

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA  
DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO E  
AUTOMAÇÃO INDUSTRIAL  
MAIO 1988

*Este exemplar corresponde à red  
final da tese defendida por  
L. Valéria R. Arruda e aprovada  
pela Comissão Julgadora em 20/05/88*  
@Amaral

UM SUPERVISOR BASEADO EM CONHECIMENTO  
PARA MODELAGEM DE PROCESSOS

Por: Lúcia Valéria Ramos de Arruda  
Orientador: Prof. Wagner C. Amaral

Tese apresentada à Faculdade de  
Engenharia Elétrica FEE - UNICAMP  
como parte dos requisitos exigidos  
para a obtenção do título de MESTRE  
EM ENGENHARIA.

Campinas - 1988

UNICAMP  
BIBLIOTECA CENTRAL

Este trabalho contou com o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq e da SID Informática.

## AGRADECIMENTOS

Ao prof. Wagner Amaral, não só pela orientação segura e atenciosa mas pelo interesse e oportunidades oferecidos.

Aos pesquisadores da divisão MAP do Centro Tecnológico para Informática e a todas as pessoas que direta ou indiretamente contribuíram para a realização deste trabalho.

Ao colega Antonio Coelho pelas discussões e sugestões apresentadas.

Ao amigo João César que sempre foi um incentivo e exemplo.

À Helga, um agradecimento especial.

À Kátia e Keiko, pela amizade.

À minha família, pelo apoio e segurança.

## RESUMO

Este trabalho apresenta um sistema baseado em conhecimento para modelagem de processos que auxilia os usuários na utilização e regulagem de algoritmos de estimação, e na modelagem de processos variantes no tempo. O sistema desenvolvido é composto dos seguintes módulos: base de dados, base de conhecimentos, módulo de inferência e explicação, módulo supervisor, módulo de identificação e interface homem-máquina.

A base de conhecimentos é formada por regras de produção do tipo SE (condição) ENTÃO (conclusão), que são estabelecidas a partir de considerações sobre os valores de determinadas variáveis do algoritmo de identificação.

O módulo de identificação é composto de dois algoritmos capazes de detetar e rastrear variações nos parâmetros do processo: o algoritmo do traço adaptativo e o filtro de Kalman adaptativo. Estes algoritmos são obtidos a partir de um algoritmo de identificação recursivo, mínimos quadrados com traço limitado ou filtro de Kalman, e um método de deteção de não-estacionaridades baseado ou no erro previsto ou nos parâmetros estimados.

## ÍNDICE

### CAPÍTULO 1 - Introdução

1.1 - Objetivos do trabalho .....	01
1.2 - Sistemas especialistas: definição e características ...	05
1.3 - Representação do conhecimento.....	07
1.3.1 - Métodos baseados em frames .....	08
1.3.2 - Métodos baseados em regras .....	09
1.4 - Sistemas de produção .....	11
1.4.1 - Representação dos dados (Base de dados) .....	11
1.4.2 - Representação do conhecimento .....	12
1.4.3 - Máquina de inferência .....	13
1.4.3.1 - Sistemas de produção com encadeamento para frente .....	13
1.4.3.2 - Sistemas de produção com encadeamento para trás .....	17
1.4.3.3 - Sistemas de produção com encadeamento misto .....	20
1.5 - Considerações finais .....	20

### CAPÍTULO 2 - Algoritmos Adaptativos Para a Identificação de Sistemas Variantes no Tempo

2.1 - Introdução.....	22
2.2 - Modelo do processo .....	23
2.3 - Estimador dos mínimos quadrados ponderados recursivo ..	23
2.4 - Algoritmo da variável instrumental (v.i.) .....	25
2.5 - Algoritmos de identificação adaptativos .....	26
2.5.1 - Algoritmos para rastrear variações lentas no tempo	

.....	27
2.5.1.1 - Regulagem do fator de esquecimento .....	28
2.5.1.2 - Busca aleatória .....	29
2.5.2 - Métodos de detecção de não-estacionaridades .....	30
2.5.2.1 - Critério do erro previsto .....	31
2.5.2.2 - Critério dos parâmetros estimados .....	35
2.6 - Algoritmos fatorizados .....	39
2.6.1 - Algoritmo dos mínimos quadrados com fatorização U-D .....	40
2.6.2 - Algoritmo de busca aleatória com fatorização U - D .....	41
2.7 - Conclusões .....	43

CAPÍTULO 3 - Análise do Desempenho dos Algoritmos Adaptativos  
Através de Exemplos de Simulação

3.1 - Introdução .....	45
3.2 - Critérios para comparação do desempenho dos algoritmos de estimação .....	46
3.2.1 - Critérios de detecção .....	46
3.2.1.1 - Taxas de alarme falso (TAF) .....	47
3.2.1.2 - Taxas de detecção e não-deteção (TD e TND) ..	48
3.2.1.3 - Atraso na detecção .....	48
3.2.2 - Critérios de identificação .....	49
3.2.2.1 - Polarização dos parâmetros .....	49
3.2.2.2 - Variância do erro de estimação .....	49
3.3 - Características das simulações .....	50
3.3.1 - Inicialização .....	50
3.3.2 - Geração das medidas .....	50
3.4 - Influência dos parâmetros de regulagem nos algoritmos de estimação .....	51
3.4.1 - Escolha dos níveis de adaptação para o algoritmo do traço adaptativo .....	52
3.4.2 - Escolha dos horizontes de cálculo dos critérios para o algoritmo do traço adaptativo .....	54
3.4.3 - Escolha dos níveis de adaptação para o algoritmo do filtro de Kalman adaptativo .....	55

3.4.4 - Escolha dos horizontes de cálculo dos critérios para o algoritmo do filtro de Kalman adaptativo .....	58
3.5 - Conclusões sobre a regulação dos parâmetros de detecção dos algoritmos adaptativos .....	71
3.5.1 - Relação sinal - ruído .....	72
3.5.2 - Tipo de variação (amplitude, velocidade) .....	72
3.5.3 - Ordem de grandeza dos polinômios A e B .....	72
3.5.4 - Tipo de entrada .....	73
3.6 - Análise comparativa entre os dois métodos de detecção de não-estacionaridades .....	73
3.7 - Seleção dos métodos de detecção .....	80

## CAPÍTULO 4 - Sistema Especialista em Modelagem

4.1 - Introdução .....	82
4.2 - Base de dados .....	84
4.3 - Base de conhecimentos .....	86
4.3.1 - Critério de convergência .....	90
4.3.2 - Critérios de desempenho .....	91
4.3.2.1 - Teste da correlação do erro previsto .....	92
4.3.2.2 - Teste da correlação cruzada .....	93
4.3.3 - Critérios para análise de excitação persistente ..	94
4.4 - Módulo de inferência e explicação .....	95
4.5 - Comunicação com o usuário .....	97
4.6 - Módulo supervisor e módulo de identificação .....	99
4.7 - Implementação computacional do sistema especialista em modelagem .....	100
4.7.1 - Implementação e operação da base de dados .....	101
4.7.2 - Implementação do módulo supervisor .....	103
4.7.3 - Implementação do módulo de identificação .....	104
4.7.4 - Implementação do módulo de inferência e base de conhecimento .....	104
4.7.5 - Implementação do módulo de explicação e comunicação com o usuário .....	110
4.8 - Exemplos da utilização do sistema especialista em modelagem na identificação de processos simulados .....	112
4.9 - Conclusões .....	120

CAPÍTULO 5 - Conclusões	
5.1 - Síntese do trabalho .....	122
5.2 - Conclusões e recomendações .....	123
APÊNDICE A - Algoritmo Com Fatorização U - D .....	125
APÊNDICE B - Algoritmo Com Fatorização U - D Para $P(t) = \tilde{P}(t) + Q(t)$ .....	133
REFERÊNCIAS BIBLIOGRÁFICAS .....	135

# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 OBJETIVOS DO TRABALHO

Nos últimos anos tem crescido o interesse por sistemas de controle que ajustem automaticamente os parâmetros do controlador, para compensar mudanças no processo e na vizinhança. Estes controladores são denominados controladores adaptativos [02, 19]. Entre os controladores adaptativos, o controlador auto-ajustável têm tido aceitação crescente em aplicações industriais [04, 19].

Este controlador é obtido a partir da separação das etapas de controle e identificação [04, 12], conforme mostrado na figura 1.1.

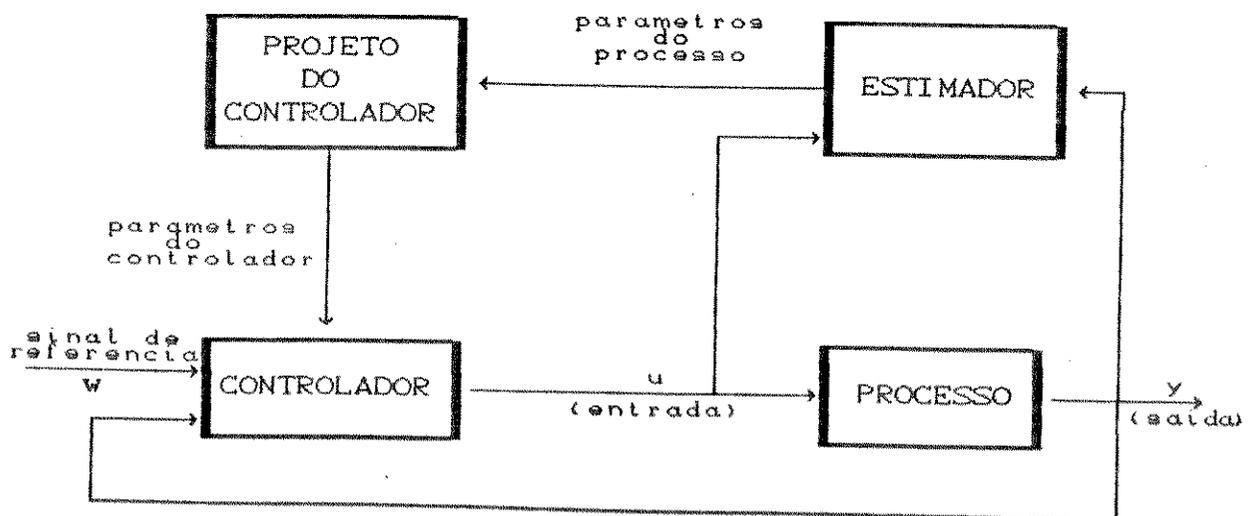


FIGURA 1.1

Diagrama de blocos do controlador auto-ajustável

Entretanto, alguns problemas têm limitado uma maior aplicação destes controladores em processos reais [12]. Um destes problemas é a inicialização dos algoritmos de identificação, pois é necessário conhecer a priori informações como: valor inicial do estimador e da sua matriz de covariância, ordem e estrutura do modelo a ser identificado, período de amostragem, etc, antes de se colocar o controlador em operação.

Outro problema é a necessidade de monitorar o algoritmo de estimação, para desligá-lo quando o sistema atinge o estado estacionário, ou alterar os valores iniciais para permitir a adaptação do modelo às variações no ponto de operação. A solução destes problemas requer o conhecimento heurístico de especialistas na área, impedindo a utilização de controladores por usuários que não tenham participado de sua implementação.

Assim, uma nova técnica, em que o controle "on-line" é baseado em um conjunto de diagnósticos sobre a identificação, utilizando um sistema especialista, foi proposta em [03] para viabilizar a utilização de controladores auto-ajustáveis.

O objetivo de um sistema especialista é codificar o conhecimento heurístico e a capacidade de decisão utilizados por um especialista, para obter a solução de problemas bem definidos em uma determinada área. No caso de controladores auto-ajustáveis, o sistema especialista através de recomendações automáticas possibilitará ao usuário, mesmo os não especializados em modelagem, a inicialização e a monitoração dos algoritmos de estimação e controle. O sistema especialista também fornecerá um diagnóstico sobre o desempenho deste algoritmo, auxiliando o usuário na tomada de decisões rápidas e inteligentes sobre a operação do controlador.

É dentro deste contexto que se propõe, neste trabalho, um sistema especialista para modelagem de processos. Este sistema especialista deve ser capaz de realizar a cada iteração, um diagnóstico sobre o comportamento do algoritmo utilizado na

estimação de um sistema variante no tempo.

Os diagnósticos são obtidos pelo sistema especialista, através da monitoração de parâmetros que caracterizam o algoritmo de identificação, isto é, o traço da matriz de covariância, o fator de esquecimento e uma estimativa das características das perturbações existentes no processo, em cada instante de amostragem. Os diagnósticos são fornecidos ao usuário quando determinadas condições de operação contidas na base de conhecimentos do sistema especialista não são satisfeitas.

O sistema especialista em modelagem, proposto neste trabalho, é composto de módulos independentes, gerenciados por um módulo supervisor. Estes diversos módulos realizam as seguintes tarefas: i) identificação do modelo do sistema; ii) aquisição de dados, incluindo aqueles que são candidatos potenciais a causarem dificuldades no processo de identificação; iii) conversão do conhecimento empírico e informal, fornecido pelo especialista, em uma estrutura passível de análise lógica; iv) especificações dos objetivos da identificação, considerando inclusive as restrições do algoritmo; v) realização de inferências; vi) explicação da linha de raciocínio utilizada para obter os diagnósticos; vii) realização de alterações nas condições iniciais e em parâmetros de modo a auxiliar a etapa de estimação.

O software que compõe o sistema especialista em modelagem de processos é formado pelos seguintes módulos: base de conhecimento, módulo de inferência e explicação, módulo supervisor, módulo de identificação, base de dados e interface com o usuário. Os algoritmos de estimação desenvolvidos para o módulo de identificação são baseados numa modificação do algoritmo dos mínimos quadrados ponderados recursivo, que permite estimar sistemas variantes no tempo [07].

