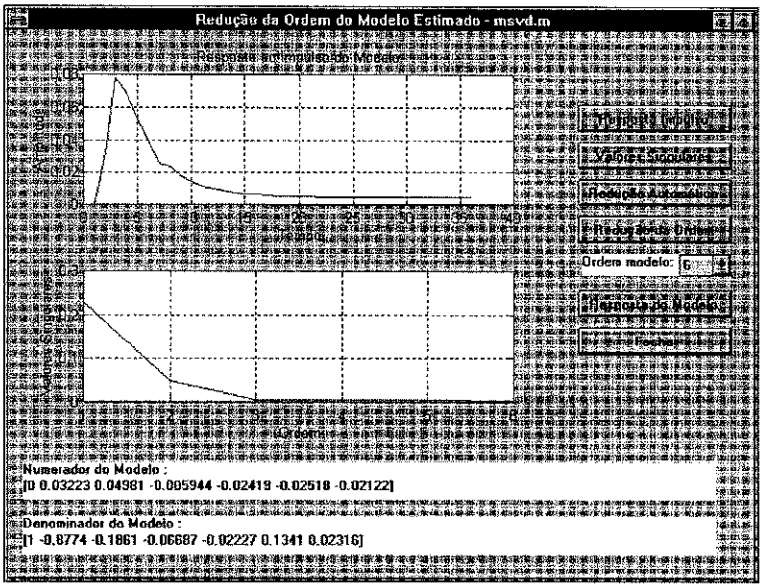

Figura 6.60: Tela para redução da ordem do modelo estimado.

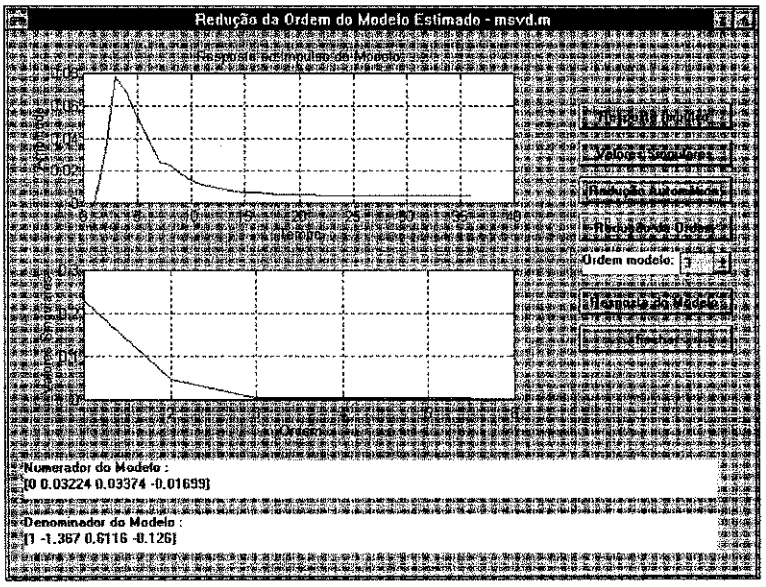

Figura 6.61: Redução automatica da ordem do modelo estimado.

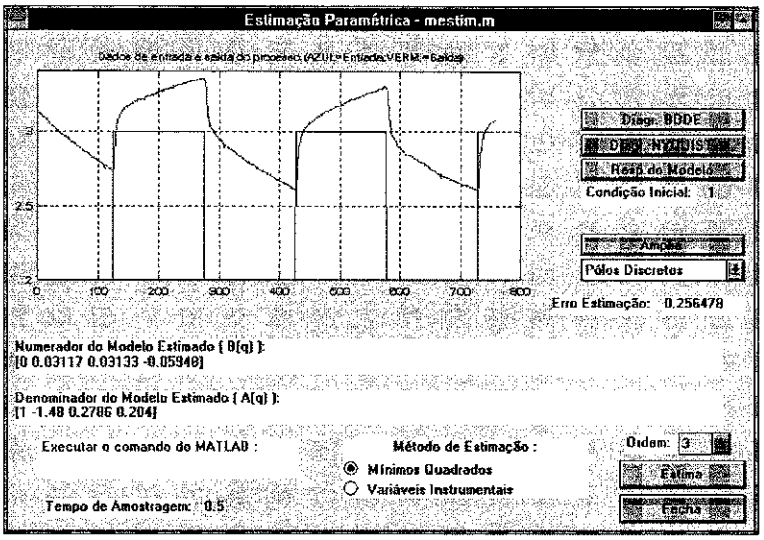

Figura 6.62: Modelo estimado de terceira ordem do processo.

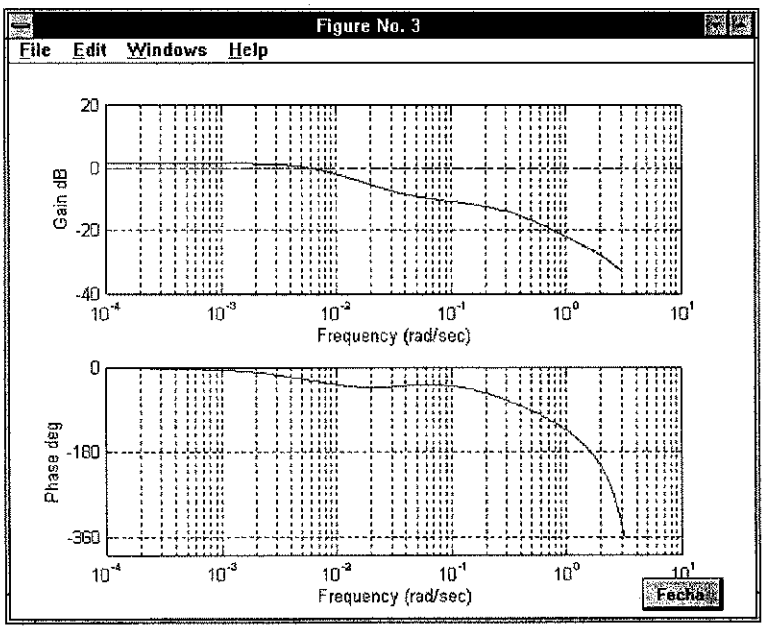

Figura 6.63: Diagrama de Bode da planta.

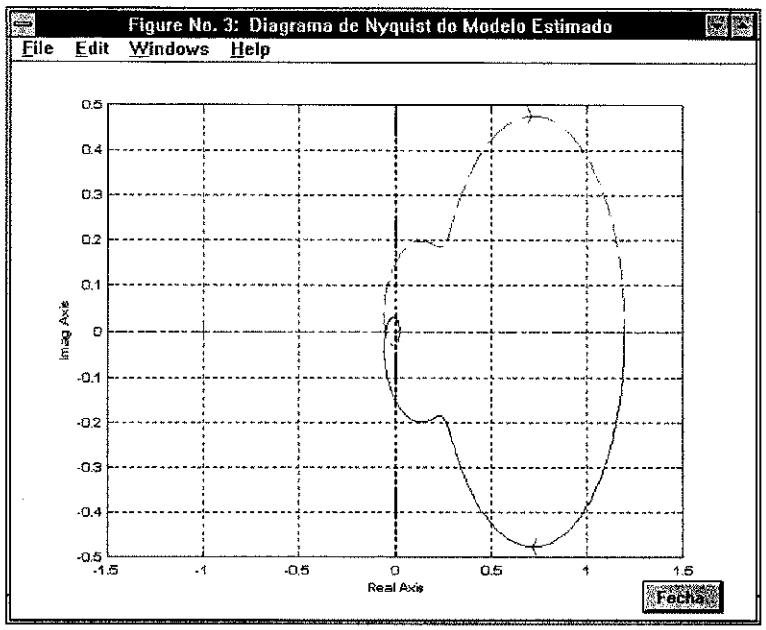

Figura 6.64: Diagrama de Nyquist para o processo estimado.

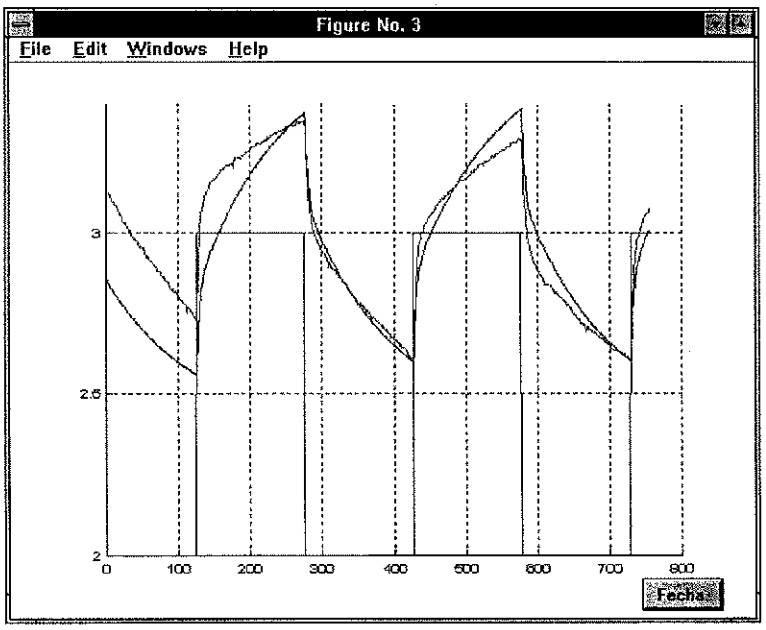

Figura 6.65: Resposta do modelo ao sinal de entrada original, com condições iniciais.

ISAC.

A planta utilizada para a simulação dos dados possui a seguinte função de transferência,

$$G(s) = \frac{1}{(s+1)(2,6s+1)}, \tag{6.9}$$

que foi discretizada utilizando-se o método ZOH, apresentado no capítulo 4, com taxa de amostragem de $T = 0,1$. A planta foi discretizada para ser simulada no próprio ISAC e é apresentada a seguir,

$$G_{ZOH}(q) = \frac{0,0018q^{-1} + 0,0018q^{-2}}{1 - 1,8671q^{-1} + 0,8707q^{-2}} \tag{6.10}$$

O sinal de excitação utilizado no experimento foi uma onda quadrada com 2500 pontos e três períodos, gerada na janela de edição da forma de onda, apresentada na Figura 6.66.

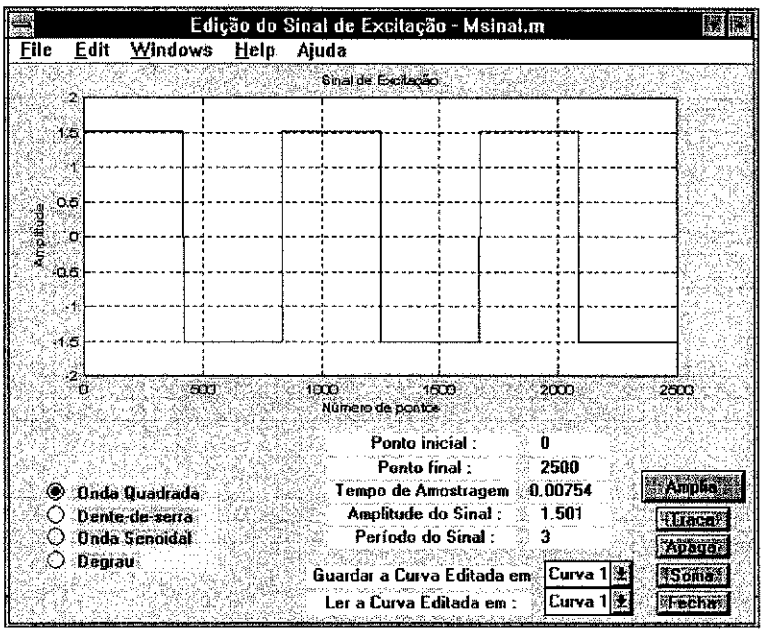

Figura 6.66: Edição do sinal de excitação.

A edição da função de transferência do processo e a simulação para obtenção dos dados é apresentada na Figura 6.67.

Estimou-se um modelo de ordem mais elevada que a ordem real. A janela de estimação é apresentada na Figura 6.68, a comparação dos parâmetros estimados com os parâmetros reais da planta, atestam a eficácia do método utilizado.

Obtido um modelo de ordem maior que dois, torna-se necessária uma redução da ordem para que possa ser efetuado o projeto por alocação de pólos. A janela de redução da ordem é apresentada na Figura 6.69

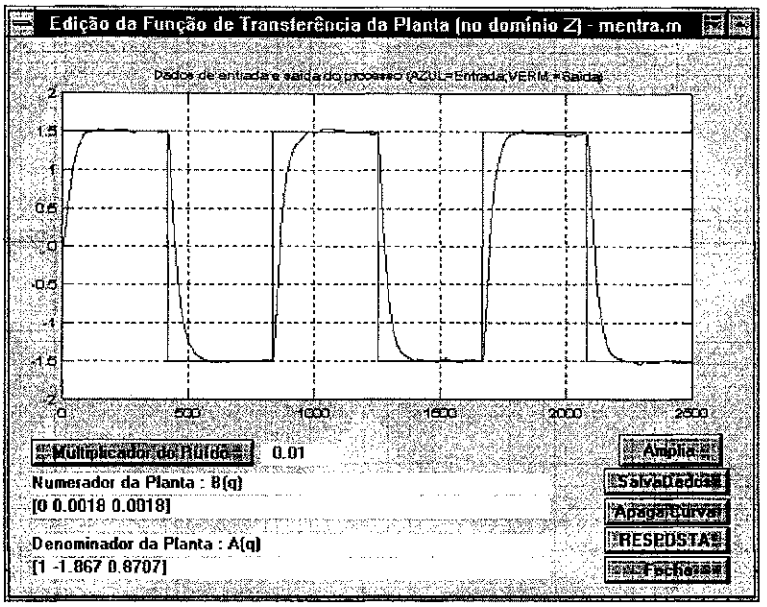

Figura 6.67: Edição e simulação do processo.