A apresentação deste trabalho está organizada da seguinte forma: neste capítulo definem-se as principais partes de um sistema especialista e especificam-se as características

desejadas para um funcionamento adequado. Em seguida, resumem-se as várias representações de conhecimento que existem em inteligência artificial, e justifica-se a escolha de um sistema especialista baseado em regras como representação adequada para um sistema especialista em modelagem. Por fim descreve-se a representação de conhecimento e as estratégias de controle utilizadas em um sistema especialista baseado em regras.

No capítulo 2, apresentam-se os algoritmos de identificação de processos propostos para a modelagem de sistemas variantes no tempo. Estes algoritmos são obtidos a partir da utilização de métodos de detecção de não-estacionaridades, aliados a um procedimento de regulagem da matriz de covariância  $P(t)$ , do algoritmo dos mínimos quadrados, que permite rastrear as não-estacionaridades em sistemas variantes no tempo. Dois métodos de detecção de não-estacionaridades são desenvolvidos neste capítulo, um baseado no erro previsto, e o outro baseado nos parâmetros estimados. Dois métodos de regulagem da matriz de covariância são também apresentados: utilização de fator de esquecimento e busca aleatória. Da combinação destes métodos de detecção e de regulagem da matriz  $P(t)$  obtêm-se os algoritmos de identificação denominados adaptativos.

Em particular, desenvolvem-se no capítulo 2 os seguintes estimadores adaptativos: o algoritmo do traço adaptativo, que regula a matriz de covariância  $P(t)$  utilizando fator de esquecimento e detecta não-estacionaridades utilizando o método do erro previsto [07]; e o algoritmo do Filtro de Kalman Adaptativo que utiliza o método de detecção de não-estacionaridades baseado nos parâmetros estimados e busca aleatória para regulagem dos elementos da matriz  $P(t)$ .

No capítulo 3 realiza-se um estudo comparativo do desempenho dos algoritmos adaptativos baseado em exemplos simulados. O desempenho dos algoritmos é avaliado a partir de critérios relacionados com a eficiência na detecção de não-estacionaridades e a convergência do estimador para os

parâmetros verdadeiros.

O capítulo 4 contém uma descrição do sistema especialista em modelagem, tendo como referência os conceitos e definições apresentados no capítulo 1. A implementação de cada um dos módulos que formam o sistema especialista é explicitada, e apresentam-se resultados obtidos com a utilização do sistema especialista na identificação de processos simulados.

No capítulo 5 resumem-se as conclusões alcançadas com este trabalho e novos trabalhos são sugeridos.

## 1.2 SISTEMAS ESPECIALISTAS: DEFINIÇÃO E CARACTERÍSTICAS

Sistemas especialistas são programas que resolvem problemas, cuja solução requer o conhecimento de um especialista [16, 17, 25]. Nos programas convencionais, baseados em algoritmos, o conhecimento é inserido dentro do programa código, como uma seqüência de operações determinadas sobre os dados. Nos sistemas especialistas, existe uma separação entre o conhecimento utilizado para resolver o problema e o programa código que o manipula para a obtenção da solução desejada. Esta separação permite que o usuário do sistema especialista avalie e modifique o programa - código, sem interferir com o conhecimento utilizado para resolvê-lo.

Um sistema especialista é constituído de três partes principais [14, 16, 25]: i) a Base de Dados que contém a representação simbólica dos fatos característicos da área de aplicação; ii) a Base de Conhecimentos que contém a representação do conhecimento (regras, frames) que usam os fatos da Base de Dados para tomar uma decisão e, iii) a Máquina de Inferência que controla o fluxo de informação no sistema e aplica a Base de Conhecimentos à Base de Dados, para obter novos dados e

conhecimentos.

As mais importantes características de um sistema especialista são a quantidade de conhecimento que constitui a Base de Conhecimento, e como este conhecimento é manipulado através de um raciocínio simbólico, para se atingir a solução de um problema específico. Como consequência, a escolha, a forma e a interpretação dos símbolos utilizados na representação do conhecimento são pontos críticos na elaboração de um sistema especialista. Portanto, um sistema especialista deve possuir uma base de conhecimento transparente e flexível.

Como flexibilidade entende-se a capacidade de adquirir novo conhecimento e modificar o conhecimento anterior [16].

Transparência é a capacidade de um sistema especialista explicar seu raciocínio [16, 25]. Os usuários de um sistema especialista dificilmente aceitarão resultados, sem que tenham sido convencidos da precisão do raciocínio utilizado, para se chegar a estes resultados. Portanto, é importante que os sistemas especialistas sejam capazes de explicar sua linha de raciocínio, e de responder questões relativas a esse raciocínio. Para tanto um sistema especialista deve possuir conhecimento sobre seu próprio modo de operação [25]. O conhecimento que um sistema especialista tem sobre sua forma de raciocinar é denominado *metaconhecimento* ou conhecimento de segunda ordem, que significa conhecimento sobre conhecimento.

A habilidade de examinar seu processo de raciocínio e explicar sua operação confere aos sistemas especialistas as seguintes vantagens sobre os programas convencionais: i) o usuário acredita nos resultados obtidos com um sistema especialista, pois é capaz de acompanhar passo a passo a obtenção desta solução, aumentando a confiabilidade da solução; ii) o desenvolvimento do sistema é mais rápido, devido a facilidade de detecção e correção de erros; iii) a operação do sistema é explícita ao invés de implícita como nos programas convencionais;

iv) torna-se fácil prever e testar os efeitos de mudanças sobre a operação do sistema.

Um sistema especialista também deve dispor de uma interface homem-máquina eficiente, para permitir uma iteração fácil e adequada com o usuário.

O usuário de um sistema especialista pode ser uma pessoa especializada, com grandes conhecimentos na área. Neste caso, o sistema especialista deve ser capaz de extrair do usuário conhecimento novo, a fim de aumentar sua base de conhecimentos. O usuário também pode ser um iniciante, neste caso, o sistema especialista deve ser capaz de transmitir informações suficientes, treinando o usuário na resolução de um problema específico.

Desta forma, os programas que compõem a interface homem-máquina de um sistema especialista incluem ferramentas de edição, para facilitar a aquisição e modificação do conhecimento especializado, ou dos dados contidos na base de dados. Esta interface contém ainda software de entrada e saída, que ajudam o usuário a fornecer ou receber informações do sistema especialista.

### 1.3 REPRESENTAÇÃO DO CONHECIMENTO

Sistemas especialistas manipulam uma grande quantidade de conhecimento especializado e distinto: o conhecimento sobre o problema específico a resolver, o conhecimento geral de como resolver problemas, o conhecimento de como interagir com o usuário, etc [16, 25]. Portanto, a escolha da representação de conhecimento torna-se um ponto crítico na elaboração de qualquer sistema especialista.

Entende-se aqui por representação de conhecimento como o conhecimento é estruturado dentro de um programa [25]. Existem, em inteligência artificial, várias técnicas para representação de conhecimento que podem ser usadas sozinhas, ou em conjunto para construir sistemas especialistas. A escolha de uma destas técnicas considera: i) a capacidade de representar o conhecimento necessário para a solução dos problemas da área de interesse, ii) a capacidade de adquirir novas informações e, iii) a facilidade em manipular as estruturas representacionais existentes, para obter novas estruturas que corresponderão ao conhecimento inferido. Também é importante que a estrutura de conhecimento seja capaz de incorporar informações adicionais que aumentem a eficiência da máquina de inferência.

As duas formas de representação de conhecimento mais utilizadas na construção de sistemas especialistas são as regras e os frames [17, 25]. A seguir discute-se brevemente cada uma delas.

### 1.3.1 Métodos baseados em frames

A representação de conhecimento baseada em frames utiliza uma rede de nós conectados por arcos, e organizada em hierarquia. Os nós representam um conceito que pode ser descrito por atributos e valores associados a este nó, os arcos são as relações existentes entre os conceitos. Os nós dos níveis inferiores possuem automaticamente as propriedades dos nós do nível superior da hierarquia.

Neste tipo de representação, o conhecimento é representado como uma coleção de fatos estáticos (invariantes no tempo) acompanhados por um conjunto de procedimentos para manipulá-los. Esta representação facilita a adição de novos fatos ao sistema, sem alterações, nem dos fatos antigos, nem dos procedimentos para manipulá-los.

Os frames contém informações sobre os vários aspectos dos objetos ou situações que eles descrevem [17, 25]. A estrutura de um método baseado em frames, sugere que eles devem ser utilizados em áreas onde a forma e o conteúdo dos dados representam uma condição fundamental na solução do problema, como por exemplo, em interpretação de imagens e cenas, ou em compreensão da fala [25].

Considera-se como métodos baseados em frames as redes semânticas, os frames propriamente ditos, os scripts e as redes de dependência conceitual. Uma descrição mais detalhada destes e de outros métodos de representação de conhecimento pode ser encontrada em [14] e [17].

### 1.3.2 Métodos baseados em regras

A representação de conhecimento através de regras é um modo natural de descrever processos onde ocorrem mudanças rápidas e complexas do meio [25]. Um conjunto de regras especificam como o programa deve reagir a uma mudança de dados, sem precisar de um fluxo de controle detalhado [25].

Métodos baseados em regras usam sentenças do tipo SE condição ENTÃO ação. Quando a situação corrente satisfaz a parte "SE" de uma regra, a ação especificada na parte "ENTÃO" é executada. Estas ações são do tipo cálculo do valor de um determinado atributo, escolha de estratégias de controle na máquina de inferência, enviar uma mensagem de advertência, etc.

Nos métodos baseados em regras, tanto o conhecimento factual (ex.: condições iniciais do problema), como o heurístico (ex.: estratégia de resolução de problemas) são descritos na forma SE \_\_ ENTÃO. A operação destes sistemas é controlada por um procedimento que analisa os antecedentes (condição) das regras, e executa seus consequentes (conclusão). Este procedimento simplifica o funcionamento da máquina de inferência. A utilização

de regras também facilita a explicação do raciocínio que o sistema especialista utilizou para obter alguma conclusão particular, pois os conseqüentes de uma determinada regra podem ser utilizados como antecedentes para uma outra regra, até se alcançar a conclusão ou diagnóstico desejado. Esse encadeamento sucessivo de regras é denominado cadeia de inferências.

Neste trabalho escolheu-se um método baseado em regras, denominado Sistemas de Produção, para representação do conhecimento do sistema especialista em modelagem. Este método foi escolhido por causa da diversidade de conhecimento a ser representada. As várias tarefas a serem realizadas pelo sistema especialista são de naturezas diversas, o sistema especialista deve fazer aquisição de dados, identificar o modelo, monitorar os parâmetros do modelo identificado, realizar um diagnóstico sobre a identificação, sugerir modificações para melhorar o desempenho do algoritmo que está sendo executado, além de fornecer a linha de raciocínio utilizada para obter um determinado diagnóstico.

As regras fornecem um modo formal e simples de representar recomendações, diretivas e estratégias, além de serem bastante apropriadas, quando a área de conhecimento requer muita heurística [25], como é o caso da regulagem das variáveis dos algoritmos de identificação, usados no sistema especialista para modelagem de processos. A cadeia de inferências formada naturalmente pelas sucessivas regras executadas, também fornece um modo simples de explicação de raciocínio, pois sempre que o usuário desejar uma explicação, a cadeia de inferências pode ser mostrada passo a passo, indicando o caminho utilizado para se chegar a conclusão questionada.

Para a escolha do método de representação do conhecimento também considerou-se a linguagem a ser utilizada na elaboração dos programas. Como era necessário realizar cálculos algébricos, operações com ponto flutuante, etc, na execução do algoritmo de identificação utiliza-se a linguagem PASCAL que possui estruturas sintáticas que facilitam a implementação de

regras do tipo SE \_\_ ENTÃO.

#### 1.4 SISTEMAS DE PRODUÇÃO

Sistemas de produção são os sistemas utilizados em inteligência artificial, para resolver problemas que utilizam regras para representar o conhecimento. São constituídos de três partes distintas: a Base de Dados, a Base de Conhecimentos e a Máquina de Inferência. A Base de Dados contém os dados relativos à solução do problema procurada, a Base de Conhecimentos é o conjunto de Regras de Produção que contém todo o conhecimento codificado na forma SE \_\_ ENTÃO e a Máquina de Inferência realiza o controle do sistema.

Para a operação do sistema de produção, a máquina de inferência aplica o conjunto de regras sobre a base de dados. Cada regra contém uma ou mais condições que, se satisfeitas pela base de dados, colocam a regra em condição de ser executada. A máquina de inferência seleciona uma dessas regras e a executa, gerando uma nova base de dados. Em seqüência, novas regras são selecionadas e executadas até que se obtenha uma condição sobre a base de dados, que corresponda ao objetivo desejado.

Construir um sistema especialista utilizando sistema de produção, requer a especificação da Base de Dados, do conjunto de Regras de Produção e da estratégia de controle utilizada pela Máquina de Inferência, para alcançar os objetivos especificados. Cada um destes itens é discutido a seguir.

##### 1.4.1 Representação dos dados (Base de Dados)

Os vários métodos de representação de conhecimento estruturado [17] embora bastante diferentes, baseiam-se no fato de que entidades complexas podem ser descritas como uma coleção de atributos e valores associados.

A mais completa forma de descrição de um fato é através da tripla associativa

### OBJETO X ATRIBUTO X VALOR

que pode ser interpretada como "objeto" tem "atributo" com "valor". O nome geral das triplas associativas é uma "característica" do domínio do problema, e "atributo" é um nome geral do conjunto de valores. Por exemplo, o atributo COR pode ter como valores azul, verde ou vermelho. Portanto, a descrição de um objeto é uma combinação única dos valores dos vários atributos, no conjunto de todas as possíveis descrições (espaço de descrição).