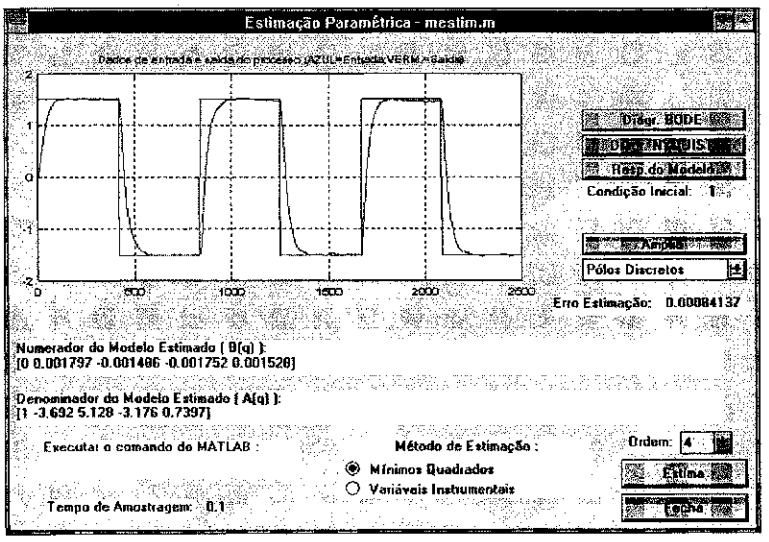

Figura 6.68: Estimação de um modelo de ordem elevada.

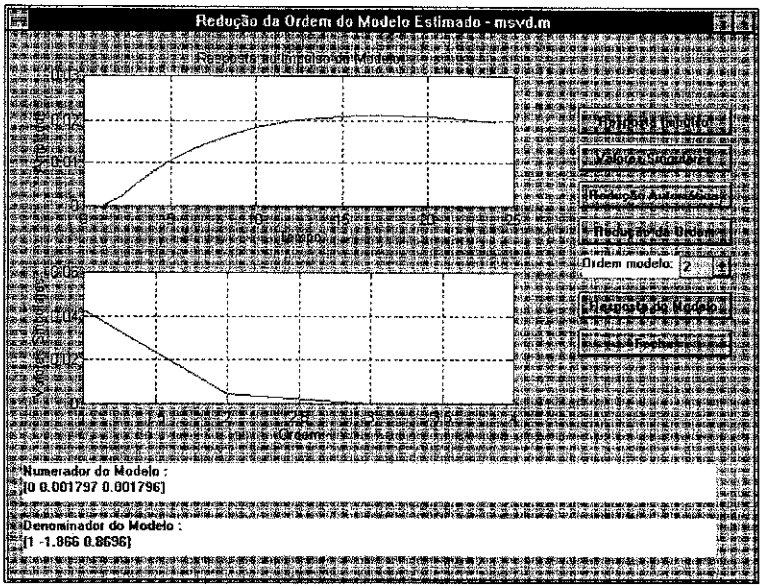

Figura 6.69: Redução da ordem do modelo.

Finalmente, o projeto do controlador PID utilizando-se o modelo estimado e reduzido, e baseado no método de alocação de pólos, é apresentado na Figura 6.70.

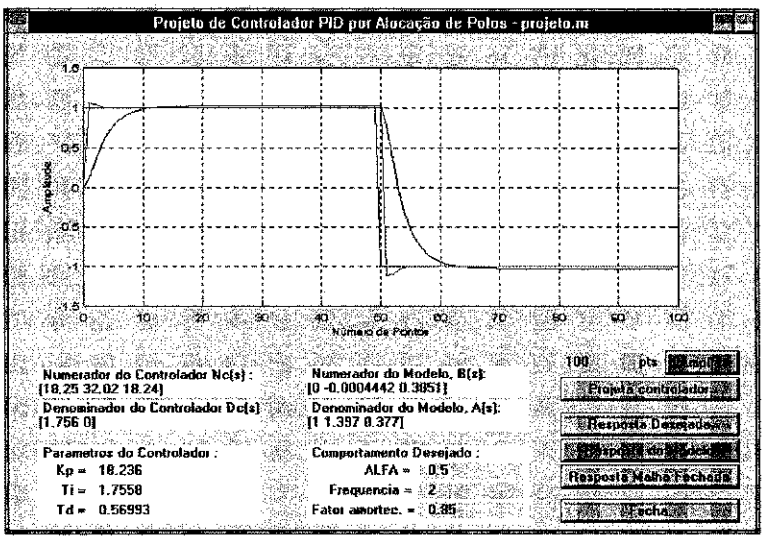

Figura 6.70: Projeto do controlador PID.

O teste com uma planta simples de ser controlada, lenta e "bem comportada", foi utilizado apenas para ilustrar o funcionamento e a facilidade de execução do pacote. Plantas mais complexas e com comportamento distinto podem ser utilizadas neste experimento da mesma forma em que foi apresentado este caso. O tempo necessário para execução de todo o procedimento apresentado nesta seção, não foi maior que cinco minutos, atestando a simplicidade e eficiência do pacote ISAC.

6.5 Conclusões

Foram apresentados os testes no algoritmo recursivo com atualização da matriz de covariancia por SVD (Zhang). O algoritmo de Zhang foi comparado com uma implementação do método dos mínimos quadrados recursivo (RLS). Os resultados mostraram que no método de Zhang, a convergência e estabilidade paramétrica são superiores aos do método RLS. Mesmo com pouca excitação e ruído bastante elevado o método de Zhang ainda proveu resultados melhores. A velocidade de convergência dos parâmetros foi vista com uma das principais vantagens deste método. Isto resulta da utilização da SVD, que mostrou-se uma ferramenta capaz de extrair mais informações dos dados disponíveis do processo, que o método usual dos mínimos quadrados.

Apresentou-se ainda, a aplicação do pacote de rotinas ISAC para um sistema não-linear. O sistema testado foi um protótipo de secador de grãos industrial montado em laboratório. Inicialmente foi feita uma estimação de sexta ordem e depois reduzida, com base em uma função de custo implementada no ISAC, para um sistema de terceira ordem. Para verificar a eficácia desta redução, foi feita uma nova estimação paramétrica, agora para terceira ordem e as funções de transferência obtidas pelos dois modos foram comparadas, juntamente com os sinais de resposta temporal da planta reduzida e da planta estimada para terceira ordem. A utilização do ISAC para o projeto de controladores PID é simples e mostrou-se eficiente.

Capítulo 7

Conclusões Gerais

7.1 Conclusões Gerais

O presente trabalho fornece uma contribuição para usuários de identificação de sistemas, provendo-lhes com algorítmos baseados na SVD, uma técnica de redução de modelos baseada na SVD, como também, com um pacote de rotinas implementada para o ambiente gráfico do MATLAB. O ISAC possui rotinas de fácil execução pelo usuário e contém ferramentas úteis para a obtenção, validação e teste de modelos estimados de processos. O pacote permite que sejam utilizados arquivos de dados amostrados do processo para a obtenção de um modelo. Simulações de plantas editadas no ISAC também podem ser utilizadas em experimentos de identificação.

Foi visto que a definição da qualidade do modelo está associada à aplicação para a qual o modelo está direcionado. Em certas aplicações, como o projeto de controladores baseados em representações matemáticas da planta, tais modelos possuem indiscutível importância. A aplicação do modelo estimado também pode ser entendida como uma forma de qualificar o método de estimação utilizado. A validação do modelo pode ser feita pela comparação do erro médio quadrático entre a saída do modelo e a saída real da planta. Uma outra forma de validação pode ser pela reprodução de dados que não foram utilizados para a estimação, ou seja, tomando-se segmentos de entrada e saída não utilizados nos cálculos e tentando-se reproduzir a saída real da planta.

A qualidade da estimação obtida está diretamente relacionada às características do sinal de excitação. Alguns métodos de identificação são mais sensíveis às perturbações, que outros. Para minimizar estes problemas, uma alternativa é a utilização de algorítmos mais robustos, que possuam rotinas mais estáveis e menos sensíveis a perturbações, como o método de estimação recursiva baseada em SVD.

Os algorítmos de identificação de sistemas implementados foram testados com plan-

tas de comportamentos variados. O método recursivo baseado em SVD mostrou-se superior ao método dos mínimos quadrados recursivo. Para os casos em que havia excitação de pouco persistente e com a presença de níveis consideráveis de perturbação no sistema, o método Zhang ainda foi a melhor opção. Em alguns casos, tal superioridade apresentou-se na forma de uma melhor convergência dos parâmetros estimados, em outros casos, a estabilidade do algorítmo mostrou-se melhor. Foram observados ainda, casos em que a aplicação do modelo final obtido pelo método baseado em SVD propiciou melhores respostas. A velocidade da convergência dos parâmetros estimados foi uma das características positivas mais marcantes do método recursivo baseado na SVD. O preço que se paga pela utilização deste método está no tempo de execução da rotina, que nos testes efetuados foi, aproximadamente, duas vezes maior que o método dos mínimos quadrados recursivo.

O problema da determinação de uma ordem apropriada para o modelo foi mencionado. Algorítmos de redução da ordem do modelo obtido foram apresentados como uma alternativa para o usuário. O método utilizado basea-se na SVD da resposta ao impulso do modelo estimado. Os resultados alcançados atestaram a eficácia do método para a aplicação a sistemas lineares. Há ainda métodos de redução da ordem do modelo que não são baseados na resposta ao impulso do modelo. Tais métodos não foram explorados neste trabalho e estão incluidos na relação de sugestões para trabalhos futuros. Outro modo de redução da ordem do modelo pode ser feita diretamente com o ISAC, através da comparação entre a resposta temporal dos modelos estimados para diversas ordens. Logo, para isto tornam-se necessárias várias execuções das rotinas de identificação do programa, tornando a opção mais trabalhosa. O terceiro método de redução mencionado é empírico, sendo que, para aplicações automáticas o método matemático é a melhor opção. Na aplicação do método de redução de ordem estudado a um sistema não linear, como foi o caso do protótipo de um secador de grãos industrial, este apresentou resultados satisfatórios com o método de redução baseada na SVD.

A utilização do ISAC para o projeto de controladores PID mostrou-se uma tarefa simples e que não exige muito tempo do usuário para a realização dos procedimentos.

O ambiente MATLAB é mais adequado para o processamento de arquivos em lote. Para a estimação recursiva de processos reais este mostra-se limitado. A limitação ocorre tanto em sua capacidade de lidar com a entrada ou saída de dados pelo computador, como pela velocidade de execução de suas rotinas, que são interpretadas linha a linha. A utilização de linguagens compiladas, como a linguagem "C", permite um menor tempo na execução das rotinas e com isto obtem-se uma maior faixa de aplicação para os mesmos algorítmos. Algumas rotinas matemáticas mais complexas, como por

exemplo a decomposição em valores singulares, já estão disponíveis em literaturas como em [40].

Apesar de estar pronto para ser utilizado pelos usuários em experimentos de identificação, o ISAC pode ainda receber modificações e otimizações em seu código. Uma janela de ajuda em tempo real, otimizações na interface, inclusão de novos algorítmos de identificação e modificações nas ferramentas para o usuário são alguns exemplos de otimizações que podem ser feitas. A seguir são apresentadas algumas sugestões para trabalhos futuros.

7.2 Sugestões Para Trabalhos Posteriores

Apesar de estar pronto para ser utilizado pelos usuários em experimentos de identificação, o ISAC pode ainda receber modificações e otimizações em seu código. Uma janela de ajuda em tempo real, otimizações na interface, inclusão de novos algorítmos de identificação e modificações nas ferramentas para o usuário são alguns exemplos de otimizações que podem ser feitas. A seguir são apresentadas algumas sugestões para trabalhos futuros.

- O pacote de rotinas implementado pode ser convertido para um programa executável, com a utilização de um programa gerador de códigos fabricado pela *Mathworks Corporation.*
- As rotinas de identificação podem ser implementadas em suas formas recursivas.
- Explorar novos métodos de estimação recursivos e não-recursivos.
- Explorar outras famílias de modelos para o ruído, além dos ARX e ARMAX estudados.
- Executar a estimação de modelos em tempo real com a utilização de placas de conversão analógico/digital, e a inclusão de rotinas na linguagem C.
- Implementar rotinas de redução da ordem do modelo, que não sejam baseadas na resposta ao impulso do modelo [41].
- Implementar rotinas de controle adaptativo que utilizem as rotinas de estimação já implementadas neste trabalho, e aplicar a sistemas reais.
- Implementar rotinas de estimação para sistemas de múltiplas entradas e múltiplas saídas, MIMO.

- Implementar métodos de identificação de subespaço.
- Explorar métodos de estimação para sistemas não-lineares.

Apêndice A

Código Fonte Integral do ISAC

```
% isac.m
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - september,1st,1995.
%
%  Last modify  16:46  3/03/97
%

close all; mlimpa;
%closeall %for t=1:5, close; end;
clear; clear global; clc;

%___criando flags de sinalizacao de eventos.
estimou=0; xf=0; yf=0;  clr=[.9 .9 .9];

global estimou xf yf clr;
figure('number','off','inverthardcopy','on');axis off;
set(gcf,'color',clr-.1,'name','Identificao de Sistemas Auxiliada por Computador',...
    'Units','normalized','position',[0.01 1 0.98 0.91]); %janela principal
text(0,1.03,'ISAC - Identificao de Sistemas Auxiliada por Computador','fontsize',16,...
'fontweight','bold','fontangle','italic');
cor = ['cact=get(gcf,''color'');nc=uisetcolor(cact,''Modificar cor da janela atual''),'...
' set(gcf,''color'',nc);  '];
ui0=uimenu(1,'label','      ','callback','');
ui1=uimenu(1,'Position',[3],'Label','&Processo','Tag','ui1','UserData',''); drawnow;
ui1_1=uimenu(ui1,'callback','mfile','Position',[1],'Label','&Dados de Arquivo');
ui1_2=uimenu(ui1,'Position',[2],'Label','&Simulao da Planta'); drawnow;
ui1_2_1=uimenu(ui1_2,'callback','msinal','Position',[1],'Label','&1 Edite o Sinal de Excitao');
ui1_2_2=uimenu(ui1_2,'callback','mentra','Position',[2],'Label','&2 Edite a F. de
Transferncia');
ui2=uimenu(1,'label','&Armazenar','callback','msave');
ui3=uimenu(1,'label','&Filtrar','callback','filtros');
ui4=uimenu(1,'label','&Estimar','callback','mestim');
ui5=uimenu(1,'label','A&juda','callback','');
ui5_1=uimenu(ui5,'callback','mabout','Position',[1],'Label','&Sobre o ISAC');
ui5_2=uimenu(ui5,'callback','','Position',[2],'Label','A&juda');
ui5_3=uimenu(ui5,'callback',cor,'Position',[3],'Label','&Modificar Cores');
uicontrol(gcf,'position',[.6 .02 .15 .05],'units','normalized','string','Janela Principal',...
```

```
'callback','mprinc');     % [ left down width height ]
uicontrol(gcf,'position',[.8 .02 .15 .05],'units','normalized','string','Abandona',...
'callback','close all, clear all, clc');
% Convoca o menu principal.
  mprinc;
%_____________________limpamemoria_____________________________
clear  ui0  ui1  ui1_1  ui1_2  ui1_2_1  ui1_2_2  ui1_2_3  ui1_3
clear  ui2 ui3 ui4 ui5 ui5_1 ui5_2 ui5_3

%_____________fim do programa_____________________________

% mprinc.m      (chamado por: isac.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

mlimpa;
labels = str2mat(...
'PROCESSO', ...
'FILTRAGEM',...
'ESTIMAO', ...
'REDUO DA ORDEM', ...
'PROJETO DE PID', ...
'ARMAZENA DADOS');
callbacks = [ ...
   ' mprocess'
   ' filtros '
   ' mestim  '
   ' msvd    '
   ' projeto '
   ' msave   '];

pos = [150 150];
escolha('ISAC', 'Escolha o item :', labels,...
    callbacks,pos,'Menu Principal - mprinc.m');

%limpamemoria
clear labels callbacks pos
clear fig z1 z2 z3 r1 r2 r3 cbkstr1 cbkstr2 a1 b1

%_____________fim do programa_____________________________

% mabout.m
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - september,1st,1995.

figure('position',[100 300 400 200],'number','off','Menubar','none',...
'name','SOBRE O ISAC');
text=[ 'ISAC   uma ferramenta para Identificao de Sistemas e Estimao '...
'de parmetros, que poder resolver os problemas de modelagem de '...
'processos em geral. Utiliza algoritmos de ltima gerao com nfase '...
'na Decomposio em Valores Singulares para a reduo da ordem dos '...
```

```
'modelos estimados.                                    '] ;

hh=uicontrol(gcf,'style','text','units','normal',...
             'position',[.1,.1,.8,.8 ],'string',text,...
            'Horizontal','center');
set(hh,'backg',get(gcf,'color'),'foreg',[1 1 1]-get(gcf,'color'))

%__________limpamemoria___________
clear text hh

%___________fim do programa_____________________________

% projeto.m      (chamado por: mprinc.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%

mlimpa;
%__testa se o usuario esta seguindo a ordem correta de navegacao pelo ISAC___
if isempty(y)|isempty(u)|~exist('den'),
   errordlg(str2mat(' Voc deve estimar o modelo do processo,',...
' antes de executar esta rotina.',...
' Por favor, execute as rotinas de estimao do ISAC. '),...
'Erro na Seqoncia de execuo');
   return;
end;

%___________Capturar os valores dos parametros estimados___________
  ord=length(den)-1,

%______O mtodo s funciona para modelos de segunda
ordem_________________________
  if ord~=2,
   errordlg(str2mat(' Este mtodo s funciona adequadamente para modelos de segunda
ordem.',...
' Estime um modelo de segunda ordem para o processo, antes de executar esta
rotina.'),...
'Erro na Ordem do Modelo');
   return;
  end;
%______Converter o modelo estimado de discreto para
continuo_____________________________
[numGp,denGp]=d2cm(num,den,Ta,'zoh'),
Zzeros=roots(numGp);  Ppolos=roots(denGp);
Kp=polyval(numGp,0)/polyval(denGp,0); % Estes valores sero usados adiante
T1=-1/Ppolos(1);  T2=-1/Ppolos(2); % no projeto do controlador.
strden=mat2str(denGp,4);
strnum=mat2str(numGp,4);
%_____Abre a janela onde sero apresentados os itens para projeto PID_____________
fig=figure;
set(gcf,'color',clr,'units','normalized','position',[.01 .05 .84 .75]);
set(gcf,'name',' Projeto de Controlador PID por Alocao de Polos - projeto.m ',...
'number','off','menubar','none');

%_________Mostrar na janela numerador e denominador
```

```
ESTIMADOS_____________________________
uicontrol('Style','text','HorizontalAlignment','left','string',...
' Numerador do Modelo, B(s): ','position',[.4 .264 .3 .065],'backg','w','units','norm');
uicontrol('Style','text','Horiz','left','string',' Denominador do Modelo, A(s): ',...
'position',[.4 .196 .3 .065],'backg','w','units','norm');
 if exist('enum'), clear enum; end;  if exist('eden'), clear eden; end;
 enum=uicontrol('Style','text','horiz','left','string',strnum,'units','norm',...
'posit',[.4 .27 .3 .032],'backg','w','foreg','k');
 eden=uicontrol('Style','text','horiz','left','string',strden,'units','norm',...
'posit',[.4 .20 .3 .032],'backg','w','foreg','k');

%----------Estabelecendo o COMPORTAMENTO DESEJADO para malha fechada----------------
------------
uicontrol('Style','text','Horiz','left','string',' Comportamento Desejado :',...
'position',[.4 .01 .3 .160],'backg','w','units','norm');
uicontrol('Style','text','Horiz','left','string','            ALFA =',...
'position',[.4 .10 .3 .035],'backg','w','units','norm');
uicontrol('Style','text','Horiz','left','string','    Frequencia =',...
'position',[.4 .06 .3 .035],'backg','w','units','norm');
uicontrol('Style','text','Horiz','left','string',' Fator amortec. =',...
'position',[.4 .02 .3 .035],'backg','w','units','norm');

%--------------Mostra os PARAMETROS DO CONTROLADOS-----------------------------------------
---
uicontrol('Style','text','Horiz','left','string',' Parametros do Controlador :',...
'position',[.04 .01 .3 .160],'backg','w','units','norm');
uicontrol('Style','text','Horiz','left','string','   Kp =',...
'position',[.04 .10 .3 .035],'backg','w','units','norm');
uicontrol('Style','text','Horiz','left','string','    Ti =',...
'position',[.04 .06 .3 .035],'backg','w','units','norm');
uicontrol('Style','text','Horiz','left','string','   Td =',...
'position',[.04 .02 .3 .035],'backg','w','units','norm');

%%__________valores Default ( so executa se nao houver passado nesta rotina antes )___
if ~exist('projetou'),
    alfa=0.5;  W=2.0;  csi=0.85;  %1 2 .707 %0.5 6.0 1.3
K=0; Ti=0; Td=0;
end;

%__________Entrada do comportamento desejado para a malha
fechada_____________
l12 = ['vartemp=alfa; alfa=edados(l1),  if isempty(alfa), alfa=vartemp; end; '];
l22 = ['vartemp=W;       W=edados(l2),  if isempty(   W),    W=vartemp; end; '];
l32 = ['vartemp=csi;   csi=edados(l2),  if isempty( csi),  csi=vartemp; end; '];
l1=uicontrol('Style','edit','string',[' ' mat2str(alfa,10)],'callback',l12,...
'position',[.55 .10 .14 .035],'backg','y','units','norm');
l2=uicontrol('Style','edit','string',[' ' mat2str(   W,10)],'callback',l22,...
'position',[.55 .06 .14 .035],'backg','y','units','norm');
l3=uicontrol('Style','edit','string',[' ' mat2str( csi,10)],'callback',l32,...
'position',[.55 .02 .14 .035],'backg','y','units','norm');

%__________Calculo dos ganhos por Alocao de polos e mostra Gc para usurio_______
mostraGc=['strdenGc=mat2str(denGc,4);  strnumGc=mat2str(numGc,4); '...
   'if exist(''enumGc''), clear enumGc; end;     '...
   'if exist(''edenGc''), clear edenGc; end;     '...
```

```
   'enumGc=uicontrol(''Style'',''text'',''horiz'',''left'',''string'',strnumGc,   '...
   ' ''units'',''norm'',''posit'',[.04 .266 .3 .032],''backg'',''w'',''foreg'',''k''); '...
   'edenGc=uicontrol(''Style'',''text'',''horiz'',''left'',''string'',strdenGc,   '...
   ' ''units'',''norm'',''posit'',[.04 .200 .3 .032],''backg'',''w'',''foreg'',''k''); '];

cgan = ['projetou=1;  '...
'K =(T1*T2*(W^2)*(1+2*csi*alfa)-1)/Kp; '...
'Ti=(T1*T2*(W^2)*(1+2*csi*alfa)-1)/(T1*T2*alfa*(W^3)); '...
'Td=(T1*T2*W*(1+2*csi)-T1-T2)/(T1*T2*(W^2)*(1+2*csi*alfa)-1); '...
'set(Rgan1,''string'',mat2str(K,5)); '...
'set(Rgan2,''string'',mat2str(Ti,5)); '...
'set(Rgan3,''string'',mat2str(Td,5)); '...
'denGc=[Ti 0];  numGc=[K*Td*Ti  K*Ti  K]; '...
'eval(mostraGc); '];

%-------------------mostra os ganhos calculados----------------------------------------
Rgan1=uicontrol('Style','text','Horiz','left','string',' 0',...
'position',[.12 .10 .21 .035],'backg','w','units','norm');
Rgan2=uicontrol('Style','text','Horiz','left','string',' 0',...
'position',[.12 .06 .21 .035],'backg','w','units','norm');
Rgan3=uicontrol('Style','text','Horiz','left','string',' 0',...
'position',[.12 .02 .21 .035],'backg','w','units','norm');

%------Sinal de referncia para excitar a malha fechada-------------------------------
montaref=['clear t2 U2; t2=0:numpts-1;U2=ones(1,numpts/2);U2=[U2 -1*ones(1,numpts/2)];'];

if ~exist('t2'),
 numpts=100;     % o nmero de pontos pode ser modificado pelo usurio...
 eval(montaref);
end;

defref=['vartemp=numpts, numpts=edados(dref),   '...
'if isempty(numpts), numpts=vartemp, end;  '...
'eval(montaref);  '];
dref=uicontrol('Style','edit','string',[' ' mat2str(numpts,8)],...
'callback',defref,'position',[.74 .32 .105 .035],'backg','y','units','norm');
uicontrol('Style','text','Horiz','left','string','pts','position',[.845 .32 .03 .035],...
'backg','y','units','norm');


%------Mostrar na janela numerador e denominador do CONTROLADOR----------------------
---
uicontrol('Style','text','HorizontalAlignment','left','string',' Numerador do Controlador Nc(s) : ',...
'position',[.04 .264 .3 .064],'backg','w','units','norm');
uicontrol('Style','text','Horiz','left','string',' Denominador do Controlador Dc(s) : ',...
'position',[.04 .196 .3 .064],'backg','w','units','norm');

%--------------Curvas de resposta do sistema-------------------------------------------