Fatos e regras em um sistema especialista nem sempre são verdadeiros ou falsos, muitas vezes há um certo grau de incerteza sobre a validade do fato ou a precisão da regra. Neste caso, os sistemas especialistas utilizam uma medida de confiabilidade associada à regra.

#### 1.4.2 Representação do Conhecimento

As inferências realizadas por um sistema de produção são obtidas da aplicação das regras de produção sobre a Base de Dados. As regras consistem de asserções dadas em forma implicacional do tipo

SE <condição> ENTÃO <ação>.

As regras representam a estrutura da Base de Conhecimentos, e obedecem uma hierarquia em relação ao conhecimento que representam. As regras superiores são as denominadas *metaregras*, que são regras sobre regras e representam um conhecimento de segunda ordem. Essas regras direcionam a Máquina de Inferência para as áreas aplicáveis ao objetivo

presente, e podem ser consideradas como divisores de memória, pois dividem a Base de Conhecimentos em sub-domínio.

A hierarquia na representação do conhecimento também determina a solução de conflitos entre regras. Normalmente, a Base de Conhecimentos guia a Máquina de Inferência na direção da regra que possui as seguintes características: é uma metaregra; é mais geral que as outras, isto é, ocupa um nível mais alto na hierarquia; é confirmada por redundância, isto é, pode ser inferida por diferentes caminhos.

### 1.4.3 Máquina de Inferência

A Máquina de Inferência seleciona uma regra da Base de Conhecimentos e a aplica à Base de Dados, gerando uma nova Base de Dados. Este procedimento é repetido até que a nova Base de Dados corresponda ao objetivo desejado. Para realizar a seleção e aplicação das regras a Máquina de Inferência utiliza-se de estratégias de controle. Estas estratégias de controle são classificadas em encadeamento para frente, encadeamento para trás e encadeamento misto conforme o procedimento de seleção e execução das regras.

#### 1.4.3.1 Sistemas de produção com encadeamento para frente

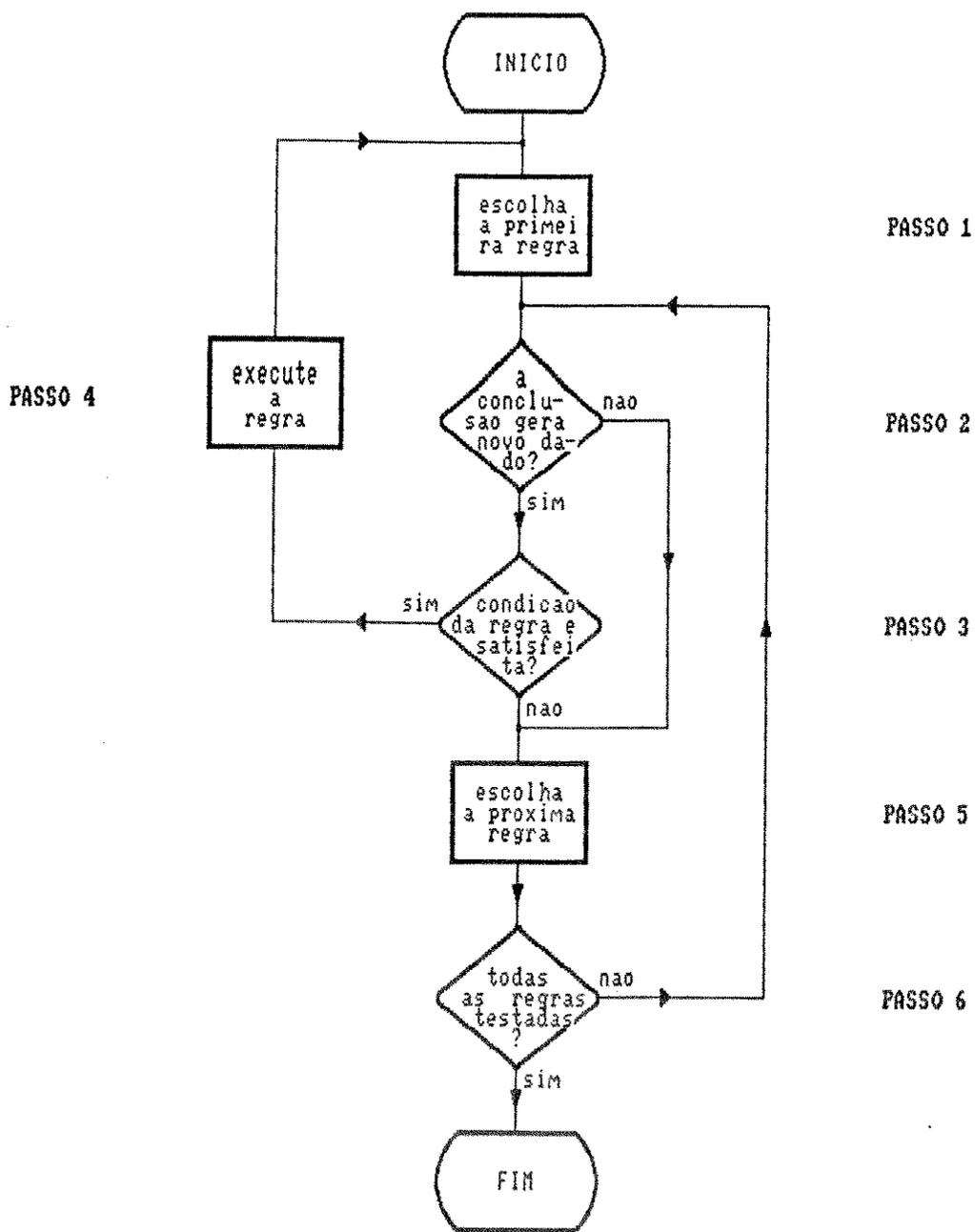
Com esta estratégia de controle, a Máquina de Inferência funciona em modo de encadeamento para frente ("forward chaining"), ou seja, parte-se de um estado inicial da Base de Dados e atinge-se o estado objetivo.

Nesta estratégia de controle a Máquina de Inferência analisa os antecedentes (condições) de todas as regras da Base de Conhecimentos, e seleciona, se houver, um conjunto de regras que tenham seus antecedentes satisfeitos. Este conjunto de regras é organizado de acordo com a hierarquia da Base de Conhecimentos e

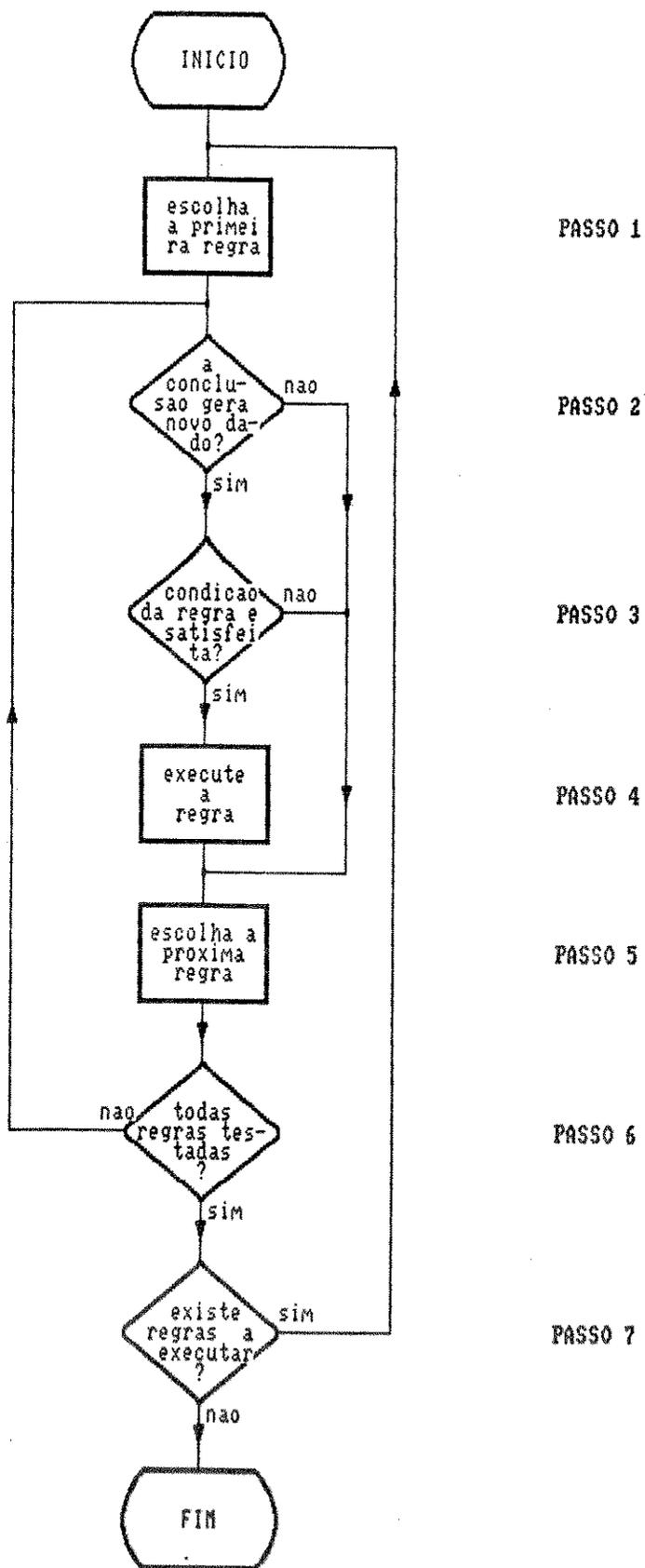
a regra de maior prioridade é executada, isto é, seus conseqüentes (ações) são executados.

Após a execução da regra de maior prioridade, existem duas possíveis estratégias a seguir. A primeira é recomeçar a procura de um novo conjunto de regras e executar novas regras, conforme o fluxograma da figura 1.2. Neste fluxograma a estratégia de controle com encadeamento para frente funciona selecionando uma regra de acordo com a lista de prioridades (passo 1), em seguida o conseqüente da regra é examinado, para se determinar se a conclusão resultante da execução da regra selecionada produz um dado útil à solução do problema desejado (passo 2). Em caso afirmativo, os antecedentes das regras são testados (passo 3) e se forem satisfeitos pela Base de Dados corrente, a regra é executada no passo seguinte (passo 4). Após a execução do passo 4 volta-se ao passo inicial (passo 1) quando um novo conjunto de regras é escolhido de acordo com a lista de prioridades. Em caso negativo nos passos 2 ou 3, a regra selecionada é descartada e uma nova regra no conjunto de regras é selecionada e volta-se ao passo 2.

A segunda estratégia é continuar executando as regras selecionadas em ordem de prioridade até que todas as regras do conjunto de regras tenham sido utilizadas. Em seguida, recomeça-se a procura de um novo conjunto de regras. O fluxograma deste método é mostrado na figura 1.3. Neste fluxograma como no anterior, escolhe-se inicialmente a regra de maior prioridade no conjunto de regras (passo 1), testa-se seus conseqüentes (passo 2) e seus antecedentes (passo 3). Se os passos 2 e 3 são satisfeitos a regra é executada (passo 4) e a próxima regra de maior prioridade do conjunto é selecionada (passo 5), voltando-se ao passo 2. Se os passos 2 ou 3 não são satisfeitos, a próxima regra é selecionada e volta-se ao passo 2. Nesta estratégia de controle com encadeamento para frente uma regra já executada não pode ser executada novamente até que tenham sido testadas todas as regras do conjunto (passo 6). Depois de testadas todas as regras, analisa-se novamente a Base de Conhecimentos (passo 7) e



**FIGURA: 1.2:**  
 "execute a regra de maior prioridade e repita"  
 encadeamento para frente



**FIGURA. 1.3:**  
 "execute todas as regras e repita"  
 encadeamento para frente

se seleciona, se houver, um novo conjunto de regras organizado em ordem de prioridades, recomeçando o processo. Caso contrário a busca e execução de regras é suspensa.

A estratégia de controle com encadeamento para frente é também conhecida como raciocínio antecedente, controle dirigido por dados ou controle "bottom-up".

#### 1.4.3.2 Sistemas de produção com encadeamento para trás

Neste tipo de sistema a Máquina de Inferência funciona em modo de encadeamento para trás ("backward chaining"), ou seja, parte-se do objetivo final, e através da aplicação de um movimento reverso atinge-se o estado inicial da Base de Dados. Cada movimento para trás produz um estado sub-objetivo, a partir do qual o estado objetivo pode ser alcançado com um movimento para frente.

Nesta estratégia de controle a Máquina de Inferência analisa os conseqüentes das regras na Base de Conhecimentos para encontrar, se houver, aquela que executada gera o objetivo desejado. Quando mais de uma regra é encontrada, forma-se um conjunto de regras organizado hierarquicamente e a regra de maior prioridade é selecionada. Os antecedentes da regra selecionada são testados para verificação de sua validade. Se válidos, a regra é executada e o objetivo é alcançado. Se os antecedentes não são satisfeitos porque uma ou mais premissas são falsas, a próxima regra na ordem de prioridades é selecionada e testada. Se uma ou mais premissas de um antecedente de uma regra selecionada são indefinidas, estes antecedentes se tornam sub-objetivos, e forma-se um conjunto de regras para se resolver esse sub-objetivo, repetindo-se o procedimento acima para os vários sub-objetivos criados.

Deste modo a Máquina de Inferência trabalha formando sub-objetivos hipóteses que devem ser estabelecidas para

estabelecer a hipótese original) que serão avaliados até que o objetivo original possa ser alcançado ou não haja mais regras a analisar. O funcionamento desta estratégia de controle pode ser melhor entendido com o exemplo a seguir.

**EXEMPLO :** Considere uma base de conhecimentos formada pelas seguintes regras:

1. A & B & C → D
2.     A & J → G
3.     D & F → G
4.         B → C
5.         F → B
6.         L → J
7.         G → H

onde as letras representam características e o símbolo → significa "implica".