ajustegraf=['set(sb,''color'',''w''); set(sb,''xcolor'',''k'',''ycolor'',''k''); '...
    'set(sb,''fontsize'',8,''ygrid'',''on'',''xgrid'',''on''), '...
    'xlabel('' Nmero de Pontos '',''color'',''k'');  '...
    'ylabel('' Amplitude '',''color'',''k'');   '];

compdes=['NUMdes=[0 0 0 alfa*W^3]; DEN1=[1 2*W*csi W^2]; DENdes=conv([1
```

```
alfa*W],DEN1); '...
   'if exist(''X2''), clear X2, end; '...
 'X2=lsim(NUMdes,DENdes,U2,t2); cla; axis off;  '...
 'sb=subplot(''position'',[.1 .45 .8 .48]); '...
 'plot(t2,X2,''r'',t2,U2,''b'');  '...
 'eval(ajustegraf); '];

respmod=['if exist(''X2''), clear X2, end; '...
 'X2=lsim(numGp,denGp,U2,t2); cla; axis off; '...
 'sb=subplot(''position'',[.1 .45 .8 .48]); '...
 'plot(t2,X2,''r'',t2,U2,''b'');  '...
 'eval(ajustegraf); '];

respmf=['eval(cgan);   '...
'NUM_MA=conv(numGc,numGp); NUM_MA=polyzer(NUM_MA);  '...
'DEN_MA=conv(denGc,denGp); [MA,MB,MC,MD] = tf2ss(NUM_MA,DEN_MA);'...
'MAF=MA-MB*MC;  '...
'if exist(''X2''), clear X2, end; '...
'X2=lsim(MAF,MB,MC,MD,U2,t2);  cla; axis off; '...
'sb=subplot(''position'',[.1 .45 .8 .48]); '...
'plot(t2,X2,''r'',t2,U2,''b'');  hold on; '...
'X3=lsim(numGp,denGp,U2,t2);   plot(t2,X3,''k'');  '...
'NUMdes=[0 0 0 alfa*W^3]; DEN1=[1 2*W*csi W^2];   '...
'DENdes=conv([1 alfa*W],DEN1); '...
'X4=lsim(NUMdes,DENdes,U2,t2); plot(t2,X2,''m''); '...
'eval(ajustegraf);  eval(mostraGc); hold off; '];

plotar=['sb=subplot(''position'',[.1 .45 .8 .48]);   '...
'plot(1:length(y),y,''r'',1:length(u),u,''b''); '...
'eval(ajustegraf); '];
eval(plotar);

%_____________Desenha botes de ao para a
janela_________________________________
uicontrol('Style','push','position',[.74 .258 .24 .05],'string','Projeta controlador',...
'callback',cgan,'fore','w','units','normalized');
uicontrol('Style','push','position',[.74 .186 .24 .05],'string','Resposta Desejada',...
'callback',compdes,'fore','w','units','normalized');
uicontrol('Style','push','position',[.74 .133 .24 .05],'string','Resposta do Modelo',...
'callback',respmod,'fore','w','units','normalized');
uicontrol('Style','push','position',[.74 .080 .24 .05],'string','Resposta Malha Fechada',...
'callback',respmf,'fore','w','units','normalized');

%___________apaga as variveis desnecessrias para as outras partes do
programa___________
%sailimpo=['mprinc'];
sailimpo=['clear numGp denGp Zzeros Ppolos Kp T1 T2 strden strnum fig enum eden  '...
  '      l12 l22 l32 l1 l2 l3 mostraGc cgan Rgan1 Rgan2 Rgan3; '...
  'clear t2 U2 X2 ajustegraf compdes respmod respmf plotar sb MA MB MC MD '...
  '      MAF NUM_MA NUMdes DEN1 DENdes DEN_MA C_zoom uiz Z0 strdenGc
strnumGc;  '...
  'clear denGc numGc num den cor respcond sa rastr ETa LeTa edenGc enumGc
defref '...
  '      numpts dref;   mprinc  '];
uicontrol('Style','push','position',[.74 .010 .24 .05],'string','Fecha','callback',sailimpo,...
```

```
'fore','w','units','normalized');

%_________________________Estabelece boto de ZOOM_________________________
global ZO;
C_zoom = ['if ZO, set(uiz,''string'',''Congela''); ZO=0; set(gcf,''pointer'',''crosshair''); '...
  ' else set(uiz,''string'',''Amplia''); ZO=1; set(gcf,''pointer'',''arrow''); end; zoom '];
uiz=uicontrol(gcf,'style', 'push','units','normal','position', [.88 .31 .10 .05],'string',...
'Amplia','callback','eval(C_zoom)');
set(gcf,'units','pixels');
ZO=1;

%_____________fim do programa_____________________________________

% mprocess.m     (chamado por: mprinc.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

mlimpa;
labels = str2mat(...
   'DADOS DE ARQUIVO', ...
   'SIMULAO DA PLANTA', ...
   'RETORNA');
callbacks = [ ...
   'mfile '
   'mexper'
   'mprinc'];

pos = [150 180];
escolha('PROCESSO', 'Escolha como quer os dados para a estimao :',...
      labels, callbacks,pos,'PROCESSO - mprocess.m');

%_____limpamemoria_______
clear labels callbacks pos

%_____________fim do programa_____________________________________

% filtros.m      (chamado por: mprinc.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - january,12nd,1995.

 global u y fig ordem Ta;
 global A B;
 mlimpa;

%____Testa se o usuario esta seguindo a ordem correta de navegacao pelo ISAC_____
if isempty(y)|isempty(u),
 load cba5;
end;

fig=figure;
set(gcf,'color',clr,'units','norm','position',[.01 .05 .85 .75],'name',...
```

```
'Filtragem dos Dados - filtros.m','number','off','menubar','none');

  Saif=y;   % Abre espaco para o sinal de saida   que sera filtrado
  Entf=u;   % Abre espaco para o sinal de entrada que sera filtrado

%__Rotinas p/ tornar exclusivos os botes tipo 'radio', e definir a escolha desejada.___
 z1 = [ 'set(r2,''value'',0);' 'set(r3,''value'',0);' 'set(r4,''value'',0);' 'flagfilt=1;'];
 z2 = [ 'set(r1,''value'',0);' 'set(r3,''value'',0);' 'set(r4,''value'',0);' 'flagfilt=2;'];
 z3 = [ 'set(r1,''value'',0);' 'set(r2,''value'',0);' 'set(r4,''value'',0);' 'flagfilt=3;'];
 z4 = [ 'set(r1,''value'',0);' 'set(r2,''value'',0);' 'set(r3,''value'',0);' 'flagfilt=4;'];
 z5 = ['set(r5,''value'',0); set(r6,''value'',1);' 'faixa=1;'];
 z6 = ['set(r5,''value'',1); set(r6,''value'',0);' 'faixa=0;'];

%_______Rotinas para os botes tipo 'radio'.
-------------------------------------------------
 r1=uicontrol('Style','radio','string','Butterworth ','units','norm','position',...
[.85 .64 .145 .042],'CallBack',z1,'backg','w');
 r2=uicontrol('Style','radio','string','Chebychev ','units','norm',  'position',...
[.85 .60 .145 .042],'CallBack',z2,'backg','w');
 r3=uicontrol('Style','radio','string','Bessel ','units','norm','position',...
[.85 .56 .145 .042],'CallBack',z3,'backg','w');
 r4=uicontrol('Style','radio','string','Elptico ','units','norm','position',...
[.85 .52 .145 .042],'CallBack',z4,'backg','w');
 r5=uicontrol('Style','radio','string','Passa-Baixas ','position',...
[.85 .44 .145 .042],'CallBack',z5,'backg','w','units','norm','value',1);
 r6=uicontrol('Style','radio','string','Passa-Altas ','position',...
[.85 .40 .145 .042],'CallBack',z6,'backg','w','units','norm');

%____________________Valores Default_____________________________________
 set(r1,'value',1);  flagfilt=1;
if ~exist('Ta'), Ta=0.5; end;
if ~exist('ord'), ord=1; end;
if ~exist('freqcorte'), freqcorte=[0.98 0.99]; end;
if ~exist('Xmult'), Xmult=1.1; end;
if ~exist('ripple'), ripple=3; end;
if ~exist('Rp'), Rp=3; end;
if ~exist('Rs'), Rs=40; end;
if ~exist('faixa'), faixa=1; end;  % (faixa=0->filtro PB ou PF),(faixa=1->filtro PA ou RF)

%_____Rotinas de tarefas ("callbacks") para os botes tipo 'radio'._____________

 filtrando=['if length(freqcorte)==1, X0=zeros(ord,1); else, X0=zeros(ord*2,1); end; '...
'tipfilt(flagfilt,ord,freqcorte,ripple,Rp,Rs,faixa); '...
'[a,b,c,d]=tf2ss(B,A); Np=length(Saif);  '...
'[yf,x]=dlsim(a,b,c,d,Saif,X0);  '...
'[yf,x]=dlsim(a,b,c,d,Saif,Xmult*x(Np-Np/6,:)); '...
'[xf,x]=dlsim(a,b,c,d,Entf,X0);  '...
'[xf,x]=dlsim(a,b,c,d,Entf,Xmult*x(Np-Np/6,:)); '...
        'axes(sb1); plot(xf,''k''); eval(ajusta); '...
        'axes(sb2); plot(yf,''k''); eval(ajusta); hold off;  '];

 diabode=['if ~exist(''a'')|~exist(''b'')|~exist(''c'')|~exist(''d''), return; end;'...
  'figure; dbode(a,b,c,d,Ta);  '...
  'uicontrol(''Style'',''push'',''position'',[.85 .01 .11 .06],''string'', '...
  ' ''Fecha'',''callback'',''close'',''units'',''norm'');            '];
```

```
fechando=['clear r1 r2 r3 r4,' 'USF=get(USF,''value'');' 'mprinc'];
  gravem=['vlr=get(Gravem,''string''); str=[vlr ''= yf;'']; eval(str); '];

%_____________________________Rotinas para os botes
-----------------------------------------------
uicontrol('Style','text','position',[.04 .146 .25 .041],'string',...
' Condies iniciais do filtro :','backg','w','Horiz','left','units','norm');
uicontrol('Style','text','position',[.04 .104 .25 .041],'string',...
' Ordem desejada p/ o filtro :','backg','w','Horiz','left','units','norm');
uicontrol('Style','text','position',[.04 .062 .25 .041],'string',...
' Frequncias de corte  :','backg','w','Horiz','left','units','norm');
uicontrol('Style','text','position',[.04 .020 .25 .041],'string',...
' Ripple em dB (p/ Cheby) :','backg','w','Horiz','left','units','norm');
USF=uicontrol('Style','radio','position',[.48 .02 .33 .041],'string',...
'Utilizar o sinal filtrado na estimao','backg','w','Horiz','left','units','norm');
uicontrol('Style','text','position',[.48 .08 .2 .041],'string',...
'Tempo de Amostragem:','backg','w','Horizont','left','units','norm');

%---Avisa ao usuario q se a order de Wn for > 1 entao o filtro tera ordem 2*N--(vide help
butter)
avisa=['uicontrol(''Style'',''text'',''position'',[.39 .104 .05 .037],''string'',''( x 2 )'','...
' ''units'',''norm'',''backg'',''w''); set(r5,''string'',''Passa-Faixa''); '...
' set(r6,''string'',''Rejeita-Faixa''); '];
noavisa=['uicontrol(''Style'',''text'',''position'',[.39 .104 .05 .037],''units'',''norm'','...
' ''backg'',''w''); set(r5,''string'',''Passa-Baixas''); '...
' set(r6,''string'',''Passa-Altas''); '];
if length(freqcorte)>1, eval(avisa); else, eval(noavisa); end;

%-------Entrada dos dados para definio do filtro desejado----------------------------------
ETa  =['Ta=edados(LeTa); ' ];
Emult=['Xmult=edados(Lemult); eval(filtrando);' ];
Eord =['ord=edados(Leord); eval(filtrando);' ];
Efreq=['freqcorte=edados(Lefreq);eval(filtrando); '...
       'if length(freqcorte)>1, eval(avisa); else, eval(noavisa); end;' ];
Eripp=['ripple=edados(Leripp); eval(filtrando);' ];

  LeTa=uicontrol('Style','edit','position',[.69 .080 .10 .041],'callback', ETa,...
'string',[' ' mat2str(Ta,8)], 'backg','w','Horizont','left','units','norm');
Lemult=uicontrol('Style','edit','position',[.292 .146 .15 .0378],'callback',Emult,...
'string',[' ' mat2str(Xmult,8)],'backg','w','units','norm');
 Leord=uicontrol('Style','edit','position',[.292 .104 .095 .0378],'callback',Eord,...
'string',[' ' mat2str(ord,8)],'backg','w','units','norm');
Lefreq=uicontrol('Style','edit','position',[.292 .062 .15 .0378],'callback',Efreq,...
'string',[' ' mat2str(freqcorte,8)],'backg','w','units','norm');
Leripp=uicontrol('Style','edit','position',[.292 .020 .15 .0378],'callback',Eripp,...
'string',[' ' mat2str(ripple,8)],'backg','w','units','norm');

 tracey=['axes(sb1); hold on; plot(Entf,''b''); eval(ajusta); hold off; '...
         'axes(sb2); hold on; plot(Saif,''r''); eval(ajusta); hold off; '];

 uicontrol('Style','push','units','norm','position',[.835 .19 .16 .05],'string',...
'D.BODE do filtro','callback',diabode,'clip','off');
 uicontrol('Style','push','units','norm','position',[.835 .13 .16 .05],'string',...
'Sinal original','callback',tracey);
 uicontrol('Style','push','units','norm','position',[.835 .07 .16 .05],'string',...
```

```
'Filtrar','callback',filtrando);
 uicontrol('Style','push','units','norm','position',[.835 .01 .16 .05],'string',...