A base de dados inicial é A e F. O objetivo é saber se a característica H é verdadeira. Considera-se que, se uma característica é verdadeira, ela pode ser incluída na base de dados.

Aplicando-se uma estratégia de controle com encadeamento para trás tem-se:

1. Usando a regra 7, se G é verdadeiro então H é verdadeiro.
2. O sub-objetivo é G e não está na base de dados.
3. As regras 2 e 3 são capazes de avaliar G.
4. Forme o conjunto de regras organizado em ordem de prioridades: a regra de maior prioridade é a de menor número.
5. Usando a regra 2, se A e J são verdadeiros então G é verdadeiro.
6. A está na base de dados portanto é verdadeiro. O sub-objetivo é J.
7. Usando a regra 6, se L é verdadeiro então J é verdadeiro.

8. L não está na base de dados, portanto passa a ser o novo sub-objetivo.
9. Nenhuma regra tem como conseqüente L então esta linha de raciocínio falha.
10. A próxima regra do conjunto formado no passo 4, é a regra 3.
11. Usando a regra 3, se D e F são verdadeiros então G é verdadeiro.
12. F é um dado na base de dados portanto verdadeiro. O novo sub-objetivo é D.
13. Usando a regra 1, se A e B e C são verdadeiros então D é verdadeiro.
14. A está na base de dados portanto é verdadeiro. Os novos sub-objetivos são B e C.
15. Usando a regra 4 se B é verdadeiro então C também o é, logo o passo 14 reduz-se a um único sub-objetivo B.
16. Usando a regra 5, se F é verdadeiro então B é verdadeiro.
17. F está na base de dados, logo B e C são verdadeiros, conseqüentemente D é verdadeiro, G é verdadeiro e por fim H é verdadeiro e deve ser colocado na base de dados.

Um diagrama da solução é mostrado na figura 1.4 a seguir.

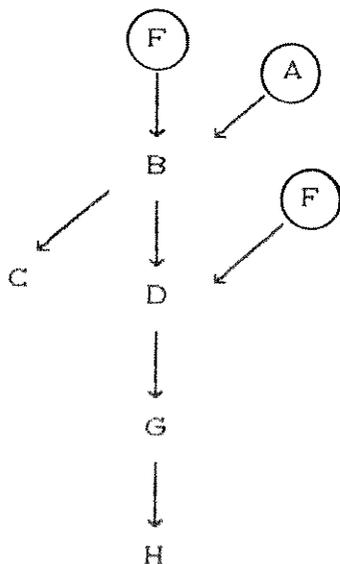


FIGURA 1.4  
 Solução do Exemplo  
 Encadeamento Para Trás

Quando uma máquina de inferência trabalha no modo de encadeamento para trás, uma forma natural e simples de explicação de raciocínio é conseguida se o programa de controle gravar um histórico das decisões tomadas. Este histórico é depois escrito em forma de regras de produção, que é uma representação de conhecimento mais simples, e é então interpretado por um programa responsável pelas respostas às perguntas do usuário, devendo justificar as decisões tomadas e modificar a base de conhecimentos a partir dessa interação com o usuário.

Esta estratégia de controle também é conhecida na literatura como raciocínio conseqüente, controle dirigido por objetivos ou controle "Top-down".

#### 1.4.3.3 Sistemas de produção com encadeamento misto

Utiliza-se neste caso, uma estratégia de controle mista onde tanto a descrição dos estados como a descrição dos objetivos fazem parte da base de dados. Regras de encadeamento para frente (regras-F) são aplicadas na descrição de estados e regras de encadeamento para trás (regras-B) são aplicadas na descrição dos objetivos. Nesta estratégia a máquina de inferência tenta casar as descrições de estado com as descrições dos objetivos na base de dados, até encontrar o objetivo desejado.

### 1.5 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas diversas características dos sistemas especialistas, ressaltando-se a escolha da representação do conhecimento como um ponto chave para o bom desempenho deste tipo de sistema.

A representação de conhecimento baseada em regras

(sistema de produção) foi escolhida para o sistema especialista modelagem, porque regras fornecem um modo formal e simples de representar a diversidade de conhecimento necessário para a regulação das variáveis dos algoritmos de identificação.

As componentes de um sistema de produção, Base de Dados, Base de Conhecimentos e Máquina de Inferência, assim como o seu funcionamento foram descritos.

No capítulo 4, o sistema especialista em modelagem será descrito tendo como referência os conceitos e definições apresentados neste capítulo.

## CAPÍTULO 2

### ALGORITMOS ADAPTATIVOS PARA A IDENTIFICAÇÃO DE SISTEMAS VARIANTES NO TEMPO

#### 2.1 INTRODUÇÃO

Neste capítulo apresentam-se as características dos algoritmos de identificação de processo que são utilizados no sistema especialista para modelagem. Estes algoritmos são uma extensão do estimador dos mínimos quadrados ponderados recursivo (MQPR) para identificação de sistemas não-estacionários.

Os algoritmos para identificação de sistemas variantes no tempo desenvolvidos neste trabalho utilizam-se de métodos de detecção de não-estacionaridades [07], aliados a um procedimento de manipulação da matriz de covariância  $P(t)$  do algoritmo dos mínimos quadrados o que permite a modelagem de sistemas variantes no tempo.

Na primeira parte deste capítulo descreve-se o método dos mínimos quadrados [02, 06] e o método da variável instrumental [15], que é utilizado quando a perturbação atuando no processo não é branca.

Os algoritmos adaptativos capazes de rastrear as variações dos parâmetros do processo são desenvolvidos na seção 2.5, onde no item 2.5.1 apresentam-se procedimentos para regulagem da matriz de covariância e no item 2.5.2 apresentam-se os métodos de detecção de não-estacionaridades.

Finalmente na seção 2.6 apresenta-se o algoritmo dos

mínimos quadrados com fatorização U-D que foi implementado no sistema especialista.

## 2.2 MODELO DO PROCESSO

Utilizam-se modelos discretos do tipo entrada-saída ARMAX descrito pela equação 2-1:

$$A(q^{-1})y(t) = B(q^{-1})q^{-d}u(t) + v(t) \quad (2-1)$$

onde os polinômios  $A(q^{-1})$  e  $B(q^{-1})$  são definidos por

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

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

e  $v(t)$  é uma seqüência de variáveis aleatórias não necessariamente branca, com média nula e variância dada por  $\sigma_v^2$ ;  $d$  é o atraso de transporte e  $na$  e  $nb$  são as ordens do polinômio  $A(q^{-1})$  e  $B(q^{-1})$  respectivamente.

## 2.3 ESTIMADOR DOS MÍNIMOS QUADRADOS PONDERADOS RECURSIVO

Para facilitar a obtenção dos estimadores dos mínimos quadrados, reescreve-se a eq.2-1 como um modelo de regressão linear, onde os elementos desconhecidos estão contidos em um único vetor de parâmetros  $\theta(t) \in \mathbb{R}^n$ , com  $n = na + nb$ , que se deseja identificar e os valores das medidas passadas da entrada e saída estão contidos em um vetor de medidas  $\varphi(t)$ . Isto é:

$$y(t) = \varphi^T(t)\theta(t-1) + v(t) \quad (2-3)$$

onde

$$\varphi^T(t) = [ -y(t-1) \dots -y(t-n_a) \quad u(t-1) \dots u(t-n_b) ]$$

$$\theta^T(t) = [ a_1(t) \quad a_2(t) \dots a_{n_a}(t) \quad b_1(t) \quad b_2(t) \dots b_{n_b}(t) ]$$

O estimador dos mínimos quadrados é obtido minimizando-se em relação ao vetor de parâmetros  $\theta(t)$  o seguinte critério quadrático:

$$J_N(\theta) = 1/N \sum_{t=1}^N w(t) [ y(t) - \varphi^T(t) \theta(t) ]^2 \quad (2-4)$$

com  $w(t)$  uma ponderação variável escolhida tal que

$$w(t) = \mu(t) \prod_{i=t}^T \lambda(i) \geq 0 \quad (2-5)$$

onde  $\mu(t)$  é um coeficiente que permite ponderar a medida em função de sua qualidade, e  $\lambda(t)$  é o fator de esquecimento que permite ao estimador rastrear variações nos parâmetros.

Da minimização da equação 2.4 resulta o estimador  $\hat{\theta}(t)$  dado por:

$$\hat{\theta}(t) = \left[ \sum_{t=1}^T w(t) \varphi(t) \varphi^T(t) \right]^{-1} \left[ \sum_{t=1}^T w(t) \varphi(t) y(t) \right] \quad (2-6)$$

A forma recursiva do estimador dos mínimos quadrados ponderados é descrita pelas seguintes equações:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + G(t) [ y(t) - \varphi^T(t) \hat{\theta}(t-1) ] \quad (2-7)$$

$$G(t) = P(t) \varphi(t) = \frac{P(t-1) \varphi(t)}{1/\mu(t) + \varphi(t)^T P(t-1) \varphi(t)} \quad (2-8)$$

$$P(t) = \left[ P(t-1) - \frac{P(t-1) \varphi(t) \varphi^T(t) P(t-1)}{1/\mu(t) + \varphi^T(t) P(t-1) \varphi(t)} \right] 1/\lambda(t) \quad (2-9)$$

onde  $G(t)$  é o ganho do estimador dos mínimos quadrados e  $P(t)$  é a matriz de covariância do erro de estimação,  $P(t) \in$

$\mathbb{R}^{(na+nb) \times (na+nb)}$

É bem conhecido na literatura [06, 15] que se a perturbação  $v(t)$  é um ruído branco, então o estimador dos mínimos quadrados  $\hat{\theta}(t)$  converge em probabilidade para o vetor  $\theta(t)$ . Quando a perturbação não é um ruído branco, em geral, existem diversos métodos [01, 13] que modelam a perturbação para se obter um estimador não polarizado. Entretanto é bem conhecido [09, 15] que estes métodos além de aumentarem o tempo de cálculo apresentam uma convergência lenta nos parâmetros estimados do modelo da perturbação.

Neste trabalho, para se obter valores estimados não-polarizados quando a perturbação não é branca, utiliza-se o método da variável instrumental [15] que como é descrito a seguir não necessita de um modelo para o ruído.

#### 2.4. ALGORITMO DA VARIÁVEL INSTRUMENTAL (V.I)

Para que o estimador  $\hat{\theta}(t)$  seja assintoticamente não polarizado utiliza-se uma nova variável, denominada variável instrumental, na equação 2-6. Nesta equação substitui-se a variável  $\varphi(t)$  por uma variável instrumental  $z(t)$ , tal que esta variável seja correlacionada com o vetor  $\varphi(t)$  mas seja não correlacionada com a perturbação [13, 15], obtendo-se a seguinte equação para o estimador:

$$\hat{\theta}(t) = \left[ \sum_{l=1}^N w(t) z(t) \varphi^T(t) \right]^{-1} \left[ \sum_{l=1}^N w(t) z(t) y(t) \right] \quad (2-10)$$

Este algoritmo implementado no sistema especialista utiliza como variável instrumental um vetor  $z(t)$  formado por

valores filtrados da saída e valores atrasados da entrada, como mostrado abaixo:

$$z^T(t) = [ -y^*(t-1) \dots -y^*(t-na) \ u(t-1) \dots u(t-nb) ] \quad (2-11)$$

os valores  $y^*(t)$  são obtidos a partir de

$$y^*(t) = z^T(t) \hat{\theta}(t) \quad (2-12)$$

O algoritmo recursivo para o estimador da variável instrumental é descrito por:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + G(t) [ y(t) - \varphi^T(t) \hat{\theta}(t-1) ] \quad (2-13)$$

$$G(t) = P(t) z(t) = \frac{P(t-1) z(t)}{1/\mu(t) + \varphi^T(t) P(t-1) z(t)} \quad (2-14)$$

$$P(t) = \left[ P(t-1) - \frac{P(t-1) z(t) \varphi^T(t) P(t-1)}{1/\mu(t) + \varphi^T(t) P(t-1) z(t)} \right] 1/\lambda(t) \quad (2-15)$$

## 2.5. ALGORITMOS DE IDENTIFICAÇÃO ADAPTATIVOS

Quando o algoritmo dos mínimos quadrados converge os elementos da matriz de covariância  $P(t)$  e o ganho de correção  $G(t)$  tendem a zero [26]. Quando o processo que está sendo identificado é invariante no tempo esta convergência é desejada. Entretanto, modelos invariantes no tempo permitem representar o comportamento do sistema real somente em torno de um ponto de operação. Neste trabalho, deseja-se obter algoritmos que possibilitem uma adaptação contínua do modelo do processo às variações do ponto de operação do sistema, isto é, deseja-se a obtenção de modelos com parâmetros variantes no tempo. Neste caso, se o ganho do estimador decresce significativamente o algoritmo perde sua capacidade de adaptação frente às variações

de parâmetros. Os algoritmos que serão apresentados neste capítulo, mantêm a capacidade de adaptação do estimador limitando a valores mínimos os elementos da matriz de covariância  $P(t)$ .

Definindo o erro previsto como

$$\varepsilon_p(t) = y(t) - \varphi^T(t) \hat{\theta}(t-1) \quad (2-16)$$

A equação do estimador 2-7 pode ser reescrita como segue

$$\hat{\theta}(t) = \hat{\theta}(t-1) + G(t) \varepsilon_p(t) \quad (2-17)$$

Quando os parâmetros do sistema variam no tempo a saída do processo desvia-se da saída do modelo, aumentando o erro previsto. Se não é permitido aos elementos da matriz  $P(t)$  tornarem-se muito pequenos, o ganho de correção  $G(t)$  [  $G(t) = P(t)\varphi(t)$  ] também não será pequeno e o erro previsto corrigirá continuamente  $\hat{\theta}(t)$  na eq. 2-17. Esta correção permite ao estimador seguir as variações de parâmetros. Entretanto, o erro previsto também inclui perturbações estocásticas que se deseja eliminar. Evitando-se que os elementos da matriz  $P(t)$  diminuam aumentam-se os erros aleatórios no estimador  $\hat{\theta}(t)$ . Tem-se portanto, um compromisso entre a qualidade do estimador ( $P(t)$  pequeno) e a sua capacidade de adaptação ( $P(t)$  grande).

### 2.5.1 Algoritmos para rastrear variações lentas no tempo

Para processos que variam lentamente no tempo, é necessário fornecer ao algoritmo dos mínimos quadrados ponderados recursivo uma capacidade de adaptação mínima, impedindo que o ganho tenda para zero. Esta capacidade pode ser obtida dando-se uma maior importância às novas medidas (regulagem do fator de esquecimento) ou através da adição de uma matriz semi-definida positiva à matriz  $P(t)$  (busca aleatória).

### 2.5.1.1 Regulagem do fator de esquecimento

Dentre os vários métodos para a regulagem do fator de esquecimento [23], utiliza-se o algoritmo a traço limitado porque este algoritmo calcula o fator de esquecimento, a partir de especificações sobre o traço da matriz  $P(t)$ , que se deseja limitar.

Neste método  $\lambda(t)$  é calculado para manter o traço da matriz de covariância  $P(t)$  superior a uma constante  $tro$  definida pelo usuário.

Definindo

$$\tilde{P}(t) = P(t-1) - \frac{P(t-1) \varphi(t) \varphi^T(t) P(t-1)}{1/\mu(t) + \varphi^T(t) P(t-1) \varphi(t)} \quad (2-18)$$

e calculando o traço de  $\tilde{P}(t)$ , têm-se o seguinte procedimento:

i) Se traço  $[\tilde{P}(t)] < tro$  então o fator de esquecimento é dado por

$$\lambda(t) = \text{traço} [\tilde{P}(t)] / tro \quad (2-19)$$

substituindo  $\lambda(t)$  e  $\tilde{P}(t)$  na equação 2-9 dos mínimos quadrados, tem-se

$$P(t) = \tilde{P}(t) / \lambda(t) \quad (2-20)$$

ii) Se traço  $[\tilde{P}(t)] \geq tro$  então

$$\lambda(t) = 1 \text{ e } P(t) = \tilde{P}(t)$$

Escolhe-se o traço mínimo ( $tro$ ) de modo que em regime, o estimador apresente pequena variância em torno do valor estimado. Esta variância deve permitir uma adaptação a variações lentas nos parâmetros a serem identificados. No

capítulo 3 são apresentados exemplos da escolha do traço mínimo tro.

### 2.5.1.2 Busca aleatória

Neste procedimento adiciona-se à matriz de covariância uma matriz constante como indicado abaixo

$$P(t) = \tilde{P}(t) + Q(t) \quad (2-21)$$

onde  $\tilde{P}(t)$  é dada pela equação 2-18.

Para a escolha da matriz  $Q(t)$  analisa-se o estimador como um filtro de Kalman. Neste caso considera-se que as equações dinâmicas dos parâmetros do sistema incorporam um ruído  $n(t)$  com matriz de covariância  $Q(t)$  que permite modelar as variações dos parâmetros, isto é:

$$\theta(t+1) = \theta(t) + n(t) \quad (2-22)$$

Reescrevendo a equação 2-3 das medidas como segue

$$y(t) = \varphi^T(t) \theta(t) + v(t) \quad (2-3)$$

onde  $n(t)$  e  $v(t)$  são seqüências de variáveis aleatórias independentes e brancas, com média nula e matriz de covariância  $Q(t)$  e  $\mu(t)$  respectivamente, as equações do filtro de Kalman para o modelo dado pelas equações acima (2-22 e 2-3) são:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + G(t) [ y(t) - \varphi^T(t) \hat{\theta}(t-1) ] \quad (2-23)$$

$$G(t) = P(t) \varphi(t) = \frac{P(t-1) \varphi(t)}{1/\mu(t) + \varphi^T(t) P(t-1) \varphi(t)} \quad (2-24)$$

$$P(t) = \left[ P(t-1) - \frac{P(t-1) \varphi(t) \varphi^T(t) P(t-1)}{1/\mu(t) + \varphi^T(t) P(t-1) \varphi(t)} \right] + Q(t) \quad (2-25)$$

As equações acima são as mesmas equações do estimador dos mínimos quadrados ponderados recursivo (2-7 e 2-8) exceto pela matriz de covariância que passa a ser descrita pela equação 2-25.

Desta forma, os elementos da matriz  $P(t)$  e o ganho  $G(t)$  podem não tender mais a zero através de uma escolha apropriada dos elementos da matriz  $Q(t)$ . Este método é muito mais seletivo que a utilização do fator de esquecimento, pois permite aumentar separadamente os elementos individuais da matriz  $P(t)$ , através de uma escolha adequada da matriz  $Q(t)$ .

Como no caso anterior, os elementos da matriz  $Q(t)$  devem ser escolhidos de modo que não se deteriore significativamente a convergência do estimador, e que este apresente ainda uma capacidade de adaptação a variações lentas nos parâmetros a serem identificados. Caso ocorram variações bruscas no processo é necessário incluir, nos algoritmos apresentados, outros procedimentos que serão discutidos na seção seguinte.

### 2.5.2. Métodos de detecção de não-estacionaridades do modelo

Quando ocorrem mudanças bruscas nos parâmetros do sistema é necessário aumentar a capacidade de adaptação do algoritmo de identificação, para que os novos valores dos estimadores sejam atingidos rapidamente. Esta condição é obtida aumentando-se bruscamente, o ganho de correção  $G(t)$ . Entretanto, como não se conhece o instante de variação dos parâmetros deve-se associar ao algoritmo de identificação procedimentos que detetem a não-estacionaridade e determinem o nível de adaptação necessário para rastreá-la.

Os métodos de detecção de não-estacionaridade utilizados neste trabalho, baseiam-se na comparação a cada instante de amostragem, dos valores novos e passados de um determinado critério. Os critérios propostos são baseados no erro previsto e nos parâmetros estimados [23].

#### 2.5.2.1 Critério do erro previsto

Este critério é baseado na comparação entre a variância do erro previsto e a variância da perturbação  $v(t)$  existente no modelo do processo.

Já foi mostrado que o erro previsto é descrito por

$$\varepsilon_p(t) = y(t) - \varphi^T(t) \hat{\theta}(t-1) \quad (2-16)$$

A substituição da equação das medidas (2-3) na equação do erro previsto, reescrita acima, resulta em

$$\varepsilon_p(t, \hat{\theta}) = \varphi^T(t) [ \theta(t-1) - \hat{\theta}(t-1) ] + v(t) \quad (2-26)$$

Na equação 2-26 verifica-se que se o estimador  $\hat{\theta}(t)$  converge para o valor verdadeiro  $\theta(t)$ , o erro previsto  $\varepsilon_p(t)$  tende para o valor da perturbação no processo  $v(t)$ . Então em regime, se  $\hat{\theta}(t)$  converge para  $\theta(t)$ , o erro previsto é uma estimativa do ruído no processo.

Da equação 2-26 verifica-se também que, se o processo sofre uma não-estacionaridade abrupta, o valor do erro previsto varia rapidamente, e esta variação é proporcional à amplitude da variação do parâmetro.

Portanto, a informação contida nas variâncias do erro previsto e da perturbação permite detetar quando ocorre uma não-estacionaridade. Contudo como os valores destas variâncias

não são conhecidos é necessário obter estimativas destes valores.

A estimativa da variância do erro previsto é obtida a partir dos últimos valores calculados deste erro e é realizada sobre um intervalo de tempo (horizonte) pequeno de acordo com:

$$\sigma_c^2(t) = \frac{1}{N_c} \sum_{\tau=t-N_c+1}^t \varepsilon_P^2(\tau, \hat{\theta}) \quad (2-27)$$

onde  $N_c$  é denominado horizonte curto.

A estimativa da variância da perturbação é obtida supondo-se que nos instantes de tempo anteriores a  $(t-N_c)$ , o sistema estava em regime, e portanto a variância do erro previsto representa uma estimativa da variância da perturbação. Deste modo, a estimativa da variância da perturbação é realizada sobre um intervalo de tempo longo e anterior a  $(t-N_c)$ , de acordo com:

$$\sigma_l^2(t) = \frac{1}{N_l} \sum_{\tau=t-N_l-N_c+1}^{t-N_c} \varepsilon_P^2(\tau, \hat{\theta}) \quad (2-28)$$

onde  $N_l$  é denominado horizonte longo.

O critério de detecção é baseado na relação entre as estimativas da variância do erro previsto e da variância da perturbação, isto é

$$J(t) = \sigma_c^2(t) / \sigma_l^2(t) \quad (2-29)$$

que representa a "probabilidade" de modificação do modelo.

Neste trabalho utiliza-se o critério do erro previsto e o método do traço limitado para a identificação de sistemas variantes no tempo [07]. Este algoritmo denominado traço adaptativo realiza a cada iteração, as seguintes etapas:

1. Estimativa das variâncias e cálculo do critério de detecção através das equações 2-27, 2-28 e 2-29.

$$\sigma_c^2(t) = \frac{1}{N_c} \sum_{\tau=t-N_c+1}^t \varepsilon_p^2(\tau, \hat{\theta}) \quad (2-27)$$

$$\sigma_l^2(t) = \frac{1}{N_l} \sum_{\tau=t-N_l-N_c+1}^{t-N_c} \varepsilon_p^2(\tau, \hat{\theta}) \quad (2-28)$$

$$J(t) = \sigma_c^2(t) / \sigma_l^2(t) \quad (2-29)$$

2. Fase de decisão e cálculo do traço mínimo da matriz de covariância

i) Se  $0 \leq J(t) < J_{\min}(t)$  onde  $J_{\min}(t)$  é o limite mínimo para a detecção, então não ocorreu detecção e utiliza-se um traço de regulagem mínimo.

$$tro = tr_{\min}$$

ii) Se  $J_{\min}(t) \leq J(t) < J_{\max}(t)$  onde  $J_{\max}(t)$  é o limite máximo para a detecção então, a detecção é incerta e utiliza-se um traço de regulagem intermediário.

$$tro = tr_{\text{int}}$$

iii) Se  $J(t) \geq J_{\max}(t)$  então ocorreu uma detecção e utiliza-se um traço elevado.

$$tro = tr_{\max}$$

3. Fase de estimação dos parâmetros

Aplica-se o algoritmo de traço limitado segundo as equações 2-7, 2-8 e 2-20 utilizando como constante "tro" de regulagem os valores obtidos na fase anterior.

A implementação do algoritmo do traço adaptativo como foi descrito acima, em especial na sua fase de decisão, deve considerar os seguintes aspectos:

i) O teste de detecção deve ser aplicado pela primeira vez após a convergência dos parâmetros. Na ausência de um teste de convergência aplica-se a detecção somente após um tempo igual ou superior ao tempo de estabilização do estimador. A figura 2.1 ilustra o comportamento de um critério de detecção  $J(t)$  durante 100 iterações de um processo que sofre uma não-estacionaridade em  $t = 50$ . Neste caso o tempo de estabilização é fixado igual a 25 e está representado na figura 2.1 pelo intervalo de tempo A.

ii) O nível de adaptação máximo ( $tr_{max}$ ) só é aplicado após um determinado número de iterações de validação. Este número de iterações de validação é utilizado para se evitar detecções falsas devido às variações aleatórias que possam ocorrer no processo. Na figura 2.1 o número de iterações de validação é representado por B e é igual a 2.

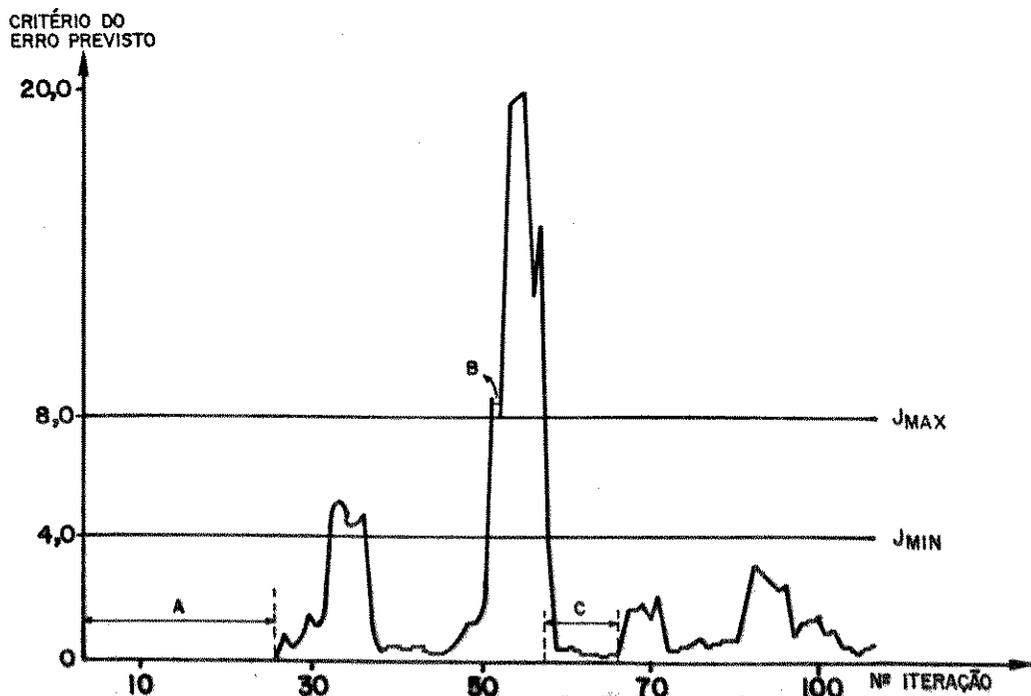


FIGURA 2.1

Comportamento do critério do erro previsto quando acontece uma não-estacionaridade

iii) Após uma detecção, se o critério decresce abaixo do limite mínimo, ele não é mais calculado até que se ultrapasse um tempo de estabilização. Este tempo representa o período transitório que segue uma não-estacionaridade e é mostrado em C na figura 2.1. Este procedimento evita a ocorrência de fortes oscilações na estimação logo após a detecção de não-estacionaridades.