'Fechar','callback',fechando);

 sb1=subplot('position',[.04 .24 .76 .33]);   p1=plot(Entf,'b');
 sb2=subplot('position',[.04 .61 .76 .33]);   p2=plot(Saif,'r');

 ajusta=[' set(sb1,''color'',''w''); set(sb1,''xcolor'',''k'',''ycolor'',''k'');'...
 ' set(sb1,''fontsize'',8,''ygrid'',''on'',''xgrid'',''on''),'...
        ' set(sb2,''color'',''w''); set(sb2,''xcolor'',''k'',''ycolor'',''k'');'...
 ' set(sb2,''fontsize'',8,''ygrid'',''on'',''xgrid'',''on''),'];

 eval(ajusta);

%__________________limpamemoria____________________
 clear fig z1 z2 z3 z4 a1 b1 zer zor zrr zgr

% mestim.m      (chamado por: mprinc.m e mmq.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - september,1st,1995.

global u y fig ordem Ta; mlimpa;

%-----------Testa se o usuario carregou dados para a estimao------------------------------
if isempty(y)|isempty(u),
   load cba5;
 end;

fig=figure;
%----ajusta flags de sinalizacao para os outros modulos do isac.
%- estimou => se usuario ja estimou, Btpl =>  o boto de polos do modelo.
 estimou; xf; yf;    % Cria as variaveis.
 if exist('Btpl'), clear Btpl; end;

set(gcf,'color',clr,'units','norm','position',[.01 .05 .84 .75]);
set(gcf,'name',' Estimao Paramtrica - mestim.m ','number','off','menubar','none');

uicontrol('Style','text','Horiz','left','string','Numerador do Modelo Estimado ( B(q) ):',...
'position',[.01 .31 .95 .07],'backg','w','units','norm');
uicontrol('Style','text','Horiz','left','string','Denominador do Modelo Estimado ( A(q) ): ',...
'position',[.01 .21 .95 .07],'backg','w','units','norm');

z1 = [  'set(r1,''value'',1); set(r2,''value'',0);' 'flagmet=1;'];
z2 = [  'set(r1,''value'',0); set(r2,''value'',1);' 'flagmet=2;'];
uicontrol('Style','text','string','Mtodo de Estimao :','position',[.45 .14 .33 .04],...
'backg','w','units','norm');
ri=uicontrol('Style','radio','string',' Mnimos Quadrados       ','position',...
[.45 .10 .33 .042],'CallBack',z1,'backg','w','units','norm','value',1);
r2=uicontrol('Style','radio','string',' Variveis Instrumentais  ','position',...
[.45 .06 .33 .042],'CallBack',z2,'backg','w','units','norm');

%__________valores Default_________________________
if ~exist('Ta'), Ta=0.5; end;
```

```
if ~exist('mestf'),
  set(r1,'value',1); ordem=4; flagmet=1; ordhndl=4; mestf=1;
end;

if exist('USF'),
  if USF,
    ynf=y;  unf=u; %--- Preserva sinais originais.
    y=yf;   u=xf; %--- Copia sinais filtrados para dados.
  else,
    y=ynf;  u=unf; %--- Retorna os sinais originais.
   end
end;

cbkstr1=['ordem=get(ordhndl,''value''); '];
cbkstr2=['metestim(ordem,flagmet);  mostima;    '...
 'if flagmet==1, set(r1,''value'',1); end; '...
 'if flagmet==2, set(r2,''value'',1); end; '...
 'if flagmet==3, set(r3,''value'',1); end; '];

%---------escolha de ordem do modelo estimado-----------------------------------------------
uicontrol('Style','text','posit',[.83 .15 .07 .041],'string','Ordem:',...
'backg','w','Horiz','left','units','norm');
ordhndl=uicontrol('Style','popup','position',[.9 .15 .07 .041],'backg','y',...
'string','1|2|3|4|5|6|7|8|9|10|11|12|13|14','callback',cbkstr1,...
'value',4,'units','norm');

ETa=['Ta=edados(LeTa);'];
LeTa=uicontrol('Style','edit','position',[.25 .02 .1 .041],'string',['  ' mat2str(Ta,8)],...
'backg','y','Horizont','left','units','norm','callback', ETa);
uicontrol('Style','text','position',[.05 .02 .2 .041],'string','Tempo de Amostragem:',...
'backg','w','Horizont','left','units','norm');
uicontrol('Style','push','position',[.82 .08 .17 .06],'string','Estima','callback',cbkstr2,...
'fore','w','units','norm');
uicontrol('Style','push','position',[.82 .01 .17 .06],'string','Fecha','callback','mprinc',...
'fore','w','units','norm');

%' u=dtrend(u); y=dtrend(y);'... % causando problemas no sinal "u".

plotar=[' sb=subplot(''position'',[.04 .50 .65 .42]);  a1=plot(y,''r'');hold;b1=plot(u,''b'');'...
' set(sb,''color'',''w''); set(sb,''xcolor'',''k'',''ycolor'',''k''); '...
' set(sb,''fontsize'',8,''ygrid'',''on'',''xgrid'',''on''),'...
' title('' Dados de entrada e sada do processo
(AZUL=Entrada;VERM.=Sada)'',''color'',''k'');'];
eval(plotar);

if estimou,
   mostima;
else,
   eval(cbkstr2);  %_Executa estimao logo que o usurio escolhe a opo na janela
principal.
end;
%------------fim do programa---------------------------------

% msave.m      (chamado por: mprinc.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
```

```
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

global fname pname;

[fname,pname]=uiputfile('lixo.mat','Armazenamento dos Dados do Experimento');
str=['save ' fname ' u y'];

if exist('u') & exist('y'),
  eval(str);  figure; plot([u y]);
  uicontrol('Style','push','position',[.85 .01 .11 .06],'string','Fecha',...
              'callback','close','units','norm');
end;
%__________limpamemoria_____________________________
clear str fname pname

%___________fim do programa_____________________________

% msvd.m      (chamado por: mprinc.m )
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,25th,1995.
%   REDUCAO DA ORDEM de um sistema Linear estimado

mlimpa;
global g Hsing TETA num den ordem

%___testa se o usuario esta seguindo a ordem correta de navegacao pelo ISAC__
 if ~exist('TETA') | ~exist('ordem'),
   errordlg(str2mat(' Voc deve realizar estimao de parametros antes ',...
    ' de executar esta rotina. '),'Erro na Sequncia de execuo');
   return;
 end;

figure;
set(gcf,'name',' Reduo da Ordem do Modelo Estimado - msvd.m','number','off',...
'menubar','none','position',[40 40 680 500],'color',[.8 .8 .8]);

den=[1; TETA(1:ordem)]';  num=[0; TETA(ordem+1:2*ordem)]';
strden=mat2str(den,4);    strnum=mat2str(num,4);

%___Calculando a resposta ao impulso pelas equaes 3a e 3b_(ARAKI-
ISA'91)__________
  Ng=6*ordem; %__Nmero de pontos da resposta ao
impulso._____________________
 % Para a reduo da ordem, o algoritmo so utiliza os primeiros (2*ordem) pontos da
 % resposta ao impulso do sistema. O tamanho Ng acima  para garantir dados suficientes.
   a=den;  b=num;   g=zeros(Ng,1);
   a(ordem+2:Ng)=zeros(1,Ng-ordem-1); %__Inicializao p/ os coefic.da resp. impulso.
   b(ordem+2:Ng)=zeros(1,Ng-ordem-1); %__Para um modelo estimado de quarta
ordem_____
   g(1)=b(1)/a(1);
   for i=2:Ng;
```

```
      som=0;
      for j=1:i-2;  som=som+g(j+1)*a(i-j);   end;
      g(i,1)=(b(i)-som)/a(1);
   end;

%_____Monta as matrizes de Hankel H1 e H2 da resposta ao
impulso._____________________
H1=hankel2(g(2:2*ordem+1)',ordem);
H2=hankel2(g(3:2*ordem+2)',ordem);
[U,Hsing,V] = svd(H1); %..Calculos dos valores singulares._____Hi=U*S*V'__

V=V';    %__corrige a forma apresentada da decomposio SVD_____________________
UT=U'; VT=V'; %_____Para que possam ser utilizados dentro da string RedOrd (abaixo)

%_____Traa resposta ao
impulso_____________________________________________________
  sb=subplot('position',[.1 .68 .58 .24]);
  plot(g,'b');  set(sb,'color','w');  set(sb,'xcolor','k','ycolor','k');
  set(sb,'fontsize',10,'ygrid','on','xgrid','on'),
  title('Resposta ao Impulso do Modelo','color','k');
  xlabel('Tempo','color','k'); ylabel('Amplitude','color','k');

%_____Traa Valores
Singulares_____________________________________________________
  sb=subplot('position',[.1 .32 .58 .24]);
  plot(diag(Hsing),'b');
  set(sb,'color','w');    set(sb,'xcolor','k','ycolor','k');
  set(sb,'fontsize',10,'ygrid','on','xgrid','on');
  xlabel('Ordem','color','k'); ylabel('Valores Singulares','color','k');

erro2=[ 'errordlg(str2mat('' A ordem da planta j  minima, portanto no poder'','...
''' ser reduzida por este mtodo.''),'' Falha de execuo''); return; '];
erro3=[ 'errordlg(str2mat('' A ordem desejada deve ser menor que a ordem do modelo'','...
''' estimado. Repita a operao.''),'' Dados Incorretos ''); return; '];

%_______Mostrar na janela numerador e denominador
estimados_____________________________
 uicontrol('Style','text','Horiz','left','string','Numerador do Modelo : ',...
'position',[.02 .14 .93 .07],'backgroundcolor','w','units','normalized');
 uicontrol('Style','text','Horiz','left','string','Denominador do Modelo : ',...
'position',[.02 .04 .93 .07],'backg','w','units','norm');
 if exist('enum'), clear enum; end;
 if exist('eden'), clear eden; end;
 enum=uicontrol('Style','text','horiz','left','string',strnum,'units','norm','position',...
[.02 .14 .93 .04],'backg','w','foreg','k');
 eden=uicontrol('Style','text','horiz','left','string',strden,'units','norm','position',...
[.02 .04 .93 .04],'backg','w','foreg','k');

%____Reduo da ordem do
modelo_____________________________________________________
RedAut=[' szH=max(size(Hsing)); '...
' for t=2:szH, FCH(t-1)=Hsing(t-1)-Hsing(t); end; '...
' if ~FCH, eval(erro2); return; end; '...
' disp(''Ordem Reduzida (Mtodo 1) = ''),  '...
' n=find(FCH==max(FCH))+1,'...
```

```
' szH=min(size(Hsing)); epsilon=0.001; '...
  ' SN=0;SD=0; '...
' for i=1:szH, SD=SD+(Hsing(i,i)^2); end; '...
' for i=1:szH, SN=SN+(Hsing(i,i)^2); J(i)=SN/SD;  '...
' JZ(i)=J(i)-1+epsilon; '...
' if ~exist(''achou'')&J(i)>1-epsilon,  '...
' disp(''Ordem Reduzida (Mtodo 2)='',n=i,achou=1;end;end; '...
' set(ordm,''value'',n,''back'',''y''); '...
' S2(:,:)=Hsing(1:n,1:n)^(1/2); '...
' Sm2(:,:)=Hsing(1:n,1:n)^(-1/2); '...
' Aar=Sm2*UT(1:n,:)*H2*VT(:,1:n)*Sm2,'...
' Bar=S2*V(1:n,:)*[1; zeros(ordem-1,1)],'...
' Car=[1 zeros(1,ordem-1)]*U(:,1:n)*S2,'...
' Dar=g(1),'...
' [num,den]=ss2tf(Aar,Bar,Car,Dar,1); '...
' strden=mat2str(den,4);   '...
' strnum=mat2str(num,4); '...
' enum=uicontrol(''Style'',''text'',''horiz'',''left'', '...
' ''string'',strnum,''units'',''norm'',''position'',    '...
' [.02 .14 .93 .04],''backg'',''w'',''foreg'',''k'');   '...
' eden=uicontrol(''Style'',''text'',''horiz'',''left'', '...
' ''string'',strden,''units'',''norm'',''position'',' ...
' [.02 .04 .93 .04],''backg'',''w'',''foreg'',''k''); '];

RedOrd=[' if n>ordem, eval(erro3); return; end; '...
' S2(:,:)=Hsing(1:n,1:n)^(1/2); '...
' Sm2(:,:)=Hsing(1:n,1:n)^(-1/2); '...
' Aar=Sm2*UT(1:n,:)*H2*VT(:,1:n)*Sm2,'...
' Bar=S2*V(1:n,:)*[1; zeros(ordem-1,1)],'...
' Car=[1 zeros(1,ordem-1)]*U(:,1:n)*S2,'...
' Dar=g(1),'...
' [num,den]=ss2tf(Aar,Bar,Car,Dar,1); '...
' strden=mat2str(den,4);   '...
' strnum=mat2str(num,4); '...
' enum=uicontrol(''Style'',''text'',''horiz'',''left'', '...
' ''string'',strnum,''units'',''norm'',''position'',    '...
' [.02 .14 .93 .04],''backg'',''w'',''foreg'',''k'');   '...
' eden=uicontrol(''Style'',''text'',''horiz'',''left'', '...
' ''string'',strden,''units'',''norm'',''position'',' ...
' [.02 .04 .93 .04],''backg'',''w'',''foreg'',''k''); '];

%___Ler a ordem desejada para a reduo do modelo_____________________
 cbord=['n=get(ordm,''value''); '];
 strord=uicontrol('Style','text','position',[.77 .55 .135 .041],'horiz','left',...
 'string','Ordem modelo:','back','w','units','norm');