#### 2.5.2.2. Critério dos parâmetros estimados

Este critério baseia-se na comparação entre valores recentes do estimador e valores obtidos em um regime permanente anterior ao instante atual e é calculado para cada componente do vetor de estimadores.

O critério dos parâmetros estimados para a  $i$ -ésima componente do vetor  $\hat{\theta}(t)$  é descrito por

$$J_i(t) = \frac{\frac{1}{N_1} \left| \sum_{u=t-N_1+1}^t (\hat{\theta}_i(u) - \bar{\theta}_i(u)) \right|}{\left| \hat{\theta}_i(t) \right|} = \frac{K_i(t)}{\left| \bar{\theta}_i(t) \right|} \quad (2-30)$$

onde  $\bar{\theta}_i$  representa a média sobre os valores estimados do parâmetro  $i$  em regime e é dado por

$$\bar{\theta}_i(t) = \frac{1}{p} \sum_{u=t-N_2+1}^{t-N_1-\tau} \hat{\theta}_i(u) \quad (2-31)$$

com  $p = N_2 - N_1 - \tau$  e os horizontes  $N_1$ ,  $N_2$ ,  $\tau$  e  $p$  são, respectivamente, o intervalo de tempo para cálculo de  $K_i(t)$  (horizonte curto), o intervalo total para cálculo do critério  $J_i(t)$ , o retardo introduzido para o cálculo de  $\bar{\theta}_i(t)$  e o intervalo de tempo para o cálculo de  $\hat{\theta}_i(t)$ . A figura 2.2 ilustra estes intervalos de tempo.

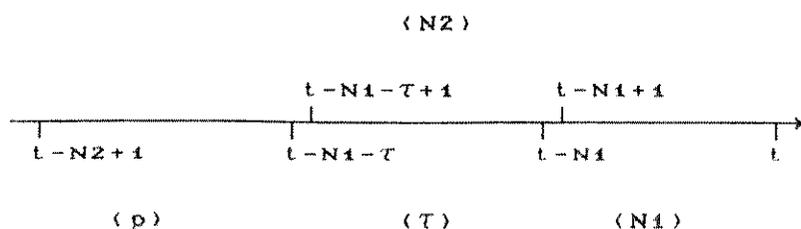


FIGURA 2.2

Horizontes para cálculo do critério

Utiliza-se o retardo de tempo  $\tau$ , para que não ocorra interação entre os estimadores de horizonte curto e longo. Esta estratégia permite aumentar significativamente o critério  $J_i(t)$  quando ocorre variações no parâmetro "i".

Neste trabalho o critério dos parâmetros é associado ao algoritmo de busca aleatória, que permite controlar a componente do ganho que corresponde ao parâmetro que varia no tempo por meio de uma escolha adequada dos elementos da matriz  $Q(t)$ . Este algoritmo é denominado Filtro de Kalman Adaptativo e é utilizado para a identificação de sistemas variantes no tempo.

O algoritmo do Filtro de Kalman Adaptativo realiza a cada iteração as seguintes etapas:

1. Fase de cálculo dos critérios

Calcular  $J_i(t)$  para todo  $i \in [1, n]$ , a partir das equações 2-30 e 2-31.

2. Fase de decisão e cálculo da matriz  $Q(t)$

i) Se  $J_i(t) < J_{imax}(t)$  para todo  $i \in [1, n]$ , onde  $J_{imax}(t)$  é o limite máximo para a detecção, então não ocorreu detecção e utiliza-se o nível de adaptação mínimo

$$q_i(t) = q_{i \min}(t)$$

onde  $q_i(t)$  corresponde ao  $i$ -ésimo elemento da matriz  $Q(t)$  que será adicionada a  $P(t)$  na etapa seguinte.

ii) Se  $\exists i \in [1, n]$  tal que  $J_i(t) \geq J_{\max}(t)$ , foi detectada uma variação e aplica-se o seguinte procedimento de regulagem (fase de regulagem):

a - Na primeira iteração após a validação, faz-se

$$q_i(t) = q_{i,\max}(t)$$

onde  $q_{i,\max}(t)$  é o nível de adaptação máximo.

b - Nas iterações seguintes à aplicação do nível máximo e enquanto  $J_i(t) > J_{\min}(t)$  aplica-se uma adaptação mais fraca

$$q_i(t) = q_{i,\text{int}}(t)$$

### 3. Fase de estimação de parâmetros

Aplica-se o algoritmo do Filtro de Kalman Adaptativo segundo as equações 2-23, 2-24 e 2-25 utilizando como elementos da matriz  $Q(t)$  os valores obtidos para  $q_i(t)$  na fase 2, onde

$$Q(t) = \begin{bmatrix} q_1(t) & & & \\ & \ddots & & \\ & & q_i(t) & \\ & & & \ddots \\ & & & & q_n(t) \end{bmatrix}$$

Como no critério do erro previsto, uma nova detecção só é autorizada após a fase de estabilização do algoritmo de estimação, isto é, assim que  $J_i(t) \leq J_{\min}(t)$  [item ii-b da fase de regulagem] espera-se um tempo de estabilização antes de se utilizar novamente o teste de detecção.

A figura 2.3 ilustra os níveis de adaptação que devem ser utilizados em cada fase do algoritmo do Filtro de Kalman

Adaptativo: enquanto o estimador está em regime utiliza-se  $q_{i,\min}$ , quando ocorre uma detecção utiliza-se  $q_{i,\max}$  na iteração de detecção e  $q_{i,\text{int}}$  durante o restante da fase de decisão. A fase de convergência inicia-se quando  $J_i(t) \leq J_{i,\min}(t)$  e utiliza-se  $q_{i,\min}$  como nível de adaptação. Decorrido o tempo de estabilização volta-se ao regime onde permite-se novamente uma detecção.

Para a utilização dos algoritmos de traço adaptativo e filtro de Kalman adaptativo o usuário deve fornecer os seguintes parâmetros de regulagem: níveis de adaptação ( $q_{i,\min}$ ,  $q_{i,\text{int}}$ ,  $q_{i,\max}$ ), limites de detecção ( $J_{i,\min}$ ,  $J_{i,\max}$  ou  $J_{i,\min}$ ,  $J_{i,\max}$ ) e horizontes de cálculo ( $N_1$ ,  $N_2$  ou  $N_1$ ,  $N_2$  e  $\tau$ ).

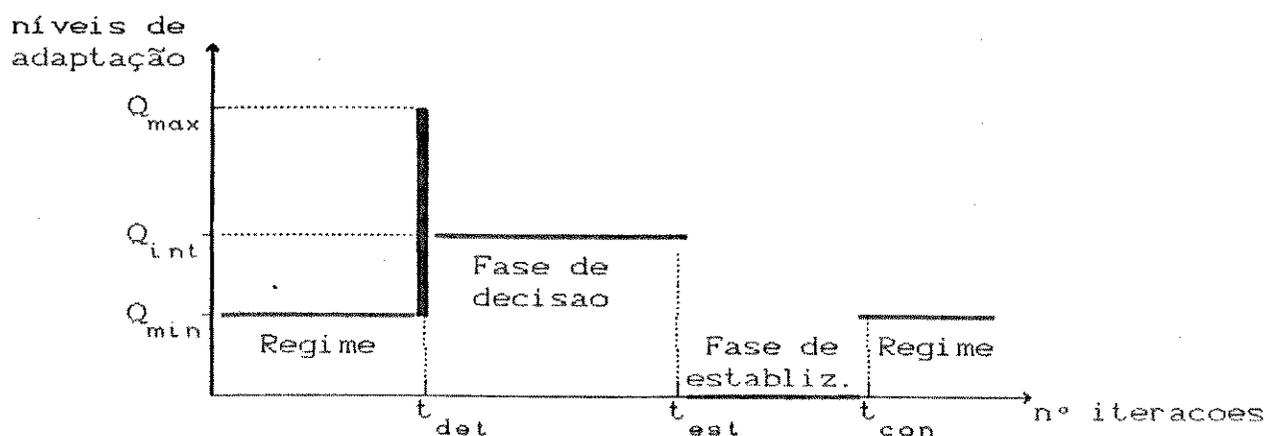


FIGURA 2.3

Regulagem de  $P(t) = \tilde{P}(t) + Q(t)$  para o Algoritmo do Filtro de Kalman Adaptativo

Uma análise da influência destes parâmetros de regulagem no desempenho dos algoritmos é realizada no capítulo 3.

Quando a perturbação no processo não é um ruído branco, utiliza-se o método da variável instrumental como descrito na seção 2.4. Neste caso, os algoritmos adaptativos durante a fase de estimação dos parâmetros, utilizam as equações 2-13, 2-14 e 2-15 para cálculo de  $\hat{\theta}(t)$ ,  $G(t)$  e  $\tilde{P}(t)$  respectivamente.

## 2.6 ALGORITMOS FATORIZADOS

A implementação de um algoritmo de estimação paramétrica em microprocessadores para ser bem sucedida, exige que o algoritmo seja robusto o suficiente para contornar os problemas surgidos da limitação do microprocessador utilizado.

Os problemas relacionados às limitações do microprocessador utilizado são: precisão numérica, pois a matriz de covariância na equação 2-9 é calculada através da subtração de duas matrizes semi-definidas positiva e problemas numéricos devido ao mal condicionamento da matriz  $P(t)$ .

Para evitar esta instabilidade numérica e melhorar a confiabilidade dos algoritmos utiliza-se na literatura [05, 13], as seguintes técnicas de fatorização:

### i) Fatorização do tipo raiz-quadrada

Neste tipo de fatorização a matriz  $P(t)$  é obtida como um produto de matrizes. Desde que esta matriz é definida positiva ela pode ser decomposta de acordo com:

$$P(t) = R(t) * R^T(t) \quad (2-32)$$

onde  $R(t)$  é uma matriz não-singular. A equação 2-9 é então substituída por outra que calcula  $R(t)$  a partir de  $R(t-1)$ .

### ii) Fatorização do tipo U-D

Neste tipo de fatorização a matriz  $P(t)$  é decomposta como

$$P(t) = U(t) * D(t) * U^T(t) \quad (2-33)$$

onde  $U(t)$  é uma matriz triangular superior com todos os elementos da diagonal unitário ( $U(t) \in \mathbb{R}^{n \times n}$ ),  $D(t)$  é uma matriz diagonal

$P(k) \in \mathbb{R}^{n \times n}$  e  $n$  é a ordem do vetor  $\hat{\theta}$ .

Os algoritmos fatorizados descritos acima garantem que a matriz  $P(k)$  permanece definida positiva, têm boa estabilidade numérica e eliminam os erros de arredondamento [05]. O algoritmo U-D requer um menor esforço de cálculo que a fatorização do tipo raiz-quadrada [05] e por isso é utilizado nos algoritmos de identificação desenvolvidos.

### 2.6.1. Algoritmo dos mínimos quadrados com fatorização U-D

A dedução do algoritmo de fatorização U-D é apresentada no apêndice A. O estimador dos mínimos quadrados utilizando esta fatorização executa os seguintes passos:

1. Inicializar  $U(0)$ ,  $D(0)$  em  $t=0$ , tal que  $P(0) = U(0) * D(0) * U^T(0)$ .

2. Em  $t=k$ , calcular  $G(k)$  e  $\hat{\theta}(k)$  e atualizar  $U(k-1)$  e  $D(k-1)$  realizando os passos 2.1 a 2.6.

2.1 Definir e calcular os vetores intermediários:

$$f(k) = U^T(k-1) * \rho(k)$$

$$h(k) = D(k-1) * f(k)$$

$$s_0(k) = 1/\lambda(k)$$

$$c_1(k) = 0.0$$

2.2 Para  $i = 1$  até  $n$  vá do passo 2.3 ao 2.4:

2.3 Calcular

$$s_i(k) = s_0(k) + h_i(k) * h_i(k) / d_i(k-1)$$

$$d_i(k) = d_i(k-1) * s_{i-1}(k) / s_i(k)$$

$$\beta_i(k) = -f_i(k) / s_{i-1}(k)$$

2.4 Para  $j = 1$  até  $n$  faça

$$u_{ij}(k) = u_{ij}(k-1) + \beta_i(k) * c_{ij}(k)$$

$$c_{j+1}(k) = c_{ji}(k) + h_i(k) * u_{ij}(k-1)$$

## 2.5 Calcular o ganho $G(k)$

$$G(k) = c_{i_{n+1}}(k) / s_n(k)$$

## 2.6 Calcular os parâmetros estimados

onde

$$f^T(k) = [f_1(k) \quad f_2(k) \quad \dots \quad f_i(k) \quad \dots \quad f_n(k)]$$

$$h^T(k) = [h_1(k) \quad h_2(k) \quad \dots \quad h_i(k) \quad \dots \quad h_n(k)]$$

$$s^T(k) = [s_1(k) \quad s_2(k) \quad \dots \quad s_i(k) \quad \dots \quad s_n(k)]$$

e

$$U(k) = [U_1(k) \quad U_2(k) \quad \dots \quad U_i(k) \quad \dots \quad U_n(k)], \quad U(k) \in \mathbb{R}^{n \times n} \text{ com}$$

$$U_1^T(k) = [1 \quad 0 \quad \dots \quad 0 \quad \dots \quad 0]$$

$$U_2^T(k) = [u_{12} \quad 1 \quad 0 \quad \dots \quad 0 \quad \dots \quad 0]$$

$$U_i^T(k) = [u_{1i} \quad u_{2i} \quad \dots \quad u_{i-1i} \quad 1 \quad 0 \quad \dots \quad 0]$$

$$U_n^T(k) = [u_{1n} \quad u_{2n} \quad \dots \quad u_{in} \quad \dots \quad U_{n-1n} \quad 1]$$

e

$$D(k) = [D_1(k) \quad D_2(k) \quad \dots \quad D_i(k) \quad \dots \quad D_n(k)], \quad D(k) \in \mathbb{R}^{n \times n} \text{ com}$$

$$D_1^T(k) = [d_1(k) \quad 0 \quad \dots \quad 0 \quad \dots \quad 0]$$

$$D_i^T(k) = [0 \quad 0 \quad \dots \quad d_i(k) \quad \dots \quad 0]$$

$$D_n^T(k) = [0 \quad 0 \quad \dots \quad 0 \quad \dots \quad d_n(k)]$$

e

$$C(k) = [c_1(k) \quad c_2(k) \quad \dots \quad c_i(k) \quad \dots \quad c_{n+1}(k)], \quad C(k) \in \mathbb{R}^{n \times n+1} \text{ com}$$

$$C_i^T(k) = [c_{1i}(k) \quad c_{2i}(k) \quad \dots \quad c_{ii}(k) \quad \dots \quad c_{ni}(k)]$$

$$C_{n+1}^T(k) = [c_{1n+1}(k) \quad c_{2n+1}(k) \quad \dots \quad c_{in+1}(k) \quad \dots \quad c_{nn+1}(k)]$$

### 2.6.2 Algoritmo de busca aleatória com fatorização U-D

Nos algoritmos de busca aleatória para ajuste da matriz de covariância  $[P(t) = \tilde{P}(t) + Q(t)]$  é necessário que a matriz  $Q(t)$  seja fatorada como

$$Q(t) = V(t) * V^T(t) \quad (2-34)$$

onde  $V(t)$  é uma matriz cujo posto é o número de elementos não nulos ( $s$ ) da diagonal de  $Q(t)$  e  $V(t) \in \mathbb{R}^{n \times s}$ .

A matriz  $P(t)$  é então decomposta como na equação abaixo

$$P(t) = U(t) * D(t) * U^T(t) = \tilde{U}(t) * \tilde{D}(t) * \tilde{U}^T(t) + V(t) * V^T(t) \quad (2-35)$$

onde  $U(t)$ ,  $\tilde{U}(t)$  são matrizes triangulares superior, com todos os elementos da diagonal unitários ( $U(t)$ ,  $\tilde{U}(t) \in \mathbb{R}^{n \times n}$ ),  $D(t)$  e  $\tilde{D}(t)$  são matrizes diagonais ( $D(t)$ ,  $\tilde{D}(t) \in \mathbb{R}^{n \times n}$ ).

A dedução do algoritmo de fatorização U-D segundo a equação 2-35 é apresentada no apêndice B e o algoritmo dos mínimos quadrados ponderados recursivo com busca aleatória, usando fatorização U-D segue os passos abaixo:

1. Em  $t = k-1$  são dados  $U(k-1)$ ,  $D(k-1)$  e  $Q(k) = V(t) * V^T(t)$ .

2. Em  $t = k$ , calcular o ganho  $G(k)$ , as matrizes  $U(k)$  e  $D(k)$  e o estimador  $\hat{\theta}(k)$ , realizando os passos 2.1 a 2.6 dados no item 2.6.1 anterior ( $\tilde{U}(k)$  e  $\tilde{D}(k)$  são  $U(k)$  e  $D(k)$  no algoritmo anterior) e atualizar  $U(k-1)$  e  $D(k-1)$  realizando os passos 2.1 a 2.6 a seguir.

2.1 Formar a matriz  $W(k)$  onde a  $i$ -ésima coluna ( $W_i$ ) é constituída pela  $i$ -ésima coluna de  $U^T(k)$  empilhada sobre a  $i$ -ésima coluna de  $V^T(k)$  com  $i=1, 2, \dots, d$  onde  $d = n+s$ .

2.2 Formar a matriz diagonal  $D(k)$  ( $D(k) \in \mathbb{R}^{d \times d}$ ) como uma matriz bloco-diagonal constituída por  $\tilde{D}(k)$  e a identidade  $I_{s \times s}$ .

2.3 Para  $j = d, d-1, \dots, 2$  vá do passo 2.4 ao 2.5

2.4 Calcular

$$D_{jj}(k) = W_j^T(k) * D(k-1) * W_j(k)$$

2.5 Para  $i = 1$  até  $j-1$  calcular

$$u_{ij}(k) = W_i^T(k) * D(k-1) * W_j(k) / D_{jj}(k)$$

$$W_i(k) = W_i(k) - u_{ij}(k) * W_j(k)$$

## 2.6 Calcular

$$D_{11}(k) = W1^T(k) * D(k-1) * W1(k)$$

## 2.7 CONCLUSÕES

Neste capítulo foi analisado o problema de identificação de sistemas não-estacionários. Os diferentes algoritmos para estimação deste tipo de sistemas foram obtidos a partir do algoritmo dos mínimos quadrados, limitando-se os elementos da matriz de covariância  $P(t)$ .

A utilização de um fator de esquecimento variável, calculado a partir de especificações sobre o traço da matriz  $P(t)$  que neste método, é limitado a um valor constante  $tro$ , é um procedimento eficiente para o rastreamento de variações lentas no tempo dos parâmetros do processo. O algoritmo dos mínimos quadrados usando este tipo de fator de esquecimento, denominado algoritmo do traço limitado, permite ao estimador, mesmo em regime, uma capacidade de adaptação quando ocorrem variações nos parâmetros do processo.

Outro procedimento também eficiente no rastreamento de variações lentas é o de busca aleatória, que adiciona uma matriz de covariância de estado  $Q(t)$  à matriz  $P(t)$ . Este algoritmo, dos mínimos quadrados com busca aleatória, é mais seletivo que o algoritmo do traço limitado, pois consegue-se através de uma escolha apropriada dos elementos de  $Q(t)$ , regular cada componente do vetor ganho  $G(t)$  do estimador.

O algoritmo do traço limitado e o algoritmo de busca aleatória também podem ser utilizados para a estimação de sistemas com variações bruscas, quando o instante de tempo em que ocorrem as variações é conhecido. Neste caso, após o instante de variação, os valores de  $tro$  e  $Q(t)$  são escolhidos de modo que o

estimador consiga acompanhar as não-estacionaridades.

Para a identificação de sistemas com variações bruscas, quando não se conhece os instantes de variação, foi proposta a utilização de métodos de detecção de não-estacionaridades que, aliados aos algoritmos para rastrear variações lentas no tempo, permitem controlar a capacidade de adaptação do estimador. Estes algoritmos são denominados algoritmos adaptativos.

Foram desenvolvidos dois métodos de detecção de não-estacionaridade: i) o critério de detecção baseado em estimativas da variância do erro previsto e ii) o critério de detecção baseado nas médias dos parâmetros estimados.

Finalizando, dois algoritmos adaptativos foram propostos para a identificação de sistemas variantes no tempo: i) o algoritmo do traço adaptativo, que utiliza o estimador com traço limitado e o critério do erro previsto para detecção de não-estacionaridades (ETEP), ii) o filtro de Kalman adaptativo que utiliza o método de busca aleatória para estimar os parâmetros e o critério dos parâmetros estimados como método de detecção (FKPE).

## CAPÍTULO 3

### ANÁLISE DO DESEMPENHO DOS ALGORITMOS ADAPTATIVOS ATRAVÉS DE EXEMPLOS DE SIMULAÇÃO

#### 3.1 INTRODUÇÃO

Neste capítulo apresentam-se resultados obtidos em diversas simulações dos algoritmos de identificação adaptativos desenvolvidos no capítulo 2.

Os algoritmos foram utilizados para identificar vários processos com diferentes tipos de não-estacionaridades e perturbações.

A partir dos resultados obtidos, realiza-se uma análise do desempenho dos algoritmos adaptativos. Inicialmente, esta análise é feita para cada algoritmo em função dos parâmetros de regulagem e dos vários tipos de não-estacionaridades que podem ocorrer nos processos. A seguir realiza-se uma análise comparativa dos algoritmos de identificação para sistemas variantes no tempo, utilizando critérios relacionados com a eficiência na detecção e na convergência do estimador.

Na primeira parte deste capítulo, apresentam-se na seção 3.2 os critérios para comparação dos algoritmos e na seção 3.3 as características das simulações realizadas.

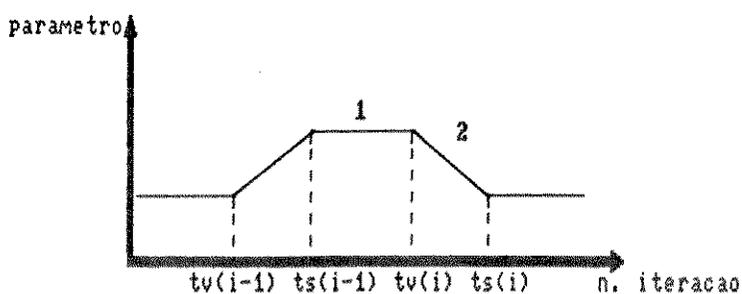
A análise da influência dos parâmetros de regulagem no desempenho dos algoritmos adaptativos é desenvolvida na seção 3.4. Na seção 3.5, condições para o ajuste destes parâmetros são estabelecidas a partir dos resultados obtidos na seção anterior.

Na seção 3.6 realiza-se a análise comparativa dos algoritmos e, por fim, na seção 3.7 ressaltam-se as principais características que devem ser consideradas quando da escolha de um dos algoritmos para identificação de um processo.

### 3.2 CRITÉRIOS PARA COMPARAÇÃO DO DESEMPENHO DOS ALGORITMOS DE ESTIMAÇÃO

#### 3.2.1 Critérios de detecção

As simulações realizadas apresentam várias regiões de estacionaridades e não-estacionaridades. A figura 3.1 ilustra as condições utilizadas para simular estas regiões. Nesta figura define-se região estacionária como o intervalo de tempo compreendido entre uma iteração de estabilização  $ts(i-1)$  e uma iteração de variação  $tv(i)$ ; nesse período o parâmetro apresenta um valor constante. A região de variação é o intervalo de tempo compreendido entre uma iteração de variação  $tv(i)$  e uma iteração de estabilização  $ts(i)$ ; neste período ocorre a não-estacionaridade, isto é, o parâmetro sofre uma variação.



- 1 - região estacionária "1",
- 2 - variação "1" caracterizada pela iteração de variação  $tv(i)$  e iteração de estabilização  $ts(i)$  e pelo valor final do parâmetro.

FIGURA 3.1

Características das não-estacionaridades simuladas

De acordo com os procedimentos apresentados no capítulo anterior, itens 2.5.2.1 e 2.5.2.2, podem ocorrer três condições nos testes de detecção: nenhuma detecção, detecção incerta (para o método do erro previsto) e detecção certa.

A partir destas informações propõem-se a utilização dos seguintes critérios, calculados através de simulações de Monte Carlo, para análise dos testes de detecção.

### 3.2.1.1 Taxas de alarme falso ( TAF )

Um erro é cometido pelos métodos de detecção quando é detetado que houve uma não-estacionaridade e esta de fato não ocorreu. Este tipo de erro é denominado erro de alarme falso e a medida da frequência em que ele ocorre durante a execução do algoritmo é chamada taxa de alarme falso.

Este tipo de erro é medido considerando tanto as detecções certas como as incertas.

Para calcular as taxas de alarme falso certo e incerto mede-se separadamente, as detecções certas e incertas sobre cada região estacionária compreendida entre  $[ tv(i-1) + D_{\max} ]$  e  $tv(i)$ , onde  $D_{\max}$  é o número de iterações permitindo passar o período transitório que segue a variação. O tempo  $D_{\max}$  é utilizado para evitar que as taxas de alarme falso sejam calculadas durante o transitório que segue uma variação. Neste trabalho  $D_{\max}$  é definido como

$$D_{\max} = 10 + 2 ( ts(i) - tv(i) ) \quad (3-1)$$

As taxas de alarme falso certo e de alarme falso incerto são obtidas dividindo-se respectivamente, os números de detecções certas e de detecções incertas, pelo número total de iterações separando as regiões estacionárias. Na contagem das detecções incertas considera-se que uma detecção certa é também considerada incerta.

### 3.2.1.2 Taxas de detecção e de não-deteção ( TD e TND )

Um outro tipo de erro é cometido pelos métodos de detecção quando ocorre uma não-estacionaridade e esta não é detetada. A medida da frequência de ocorrência deste tipo de erro é denominada taxa de não-deteção, e é calculada a partir da taxa de detecção que representa a frequência de acerto do teste, isto é, a frequência com que o método deteta uma não-estacionaridade que de fato ocorreu.

Para se obter as taxas de detecção certa e incerta calculam-se as seguintes razões sobre o conjunto de intervalos de tempo definidos por  $tv(i)$  e  $[ tv(i) + D_{max} ]$ , onde  $D_{max}$  é interpretado agora como sendo o atraso máximo permitido na detecção.

$$\text{taxa de detecção certa} = \frac{\text{número de primeiras detecção certa/região}}{\text{número de variações}}$$

$$\text{taxa de det. incerta} = \frac{\text{número de primeiras detecção incerta/região}}{\text{número de variações}}$$

As taxas de não-deteção certa e incerta são calculadas a partir das taxas de detecção como segue

$$\text{taxa de não-deteção certa} = 1 - \text{taxa de detecção certa}$$

$$\text{taxa de não-deteção incerta} = 1 - \text{taxa de detecção incerta.}$$

### 3.2.1.3 Atraso na detecção

O atraso na detecção é definido como a diferença entre a iteração onde ocorreu a primeira detecção certa e a iteração de variação propriamente dita. É calculado na região compreendida entre os instantes de tempo  $tv(i-1)$  e  $tv(i)$ .