 ordm=uicontrol('Style','popup','position',[.9 .55 .07 .041],'backg','y',...
'string','1|2|3|4|5|6|7|8|9|10|11|12|13|14','callback',cbord,...
'value',ordem,'units','norm');

%_____Achar a resposta do modelo estimado com condicao inicial_____________
 respcond=['[a,b,c,d]=tf2ss(num,den);  [y2,x]=dlsim(a,b,c,d,u); '...
'figure(''color'',''w''); axes(''xcolor'',''k'',''ycolor'',''k''); '...
'hold on; plot(u,''b'');  plot(y2,''k'');  plot(y,''r''); grid;  '...
'set(gca,''fontsize'',8,''ygrid'',''on'',''xgrid'',''on''),'...
'uicontrol(''Style'',''push'',''position'',[.85 .01 .11 .06],''string'',''...
```

```
' ''Fecha'',''callback'',''close'',''units'',''norm''); '];

%____________Botes 'push' de aes diversas_____________________________
 uicontrol('Style','push','posit',[520 408 140 25],'string','Resposta Impulso',...
'callback','svdsoh(1)','clipping','off');
 uicontrol('Style','push','posit',[520 374 140 25],'string','Valores Singulares',...
'callback','svdsoh(2)','clipping','off');
 uicontrol('Style','push','posit',[520 340 140 25],'string','Reduo Automtica',...
'callback',RedAut,'clipping','off');
 uicontrol('Style','push','posit',[520 306 140 25],'string','Reduo da Ordem',...
'callback',RedOrd,'clipping','off');
 uicontrol('Style','push','position',[520 238 140 25],'string','Resposta do Modelo',...
'callback',respcond,'clipping','off');
 uicontrol('Style','push','posit',[520 204 140 25],'string','Fecha',...
'callback','mprinc','clipping','off');

%_____limpamemoria_____________________________________________________
clear DEN NUM a b som i j sb RedOrd RedAut t FCH achou n

%____________fim do programa_____________________________________

% mfile.m     (chamado por: mprocess.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

global fname pname;

[fname,pname]=uigetfile('*.mat','Abre arquivos de dados amostrados');

if fname,
  load (fname)
end;
if ~exist('sbmfile') & ~fname,
  load cba5
end;
  if exist('sbmfile'), sbmfile; gca; axes(sbmfile); cla; axis off; end;
  sbmfile=subplot('position',[.06 .15 .9 .7]);
  plot(u,'b'); drawnow; hold on;
  plot(y,'r'); drawnow; hold off;

ajusta=['set(sbmfile,''color'',''w'',''xcolor'',''k'',''ycolor'',''k''); '...
'set(sbmfile,''fontsize'',10,''ygrid'',''on'',''xgrid'',''on''); '...
'title(''Dados de Entrada e Sada do processo (AZUL=Entrada,
VERM.=Sada,)'',''color'',''k''); '...
'xlabel(''Nmero de pontos'',''color'',''b'');ylabel(''Amplitude'',''color'',''b''); '...
'yt=get(gca,''ytick''); xt=get(gca,''xtick''); text(xt(2),yt(2)-((yt(2)-
yt(1))/5),fname,''color'',''m''); '];

eval(ajusta);
xt(2),   % diagnostico
yt(2)-(yt(1)-yt(2)/5),% diagnostico

global sbmfile;
```

```
%limpamemoria
clear fname pname

mprinc;
estimou=0;
%___________fim do programa_____________________________

% mexper.m      (chamado por: mprocess.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

mlimpa;
labels = str2mat(...
   '1. EDITE O SINAL DE EXCITAO', ...
   '2. EDITE A F.DE TRANSFERNCIA', ...
   'RETORNA');
callbacks = [ ' msinal   '
      ' mentra   '
         ' mprocess ' ];
pos = [150 180];
escolha('SIMULATION','Execute os itens na sequncia abaixo :',...
 labels, callbacks,pos,'SIMULATION - mexper.m');

%limpamemoria
clear labels callbacks pos
%___________fim do programa_____________________________

% tipfilt.m      (chamado por: filtros.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%  por WASHINGTON SILVA - janeiro, 24, 1996.
% Escolha do tipo de filtro que sera utilizado na filtragem dos dados para a estimao

function tipfilt(tipo,order,freqcorte,ripple,Rp,Rs,faixa),

global A B,
%_________________Butterworth_____________________%
 if tipo==1,
   if faixa==0,
     [B,A] = butter(order,freqcorte);  % filtro passa-baixas ou passa-faixa
   else
     [B,A] = butter(order,freqcorte,'stop'); % filtro passa-altas ou rejeita-faixa
   end;
 end;
%_________________Chebychev_____________________%
 if tipo==2,
  [B,A] = cheby1(order,ripple,freqcorte);
 end;
%_________________Bessel_____________________%
 if tipo==3,
  [B,A] = besself(order,freqcorte);
 end;
```

```
%_________________Elptico_____________________%
 if tipo==4,
  [B,A] = ellip(order,Rp,Rs,freqcorte);
 end;

% The cut-off frequency Wn must be  0.0 < Wn < 1.0, with 1.0 corresponding
% to half the sample rate.

%____________fim do programa_____________________________

% metestim.m      (chamado por: mestim.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.
%____ESTIMAO MINIMOS QUADRADOS NO-RECURSIVO__________
%____ESTIMAO VARIVEIS INSTRUMENTAIS NO-RECURSIVO__________
% parametros de entrada -> ordem do modelo (ordem), metodo desejado (flagmet).
% parametros de saida  -> erro mdio quadratico (emq).

function [emq] = metestim(ordem,flagmet)

global y u ordem
global utot TETA estimou
estimou=1;  %__Uma vez que passou por aqui o flag ir preservar o ltimo TETA estimado

Np=size(u);
R=zeros(2*ordem,2*ordem); F=zeros(2*ordem,1); Riv=R; Fiv=F;
Np=max(Np); % Este artifcio foi incluido para permitir vetores
% linha como dado. Surgiu com os dados do secador.
%     Calculo do vetor de parametros estimado Teta   %
for Tp = ordem+1:Np;
    for t=1:ordem,
FI2(t)=-y(Tp-t);
        FI2(t+ordem)=u(Tp-t);
    end;
     FI=FI2';
     R = R + FI*FI';
     F = F + FI*y(Tp);
end;
  RINV = inv(R);
  TETA = RINV*F;
  Aest(1) = 1;
  Aest(2:ordem+1) = TETA(1:ordem);
  Best(2:ordem+1) = TETA(ordem+1:ordem*2);
%  Aest(1)=1; for t=1:ordem, Aest(t+1)=TETA(t);  end;
%  Best(1)=0; for t=1:ordem, Best(t+1)=TETA(t+ordem);  end;
  yest = dlsim(Best,Aest,u);

if flagmet==2;
  for Tp = ordem+1:Np;
    for t=1:ordem, FI2(t)=-y(Tp-t);  FI2(t+ordem)=u(Tp-t);  end; %<<<
    for t=1:ordem, QS2(t)=-yest(Tp-t);  QS2(t+ordem)=u(Tp-t);  end; %<<<
    FI=FI2';
    QS=QS2';
```

```
    Riv=Riv+QS*FI';
    Fiv=Fiv+QS*y(Tp);
  end
  TTiv = inv(Riv)*Fiv;
%  hold off
  TETA=TTiv;
end;


%limpamemoria
clear Np R F Riv Fiv Tp t FI2 RINV flagmet Aest Best yest yestiv QS2 QS INViv TTiv
%------------fim do programa---------------------------------

% mostima.m      (chamado por: mestim.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

 global u TETA fig ordem den num
 figure(fig);
 clear den num

%----------------Apagar boto de polos do sistema, para estimar novos polos.---------
if exist('Btpl'), delete(Btpl); end;

%------Capturar os valores dos parametros estimados---------------------------------
 den(1)=1;
 den(2:ordem+1)=TETA(1:ordem);
 num(2:ordem+1)=TETA(ordem+1:ordem*2);
 strden=mat2str(den,4);
 strnum=mat2str(num,4);

%------Mostrar na janela numerador e denominador estimados--------------------------
 if exist('enum'), clear enum; end;
 if exist('eden'), clear eden; end;
 enum=uicontrol('Style','edit','horiz','left','string',strnum,'units','norm','position',...
[.01 .31 .95 .04],'backg','w','foreg','k');
 eden=uicontrol('Style','edit','horiz','left','string',strden,'units','norm','position',...
[.01 .21 .95 .04],'backg','w','foreg','k');
 set(ordhndl,'value',ordem);   %-----repoe o valor anterior da ordem do modelo.-----------

%-Validao do modelo estimado - Compara o dado de sada com a curva de resposta do
modelo-

  Np=length(y);
  yest = dlsim(num,den,u);

   VRLS = 0;   VZHANG = 0;
   for i=10:Np;
     VRLS = VRLS + 0.5*(y(i) - yest(i))^2;
   end
 format short e;
 strval=mat2str(VRLS/Np,7);
```

```
 val=uicontrol('Style','edit','horiz','left','string',strval,'units','norm','position',...
[.88 .43 .11 .04],'backg','w','foreg','k');
 uicontrol('Style','text','position',[.73 .43 .14 .04],'string','Erro Estimao:',...
'backg','w','foreg','k','horiz','left','units','norm');
 format short;

%-------Leitura da condio inicial que ser utilizada-------------------------------------
if ~exist('Xmult'), Xmult=1; end;
xm1=['vartemp=Xmult; Xmult=edados(xm2);  if isempty(Xmult), Xmult=vartemp; end; '];
xm2=uicontrol('Style','edit','string',['  ' mat2str(Xmult,10)],'callback',xm1,...
'position',[.927 .65 .06 .045],'backg','y','units','norm');

 respcond=['[a,b,c,d]=tf2ss(num,den); '...
'X0=zeros(ordem,1); Np=size(u); Np=max(Np); '...
'[y2,x]=dlsim(a,b,c,d,u,X0); '...
'[y2,x]=dlsim(a,b,c,d,u,Xmult*x(Np-Np/6,:)); '...
'figure(''color'',''w'');ax=axes(''xcolor'',''k'',''ycolor'',''k''); '...
'hold on; plot(u,''b''); plot(y2,''k''); plot(y,''r''); grid;         '...
'set(ax,''fontsize'',8,''ygrid'',''on'',''xgrid'',''on''),'...
'title('' Resposta do Modelo Estimado ao Sinal Usado na Entrada do processo
(AZUL'...
'=Entrada;VERM.=SadaPlanta;PRETO=SadaModelo)''); '...
'uicontrol(''Style'',''push'',''position'',[.85 .01 .11 .06],''string'',''Fecha'',' ...
' ''callback'',''close'',''units'',''norm''); '];

%________calculo do diagrama de
BODE_____________________________________________
 diabode=['figure; num3=num(2:ordem+1); dbode(num3,den,1);  '...
'uicontrol(''Style'',''push'',''position'',[.85 .01 .11 .06],''string'',''...
' ''Fecha'',''callback'',''close'',''units'',''norm'');'];

%________calculo do diagrama de
nyquist__________________________________________
 dianyq=['figure; num3=num(2:ordem+1); dnyquist(num3,den,0.5);   '...
'uicontrol(''Style'',''push'',''position'',[.85 .01 .11 .06],''string'',''...
' ''Fecha'',''callback'',''close'',''units'',''norm'');'];

%________calculo dos polos do modelo
estimado_________________________________
PCalc=0; %-----inicializa variavel.
 calcpolos=['vp=[]; ra=roots(den); sa=size(ra);  '...
       'for t=1:sa(1), rastr=mat2str(ra(t),4); vp=[vp rastr ''|'']; end; '...
'vp=[''Plos Discretos | '' vp]; set(Btpl,''style'',''popup'',''string'',vp); '];

%__________Executar um comando do MATLAB dentro da janela de
Estimacao_____________
 command=['  vlr=get(Comm,''string'');  eval(vlr);'];
 uicontrol('Style','edit','string',' Executar o comando do MATLAB : ','position',...
[.04 .14 .35 .04],'backg','w','foreg',[1 0 1],'units','norm');
 Comm=uicontrol('Style','edit','position',[.04 .1 .35 .04],'backg','w',...
'foreg',[1 0 0],'units','norm','callback',command);

%________Define Botoes para executar diversos comandos apos estimacao do
modelo______
uicontrol('Style','push','position',[.77 .80 .22 .045],'string','Diagr. BODE',...
```

```
'foreg','w','callback',diabode,'clipping','off','units','norm');
uicontrol('Style','push','position',[.77 .75 .22 .045],'string','Diagr. NYQUIST',...
'foreg','w','callback',dianyq,'clipping','off','units','norm');
uicontrol('Style','push','position',[.77 .70 .22 .045],'string','Resp.do Modelo',...
'foreg','w','callback',respcond,'clipping','off','units','norm');
uicontrol('Style','text','position',[.77 .65 .155 .045],'string',' Condio Inicial:',...
'backg','w','foreg','k','callback',respcond,'horiz','left','units','norm');

Btpl=uicontrol('Style','push','position',[.77 .50 .22 .045],'string','Plos Discretos',...
'backg','w','callback',calcpolos,'units','norm');

eval(calcpolos); %-- Modificar itens para criar o botao Btpl como popup de imediato.
%  observar os demais modulos que utilizam a variavel Btpl.

%___________Define boto para ZOOM do
grafico___________________________________
global ZO ;
C_zoom = ['if ZO, set(uiz,''string'',''Congela''); ZO=0; set(gcf,''pointer'',''crosshair''); '...
  ' else set(uiz,''string'',''Amplia''); ZO=1; set(gcf,''pointer'',''arrow''); end; zoom '];
uiz=uicontrol(gcf,'style', 'push','units','normal','position', [.77 .55 .22 .045],'string',...
 'Amplia','callback','eval(C_zoom)');
set(gcf,'units','pixels');
ZO=1;
%_____________________limpamemoria_____________________________________
----
clear strden strnum respmol leXmult ax y2 x vlr ehl XO num3 ra
clear diabode dianyq calcpolos salvarq armzarq str fname pname sb ai

%___________fim do programa_____________________________

function h = hankel2(c,nc)
% Matriz HANKEL

for j=1:nc
  h(j,1:nc) = c(j:nc+j-1);
end

% msinal.m      (chamado por: mexper.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.
%
% Construo da Forma de Onda para Excitao da Planta

global u Pi  Pf  Ta  amp  per
global  ePi ePf eTa eamp eper
global tamtot uq ut us ud,

mlimpa; figure;
set(gcf,'units','normalized','position',[.1 .1 .68 .68],'name',...
  'Edio do Sinal de Excitao - Msinal.m','number','off','units','norm',...
'menubar','figure','color',clr);

%___________________HELP DA
```

```
ROTINA_____________________________________________
 uimenu(gcf,'label','Ajuda','callback','hmsinal');

%_____rotina para o funcionamento da forma exclusiva dos radios buttons _____
 z1 = [ 'set(r,''value'',0);' 'set(r(1),''value'',1);' 'forma=1;'];
 z2 = [ 'set(r,''value'',0);' 'set(r(2),''value'',1);' 'forma=2;'];
 z3 = [ 'set(r,''value'',0);' 'set(r(3),''value'',1);' 'forma=3;'];
 z4 = [ 'set(r,''value'',0);' 'set(r(4),''value'',1);' 'forma=4;'];

%___________Radio
Buttons_____________________________________________
 r(1)=uicontrol('Style','radio','units','norm','string',' Onda Quadrada ',...
'position',[.05 .21 .23 .042],'CallBack',z1,'backg',clr);
 r(2)=uicontrol('Style','radio','units','norm','string',' Dente-de-serra ',...
'position',[.05 .17 .23 .042],'CallBack',z2,'backg',clr);
 r(3)=uicontrol('Style','radio','units','norm','string',' Onda Senoidal ',...
'position',[.05 .13 .23 .042],'CallBack',z3,'backg',clr);
 r(4)=uicontrol('Style','radio','units','norm','string',' Degrau ',...
'position',[.05 .09 .23 .042],'CallBack',z4,'backg',clr);

%___________Inicializao dos parmetros da Onda_____________________________
  forma=1; set(r(1),'value',1); Ta=0.05; Pi=0; Pf=250; amp=1.501; per=3;

entnum=['Pi=edados(ePi);'];
 entPf=['Pf=edados(ePf);Ta=(2*pi*per)/(Pf-Pi); set(eTa,''string'',mat2str(Ta,4))'];
 entTa=['Ta=edados(eTa);Pf=(2*pi*per/Ta)+Pi;set(ePf,''string'',mat2str(Pf,4))'];
entAmp=['amp= edados(eamp);'];
entper=['per= edados(eper);'];

%-------------------------------------------------------------------------
 uicontrol('Style','text','string',' Ponto inicial :','position',...
[.43 .295 .255 .04],'backg','w','units','normal');
 uicontrol('Style','text','string',' Ponto final :','position',...
[.43 .253 .255 .04],'backg','w','units','normal');
 uicontrol('Style','text','string',' Tempo de Amostragem :','position',...
[.43 .211 .255 .04],'backg','w','units','normal');
 uicontrol('Style','text','string',' Amplitude do Sinal :','position',...
[.43 .169 .255 .04],'backg','w','units','normal');
 uicontrol('Style','text','string',' Perodo do Sinal :','position',...
[.43 .127 .255 .04],'backg','w','units','normal');
 uicontrol('Style','text','string',' Guardar a Curva Editada em :','position',...
[.4 .06 .305 .04],'backg','w','units','norm');
 uicontrol('Style','text','string',' Ler a Curva Editada em :','position',...
[.4 .01 .305 .04],'backg','w','units','norm');
%-------------------------------------------------------------------------
 ePi=uicontrol('Style','edit','string',['  ' mat2str(Pi,8)],'callback',...
entnum,'position',[.7 .295 .11 .04],'backg','w','units','norm');
 ePf=uicontrol('Style','edit','string',['  ' mat2str(Pf,8)],'callback',...
entPf,'position',[.7 .253 .11 .04],'backg','w','units','norm');
 eTa=uicontrol('Style','edit','string',['  ' mat2str(Ta,8)],'callback',...
entTa,'position',[.7 .211 .11 .04],'backg','w','units','norm');
 eamp=uicontrol('Style','edit','string',['  ' mat2str(amp,8)],'callback',...
entAmp,'position',[.7 .169 .11 .04],'backg','w','units','norm');
 eper=uicontrol('Style','edit','string',['  ' mat2str(per,8)],'callback',...
entper,'position',[.7 .127 .11 .04],'backg','w','units','norm');
```

```
%_________________BOTES DE OPERACAO DA
CURVA_____________________________
 global flag; flag=0;
 apaguecurva=['clear global u uq ut us ud ' 'axis off,cla,'];
 tracecurva=['axis off; cla; clear u; monda(forma,amp,Ta,Pi,Pf,per); flag=1;'];
 somafecha=['clear r1 r2 r3 r4 r5 Ta Pi Pf amp per forma uicurva curvNb '...
' ud ut us uq tamtot, clear entTa entnum entPf entamp entper tracecurva,'...
  'clear z1 z2 z3 z4 z5 vlr letra le somafecha cbkstr1 cbkstr2 sb,'...
'clear eTa ePi ePf eamp eper apaguecurva,' 'mexper, '];

 cbkstr1=['CNb=get(uicurva1,''value''), grasiex(CNb); '];
 cbkstr2=['eval(apaguecurva), CNb=get(uicurva2,''value''), lesiex(CNb); '];

 uicurva1=uicontrol('Style','popup','units','norm','position',[.72 .06 .12 .04],...
'backg','w','string','Curva 1|Curva 2|Curva 3|Curva 4|Curva 5','callback',cbkstr1);
 uicurva2=uicontrol('Style','popup','units','norm','position',[.72 .01 .12 .04],...
'backg','w','string','Curva 1|Curva 2|Curva 3|Curva 4|Curva 5','callback',cbkstr2);

 uicontrol('Style','push','units','norm','position',[.87 .16 .1 .045],'string',...
'Trace','callback',tracecurva);
 uicontrol('Style','push','units','norm','position',[.87 .11 .1 .045],'string',...
'Apaga','callback',apaguecurva);
 uicontrol('Style','push','units','norm','position',[.87 .06 .1 .045],'string',...
'Soma','callback','somonda');
 uicontrol('Style','push','units','norm','position',[.87 .01 .1 .045],'string',...
'Fecha','callback',somafecha);

%_________Plotando a curva "DEFAULT"_______________
cla, axis off, monda(forma,amp,Ta,Pi,Pf,per),flag=1;
global Z0;
C_zoom = ['if Z0, set(uiz,''string'',''Congela''); Z0=0; set(gcf,''pointer'',''crosshair''); '...
  ' else set(uiz,''string'',''Amplia''); Z0=1; set(gcf,''pointer'',''arrow''); end; zoom '];
uiz=uicontrol(gcf,'style', 'push','units','normal','position', [.85 .21 .14 .06],...
'string', 'Amplia','callback','eval(C_zoom)');
set(gcf,'units','pixels');
Z0=1;

%____________fim do programa_____________________________

% mentra.m      (chamado por: mexper.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

global Ta; mlimpa;  figure;
set(gcf,'name','Edio da Funo de Transferncia da Planta (no domnio Z) - mentra.m ',...
'number','off','menubar','none','color',[0.85 0.85 0.85],'units','norm',...
'position',[.1 .1 .68 .68]);
%____Valores iniciais da planta (s inicializa na primeira vez que executar a rotina)

estimou=0;   %_______Reinicializa a variavel que indica estimao.
if ~exist('mentrei'),  %_____Banco de plantas para
processamento_____________________
```

```
 % num2=[0 0.0195 0.0168]; den2=[1 -1.6031 0.6394];  % Caixa Preta Discretizada
(Ts=0.5seg)
 % num2=[0 0.1 0.05]; den2=[1 -1.5 0.7]; %...FT no domnio Z, usada por ZHANG no artigo
 % num2=[0 0.1084 0.5918]; den2=[1 -0.3165 0.0340]; %Planta P2 - no dom. tempo
 % num2=[0 0.011 0.011]; den2=[1 -1.9008 0.9030];  Ta=0.1141 %..Planta P2 - discretizada
com T=0.1141
  num2=[0 0.0018 0.0018]; den2=[1 -1.8671 0.8707]; Ta=0.1;
end;
mentrei=0;  %..Esta variavel faz com que "num2" e "den2" no sejam reinicializados de novo..

 strden=mat2str(den2,4);
 strnum=mat2str(num2,4);

 uicontrol('Style','text','string','Numerador da Planta : B(q)','position',[.03 .18 .7 .04],...
'backgroundcolor','w','HorizontalAlignment','left','units','norm');
 uicontrol('Style','text','string','Denominador da Planta : A(q)','position',[.03 .07 .7 .04],...
'backgroundcolor','w','HorizontalAlignment','left','units','norm');

mostra=[' msimu(num2,den2,mult); cla, axis off, sb=subplot(''position'',[0.07 .35 .85 .55]);'...
'ca=plot(y,''r''); hold; cb=plot(u,''b''); '...
'set(sb,''color'',''w'',''xcolor'',''k'',''ycolor'',''k'',''fontsize'',8);'...
'set(sb,''ygrid'',''on'',''xgrid'',''on'');'...
'title('' Dados de entrada e sada do processo
(AZUL=Entrada;VERM.=Sada)'',''color'',''k''); '];

entden=['vlr=get(eden,''string''); '...
'if any(isletter(vlr)),eval(erro); end, '...
'den2=eval(vlr);          '...
' '];  %'eval(mostra); '];

enum=uicontrol('Style','edit','string',strnum,'callback','numden',...
'units','norm','position',[.03 .138 .7 .04],'backgroundcolor','w');
eden=uicontrol('Style','edit','string',strden,'callback',entden,...
'units','norm','position',[.03 .028 .7 .04],'backgroundcolor','w');

global u y ;
salva=['msave;']; % reset(ca), reset(cb),
apaga=[' cla, axis off, sb=subplot(''position'',[0.07 .35 .85 .55]); '...
'set(sb,''color'',''w'',''xcolor'',''k'',''ycolor'',''k'',''fontsize'',8);'...
'set(sb,''ygrid'',''on'',''xgrid'',''on'');'...
'title('' Dados de entrada e sada do processo
(AZUL=Entrada;VERM.=Sada)'',''color'',''k'');'];

saida=['clear saida ca cb, ' 'mprinc'];

mult=0.1;
entmult = ['mult=edados(emult); '];
emult=uicontrol('Style','edit','string',['  ' mat2str(mult,4)],'callback',entmult,...
'position',[.34 .232 .10 .04],'backg','w','units','norm');

uicontrol('Style','push','units','normalized','position',...
[.03 .230 .30 .05],'string','Multiplicador do Rudo');
uicontrol('Style','push','units','normalized','position',...
[.8 .179 .17 .05],'string','SalvaDados','callback',salva);
uicontrol('Style','push','units','normalized','position',...
```

```
[.8 .126 .17 .05],'string','Apaga Curva','callback',apaga);
uicontrol('Style','push','units','normalized','position',...
[.8 .073 .17 .05],'string',' RESPOSTA ','callback',mostra);
uicontrol('Style','push','units','normalized','position',...
[.8 .020 .17 .05],'string','   Fecha  ','callback',saida);

sb=subplot('position',[0.07 .35 .85 .55]);
set(sb,'color','w','xcolor','k','ycolor','k','fontsize',8);
set(sb,'ygrid','on','xgrid','on');

%__________limpamemoria_____________________
clear strden strnum entden vlr salva apaga sb

global ZO ;
C_zoom=['if ZO,set(uiz,''string'',''Congela'');ZO=0;else
set(uiz,''string'',''Amplia'');ZO=1;end;zoom'];
uiz=uicontrol(gcf,'style', 'push','units','normal','position',...
  [.82 .23 .14 .06],'string', 'Amplia','callback','eval(C_zoom)');
set(gcf,'units','pixels');
ZO=1;
eval(mostra);

figure(1);
  if exist('sbmfile'), sbmfile; gca; axes(sbmfile); cla; axis off; end;
  sbmfile=subplot('position',[.06 .15 .9 .7]);
  plot(u,'b'); drawnow; hold on;
  plot(y,'r'); drawnow; hold off;

ajusta=['set(sbmfile,''color'',''w'',''xcolor'',''k'',''ycolor'',''k''); '...
'set(sbmfile,''fontsize'',10,''ygrid'',''on'',''xgrid'',''on''); '...
'title(''Dados de Entrada e Sada do processo (AZUL=Entrada,
VERM.=Sada,)'',''color'',''k''); '...
'xlabel(''Nmero de pontos'',''color'',''b'');ylabel(''Amplitude'',''color'',''b''); '];
eval(ajusta);

figure(2);
%____________fim do programa_____________________________

% somonda.m      (chamado por: monda.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.
% SOMONDA Soma sinais construidos separadamente p/ compor o sinal de excitao.

if flag,
 global tamtot uq ut us ud u,
 u=zeros(1,max(tamtot));
 %-------------------------------------------------------------
   if exist('uq'),  %_________COMPONENTE SINAL QUADRADO________
     cont=1;
     for T=1:tamtot(1),
       u(cont)=u(cont)+uq(cont);    cont=cont+1;
     end;
   end;
```

```
%_____________________________________________________________
  if exist('ut'),   %_________COMPONENTE SINAL DENTE-DE-SERRA
    cont=1;
    for T=1:tamtot(2),
      u(cont)=u(cont)+ut(cont);    cont=cont+1;
    end,
  end;
%_____________________________________________________________
  if exist('us'),    %_________COMPONENTE SINAL SENOIDAL______
    cont=1;
    for T=1:tamtot(3),
      u(cont)=u(cont)+us(cont);   cont=cont+1;
    end;
  end;
%_____________________________________________________________
  if exist('ud'),    %_________COMPONENTE SINAL DEGRAU________
    cont=1;
    for T=1:tamtot(4),
      u(cont)=u(cont)+ud(cont);  cont=cont+1;
    end;
  end;
%_____________________________________________________________
  cla; axis off;
  sb=subplot('position',[.1 .44 .8 .5]); plot(u,'r');
  set(sb,'color','w');  set(sb,'xcolor','k','ycolor','k');
  set(sb,'fontsize',8,'ygrid','on','xgrid','on'),
  title(' Sinal de Excitao','color','k');
end;

%____________________limpamemoria_______________________________
clear sb flag cont T

%____________fim do programa_____________________________

% monda.m      (chamado por: msinal.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

function monda(forma,amp,Ta,Pi,Pf,per),

global uq ut us ud u;
global tamuq tamut tamus tamud tamtot;

sb=subplot('position',[.1 .44 .8 .5]);
%_________________Onda Quadrada_____________________%
if forma==1,
   if exist('uq'), uq=[]; end;
  cont=1;
 for T=0:Ta:2*pi*per-Ta,
  uq(cont+Pi)=amp*square(T);
  cont=cont+1;
 end;
  plot(uq,'b');   tamuq=size(uq);  u=uq;
```

```
end;
%_________________Onda Triangular_____________________%
if forma==2,
   if exist('ut'), ut=[]; end;
  cont=1;
 for T=0:Ta:2*pi*per-Ta,
  ut(cont+Pi)=amp*sawtooth(T);
  cont=cont+1;
 end;
  plot(ut,'b'); tamut=size(ut);  u=ut;
end;
%_________________Onda Senoidal_____________________%
if forma==3,
 if exist('us'), us=[]; end;
 cont=1;
 for T=0:Ta:2*pi*per-Ta,
  us(cont+Pi)=amp*sin(T);
  cont=cont+1;
 end;
plot(us,'b'); tamus=size(us);  u=us;
end;
%_________________Degrau_____________________%
if forma==4,
 if exist('ud'), ud=[]; end;
 cont=1;
  for T=0:Ta:Pf,
   ud(cont+Pi)=amp;   cont=cont+1;
  end;
 plot(ud,'b');  tamud=size(ud);  u=ud;
 end;

%_____________________________________________%

 if exist('tamuq'), tamtot(1)=tamuq(2); end;
 if exist('tamut'), tamtot(2)=tamut(2); end;
 if exist('tamus'), tamtot(3)=tamus(2); end;
 if exist('tamud'), tamtot(4)=tamud(2); end;

%_____Inicializa o vetor com sinal de excitao_______

 set(sb,'color','w'); set(sb,'xcolor',[0 0 0],'ycolor','k');
 set(sb,'fontsize',8,'ygrid','on','xgrid','on'),
 title(' Sinal de Excitao ','color','k');
 xlabel(' Nmero de pontos ','color','k');
 ylabel(' Amplitude ','color','k');
%____________fim do programa_____________________________

% numden.m      (chamado por: mentra.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
% NUMDEN gerencia a entrada de dados para edio da funo de
% transferncia da planta que ser simulada.
% chama msimu.m
```

```
vlr=get(enum,'string'); letra=any(isletter(vlr));
if ~letra,
  pos=findstr(vlr,';');  pnf=any(pos);
  if pnf,
    d=size(pos);
    for k=1:d(1,2),
      if k==1,
        vlr2=vlr(1:pos(1,k)-1);
      else,
        vlr2=vlr(pos(1,k-1)+1:pos(1,k)-1);
      end;
      vlr2(1,1)='[';  cp=size(vlr2);  vlr2(cp(1,2)+1)=']';
      vs=size(eval(vlr2)); teste(k)=vs(1,2);
      end;
    w=teste-teste(1,1);
   clear z pos d vs teste cp k ;
    z=any(w),
  else,
    num2=eval(vlr);
%    msimu(num2,den2);
  end;
  if ~z & pnf,
    num2=eval(vlr2);
%    msimu(num2,den2);
  end;
end;

%limpamemoria
clear vlr letra pos pnf d k vrl2 cp vs w z

%____________fim do programa_____________________________

% msimu.m      (chamado por: numden.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

%  Programa de simulao dos dados (planta)
% (a alterao da amplitude do ruido esta disponivel para o usuario )
% (os polinomios/equaes esto no dominio discreto - Z )
% B=num, A=den e rd=multiplicador do ruido.

function msimu(B,A,rd);

global u y;
Np=size(u);
R=zeros(4,4); %___Definio dos parmetros utilizados_____
F=zeros(4,1);
ye = zeros(Np(2),1);  yu = zeros(Np(2),1);  y = zeros(Np(2),1);

Be = [0 0 1]; %___Numerador do rudo_____________________
uv = 0.02*rd*randn(Np(2),1);
Bv = [0 0.03 0.05]; Av = [1 -1.7 0.72];
ue = dlsim(Bv,Av,uv);
```

```
%_______Simulao do sistema___________________
yu = dlsim(B,A,u); ye = dlsim(Be,A,ue);
for Tp = 1:Np(2);
  y(Tp) = yu(Tp) + ye(Tp);
end

%limpamemoria
clear Np R F ye yu y Be uv Av Bv ue Tp

%____________fim do programa_____________________________

function escolha(name,header,labels,callbacks,pos,header2)
%ESCOLHA  uma verso modificada da funo CHOICES, para atender
%         necessidades especficas de uma certa interface.
%        Para maiores informaes consulte a funo CHOICES.

% Tipo computador onde o programa esta sendo executado.
c = computer;

% Verifica se os argumentos da funo so strings.
if ~isstr(name) | ~isstr(header) | ~isstr(labels) | ~isstr(callbacks)
error('Requires string arguments.');
end
if nargin < 4
error('Not enough input arguments.')
end

% De acordo com o tipo de computador, o programa age de modo diferente.
uicok = strcmp(c(1:2),'PC') | strcmp(c(1:2),'MA');
if isunix | ~uicok   % could be VMS as well
   uicok = strcmp(lower(get(0,'TerminalProtocol')),'x');
end

%can't use uicontrols -use menu stuff instead- this is for terminals -UNIX & VMS
 if ~uicok
   labels = str2mat(labels,'Done');
   nl = size(labels,1);
   % build up menu string for evaluation
   % fix quotes, if there are any
   ss = deblank(labels(1,:));
   ss = ss(sort([1:length(ss) find(ss=='''')]));
   args = ['''',ss,''''];
   header = header(sort([1:length(header) find(header=='''')]));
   for i = 2:nl
      ss = deblank(labels(i,:));
      ss = ss(sort([1:length(ss) find(ss=='''')]));
      args = [args, ',''', ss,''''];
   end
   k = 1;

   while k > 0 & k < nl
% WMENU  uma verso modificada da funo menu.
      eval(['k = wmenu(''',header,''',', args,');']);
```

```
      if k == nl | k == 0
         return
      else
        eval(callbacks(k,:));
      end
   end
end

% can use uicontrols
global CHOICELIST
global CHOICEHANDLES
name = deblank(name);
if isempty(name)
error('Requires non-blank string argument.')
end

% ensure list doesn't go into figure 1
figs = sort(get(0,'Children'));
openfigs = size(figs);
if ~isempty(figs)
   cf = gcf;
   if cf == 1
      cf = [];
   end
else
   cf = [];
end

fig1 = 1;
if isempty(figs)
figs = figure('visible','off');
        fig1 = 0;
end
if figs(1) ~= 1
figs = [figure('visible','off'); figs];
        fig1 = 0;
end

matchn = 0;
for i = 1:size(CHOICELIST,1)
if strcmp(name,deblank(CHOICELIST(i,:)))
    matchn = i;
    break;
end
end

if ~matchn
CHOICEHANDLES = [CHOICEHANDLES; 0];
if isempty(CHOICELIST)
    CHOICELIST = name;
else
    CHOICELIST = str2mat(CHOICELIST, name);
end
matchn = size(CHOICEHANDLES,1);
else
```

```
matchh = 0;
for i = 1:size(figs,1)
    if figs(i) == CHOICEHANDLES(matchn)
        matchh = i;
        break;
    end
end

if matchh
%     figure(CHOICEHANDLES(matchn));
    return
end
end
xedge = 20;    % Distancia da borda para os botoes
ybord = pos(2);
width = 180;   % Largura da janela
%yedge = 250;
%height = 20;

if c(1:2) == 'PC'
      ha = axes('visible','off');
      hh = text(.1,.1,'Aq','units','pixel');yedge = get(hh,'extent');
      delete(hh);delete(ha);
      yedge = 1.8*yedge(4);   % 1.8
      height = 0.93*yedge;  % 0.95
end

avwidth = 5; % relacionado a largura da janela
imax = 1;    % ?
maxlen = size(labels,2);
twidth = 1.1*maxlen*avwidth; % relacionado a largura da janela

% now figure out total dimensions needed so things can get placed in pixels
mwwidth = twidth + width + 2*xedge;
mwheight = (size(labels,1)+2)*yedge;
ss = get(0,'ScreenSize');
swidth = ss(3); sheight = ss(4);
left = pos(1)+60;
bottom = sheight-mwheight-ybord;

rect = [left bottom mwwidth mwheight];

CHOICEHANDLES(matchn) = figure('Position',rect,'number','off', ...
       'name',header2,'resize','off','colormap',[],'NextPlot','new',...
       'Menubar','none');
fg = CHOICEHANDLES(matchn);
fgs = CHOICEHANDLES(CHOICEHANDLES ~= fg);
%set(fgs,'visible','off')
set(gca,'Position',[0 0 1 1]); axis off;

% Cria o ttulo do menu (cabealho).
hh=uicontrol(fg,'style','text','units','normal',...
             'position',[.05,.90,1.3,.09 ],'string',header,...
            'Horizontal','center');
set(hh,'backg',get(gcf,'color'),'foreg',[1 1 1]-get(gcf,'color'))
```

```
% set up pre-amble so figure 1 is available for rendering in
sb = ['figure(1),set(1,''visible'',''on'');set(',int2str(fg),',''visible'',''off'');'];
se = [';global
CHOICEHANDLES,set(CHOICEHANDLES(length(CHOICEHANDLES)),''visible'',''on'')'];

for ii=size(labels,1):-1:1
i = size(labels,1) + 2 - ii;
        h1 = uicontrol(fg,'position',[xedge  (i-.6)*yedge width+twidth height]);

% cria uma ligao com tarefas posteriores.
set(h1,'callback',[sb callbacks(ii,:) se])

% d nomes s strings das escolhas
        set(h1,'string',['  ',deblank(labels(ii,:)), '  ']);

% left justify string inside button
% set(h1,'HorizontalAlignment','left');

end

% Cria o boto de sada deste menu.
h1 = uicontrol(fg,'position',[xedge yedge-25 width+twidth height],'string','ABANDONA');
if gcf < 2
    set(h1,'callback','close; figure(gcf)');
else
    set(h1,'callback','close');
end

if ~isempty(cf)
   figure(cf)

end


% mlimpa.m      (chamado por: isac.m)
% TOOLBOX DE IDENTIFICAO COM REDUO DA ORDEM DO MODELO USANDO
SVD
%
%  por WASHINGTON SILVA - november,1st,1995.

function mlimpa

child=get(0,'children');
ord=length(child); %size(child)
 for l=ord:-1:2;   %ord(1)
   close(l);
 end;
%__limpamemoria___
clear ord child l
%____________fim do programa_____________________________

% edados.m
function [var] = edados(rot)




      else,
var=0; set(rot,'string','0');
```