### 3.2.2 Critérios de identificação

Os critérios de identificação procuram analisar o desempenho dos algoritmos em termos de sua convergência para os valores verdadeiros dos parâmetros. Neste trabalho, utiliza-se como critério de identificação as medidas apresentadas a seguir.

#### 3.2.2.1 Polarização dos parâmetros

A polarização  $P$  é calculada de acordo com:

$$P(i,t) = \theta_i(t) - \bar{\theta}_i(t) \quad (3-2)$$

onde  $\bar{\theta}_i(t)$  é a  $i$ -ésima componente de  $\bar{\theta}(t)$  definido pela equação abaixo:

$$\bar{\theta}_i(t) = \alpha \bar{\theta}_i(t-1) + (1 - \alpha) \bar{\theta}_i(t) \quad (3-3)$$

$\alpha$  é um coeficiente que permite obter uma média recursiva sobre um horizonte cujo comprimento é função de  $\alpha$ . Utiliza-se  $\alpha = 0.6$  que fornece um horizonte de 5 a 10 medidas.

#### 3.2.2.2 Variância do erro de estimação

A variância do erro de estimação por parâmetro é calculada pela equação

$$\sigma_i^2(t) = \alpha \sigma_i^2(t-1) + (1-\alpha) \tilde{\theta}_i^2(t) \quad (3-4)$$

onde  $\tilde{\theta}_i(t)$  é a  $i$ -ésima componente do vetor  $\tilde{\theta}(t)$  definido pela equação

$$\tilde{\theta}_i(t) = \hat{\theta}_i(t) - \bar{\theta}_i(t) \quad (3-5)$$

e  $\alpha$  é como descrito no item 3.2.2.1 anterior.

### 3.3 CARACTERÍSTICAS DAS SIMULAÇÕES

#### 3.3.1 Inicialização

Para todos os algoritmos utilizados nos exemplos deste capítulo escolheu-se os valores iniciais dos estimadores iguais a zero e a matriz de covariância inicial  $P(0) = 100 \times I$  onde  $I$  é a matriz identidade de dimensão  $n$ .

#### 3.3.2 Geração das medidas

As medidas  $y(t)$  foram obtidas a partir de um processo descrito por:

$$y(t) = -a_1 y(t-1) + b_0 u(t-2) + e(t) \quad (3-6)$$

onde os seguintes modelos foram analisados:

MOD 1:  $a_1 = -0.905$ ,  $b_0 = 1.0$ ;

MOD 2:  $a_1 = -0.980$ ,  $b_0 = 0.4$ ;

MOD 3:  $a_1 = -0.950$ ,  $b_0 = 0.5$ ;

MOD 4:  $a_1 = -0.905$ ,  $b_0 = 2.0$ .

A entrada  $u(t)$  e a perturbação  $e(t)$  são geradas independentemente. O ruído  $e(t)$  é uma seqüência branca de variáveis aleatórias com média zero e desvio padrão 0.1. A entrada  $u(t)$  é um ruído branco de média zero e variância escolhida de acordo com o valor especificado para a relação sinal-ruído na saída do processo.

Os resultados foram obtidos com simulações de Monte Carlo com médias sobre 10 realizações.

Todas as variações no processo são realizadas no instante  $t = 50$  e são representadas por uma mudança de modelo, isto é, inicialmente o modelo 1 representa o processo e depois do instante  $t = 50$ , o processo passa a ser representado por outro modelo.

De acordo com o tipo do modelo (2, 3 ou 4) que passa a representar o sistema após a variação, classifica-se as não-estacionaridades quanto à amplitude da variação em grande ou pequena. As variações grandes são as transições do modelo 1 para o modelo 2 (variação de 8,3% no parâmetro  $a_1$  e de 60% no parâmetro  $b_0$ ) ou para o modelo 4 (variação de 100% no parâmetro  $b_0$ ), e as variações pequenas são as transições do modelo 1 para o modelo 3 (variação de 5% no parâmetro  $a_1$  e de 50% no parâmetro  $b_0$ ).

Quanto ao intervalo de tempo para ocorrer uma mudança de modelo, as variações são classificadas em rápidas ou lentas. Nas simulações realizadas considera-se variações rápidas, as que ocorrem em apenas uma iteração e considera-se variações lentas as que acontecem em 5 iterações, isto é, em  $t = 50$  o sistema é representado pelo modelo 1, durante as iterações de 50 a 55, o processo modifica-se linearmente até que na iteração  $t = 55$  passe a ser representado por outro modelo.

### 3.4 INFLUÊNCIA DOS PARÂMETROS DE REGULAGEM NOS ALGORITMOS DE ESTIMAÇÃO

O algoritmo do traço adaptativo com o critério do erro previsto e o algoritmo do filtro de kalman adaptativo com o critério dos parâmetros foram utilizados na identificação de um

processo que sofre uma variação rápida ou lenta em  $t = 50$  passando do modelo 1 para o modelo 2.

As simulações para cada algoritmo foram realizadas em duas etapas; primeiro para uma determinada combinação de horizontes e limiares de detecção testaram-se várias combinações de níveis de adaptação. Em seguida para uma combinação de níveis de adaptação e limiares de detecção testaram-se vários conjuntos de horizontes.

Os resultados obtidos e as conclusões alcançadas sobre a influência dos parâmetros de regulação no desempenho dos algoritmos são apresentados a seguir.

#### 3.4.1 Escolha dos níveis de adaptação para o algoritmo do traço adaptativo.

O desempenho do algoritmo do traço adaptativo foi analisado para relações sinal/ruído iguais a 10, 5 e 2 e para os níveis de adaptação descritos na tabela 01.

n-adapt.	$tr_{min}$	$tr_{int}$	$tr_{max}$
COMB1	0.01	0.10	1.0
COMB2	0.05	0.50	5.0
COMB3	0.10	1.00	10.0

TABELA 01  
Níveis de Adaptação

Os horizontes utilizados são  $N_l = 10$ ,  $N_c = 5$  e  $\tau = 0$  e os limiares de detecção são  $J_{min} = 4.0$  e  $J_{max} = 8.0$ .

FIGURAS	características das simulações			POL. a <sub>1</sub>	POL. b <sub>0</sub>	VAR. a <sub>1</sub>	VAR. b <sub>0</sub>	atraso na detec.	TAF certo	TAF incert	TND certa	TND incert
	valor de traço	razão sinal ruído	tipo de var.									
3.6	COMB1	10	RAP.	0.53E-02	-0.15E-02	0.14E-04	0.16E-03	2.0	0.0	0.0194	0.0	0.0
	COMB2			0.64E-02	-0.22E-02	0.72E-04	0.15E-03	2.0	0.0	0.0105	0.0	0.0
	COMB3			0.93E-02	-0.22E-02	0.91E-04	0.24E-03	2.0	0.0	0.0285	0.0	0.0
	COMB1	5	RAP.	0.24E-01	0.90E-01	0.30E-04	0.16E-03	3.0	0.0	0.0106	0.5	0.0
	COMB2			0.19E-01	0.25E-01	0.82E-04	0.20E-03	3.4	0.0	0.0174	0.2	0.0
	COMB3			0.15E-01	0.19E-01	0.14E-03	0.57E-03	3.2	0.0	0.0317	0.3	0.0
	COMB1	2	RAP.	0.77E-01	0.38E-01	0.28E-05	0.83E-04	n-det.	0.0	0.0	1.0	0.3
	COMB2			0.49E-01	0.32	0.42E-04	0.11E-03	n-det.	0.0	0.0	1.0	0.3
	COMB3			0.59E-01	0.27	0.19E-04	0.15E-04	n-det.	0.0	0.0	1.0	0.4
3.2 3.3	COMB1	10	LEN.	0.16E-01	0.70E-01	0.12E-04	0.19E-03	2.5	0.0	0.0079	0.0	0.0
	COMB2			0.69E-02	-0.67E-02	0.11E-04	0.16E-03	2.1	0.0	0.0092	0.0	0.0
	COMB3			0.87E-02	-0.30E-03	0.19E-04	0.19E-03	2.6	0.0	0.0190	0.0	0.0
3.4	COMB1	5	LEN.	0.46E-01	0.27	0.42E-05	0.12E-03	4.0	0.063	0.0190	0.6	0.0
	COMB2			0.18E-01	0.30E-01	0.15E-04	0.18E-03	4.0	0.0079	0.0095	0.6	0.0
	COMB3			0.13E-01	0.42E-02	0.26E-04	0.25E-03	4.0	0.0079	0.0095	0.6	0.0
3.5	COMB1	2	LEN.	0.26E-01	0.36	0.33E-05	0.21E-05	n-det.	0.0	0.0	1.0	0.3
	COMB2			0.73E-01	0.36	0.43E-05	0.11E-04	n-det.	0.0	0.0	1.0	0.3
	COMB3			0.64E-01	0.29	0.78E-05	0.15E-03	n-det.	0.0	0.0	1.0	0.3

TABELA 02

ALGORITMO DO TRAÇO ADAPTATIVO -- CRITÉRIO DO ERRO PREVISTO

MOD 1 → MOD 2

Parametros : Nl = 10      Jmin = 4.0      COMB 1: trmin = 0.01      trint = 0.1      trmax = 1.0  
                  Nc = 5      Jmax = 8.0      COMB 2: trmin = 0.05      trint = 0.5      trmax = 5.0  
                  Tau = 0      COMB 3: trmin = 0.10      trint = 1.0      trmax = 10.0

A tabela 02 apresenta os resultados obtidos em termos dos critérios de detecção e dos critérios de identificação.

Da análise da tabela 02 para o critério do erro previsto verifica-se que

i) Para relações sinal-ruído elevadas (ver figuras 3.2 e 3.3), a polarização do estimador diminui. Os parâmetros estimados apresentam maior variância se os valores limites do traço são aumentados.

ii) Quando se aumenta o nível de ruído ( $s/r = 5$  ou  $s/r = 2$ ), o desempenho do algoritmo de estimação é prejudicado em termos de polarização e variância dos estimadores (ver figuras 3.3, 3.4 e 3.5). Também as taxas de não-deteção são maiores para um nível de ruído mais alto. Em caso de relação sinal-ruído muito baixa ( $s/r = 2$ ) o algoritmo não consegue detetar as variações.

iii) O tipo de variação dos parâmetros influencia o atraso na detecção e a convergência dos estimadores após a não-estacionaridade (ver figuras 3.2 e 3.6). Se a variação for lenta tem-se um atraso na detecção maior que no caso de variação rápida, do mesmo modo a convergência para os novos valores torna-se mais lenta.

iv) As taxas de alarme falso são maiores no caso de variações lentas.

#### 3.4.2 Escolha dos horizontes de cálculo dos critérios para o algoritmo do traço adaptativo.

Usando a combinação COMB1 de valores dos níveis de adaptação, analisa-se a seguir a influência dos horizontes de cálculo de acordo com os valores da tabela 03 a seguir.

horizonte	$N_l$	$N_c$	$\tau$
JAN1	10	10	0
JAN2	10	5	0
JAN3	10	5	5

TABELA 03  
Horizontes de Cálculo

A partir dos figuras 3.6 e 3.7 que ilustram o desempenho do algoritmo do traço adaptativo para JAN1 e JAN2 verifica-se que, se é escolhido um valor elevado para o horizonte curto ( $N_c$ ), ocorre uma atenuação no pico de detecção, aumentando as taxas de não-deteção do estimador. Por outro lado, a introdução de um retardo, como em JAN3, entre os horizontes curto e longo não resulta em melhoras significativas para os casos em que as mudanças são rápidas e a relação sinal-ruído é elevada ( $s/r = 10$ ). Os resultados para JAN3 são semelhantes aos de JAN2, no entanto, para mudanças lentas a introdução do retardo resulta em melhor desempenho na identificação do que o apresentado por JAN2.

#### 3.4.3 Escolha dos níveis de adaptação para o algoritmo do filtro de Kalman adaptativo.

O desempenho do Filtro de Kalman Adaptativo foi analisado para relações sinal-ruído iguais a 10, 5 e 2 e para os níveis de adaptação descritos na tabela 04 a seguir.

Os horizontes utilizados são  $N_2 = 20$ ,  $N_1 = 5$  e  $\tau = 5$  ( $p = 10$ ) e os limiares de detecção são  $J_{min} = 0.05$  e  $J_{max} = 0.2$ .

n-adapt.	$q_{min}$	$q_{int}$	$q_{max}$
COMB1	0.01	0.01	2.0
COMB2	0.0001	0.01	1.0
COMB3	0.10	0.10	5.0

TABELA 04  
Níveis de Adaptação

A tabela 05 apresenta os resultados obtidos com o algoritmo do filtro de Kalman Adaptativo sob estas condições, na identificação dos mesmos processos simulados para a tabela 02.

Da análise da tabela 05 para o critério dos parâmetros conclui-se que:

i) O tipo de variação não influi no desempenho da detecção nem na convergência do estimador ( ver figuras 3.8 e 3.12 ). Na tabela, vê-se que os valores de polarização, variância e atraso são semelhantes, independente da variação ser rápida ou lenta.

ii) O aumento do nível de ruído leva a um maior atraso na detecção, a uma menor fase de regulagem ( a fase de regulagem como definida no item 2.5.2.2, inicia-se quando  $J_i(t) > J_{i,max}(t)$  e perdura enquanto  $J_i(t) > J_{i,min}(t)$  ) e a uma convergência mais lenta ( ver figuras 3.8 e 3.11 ). Nos casos em que o nível de adaptação mínimo é baixo (COMB2) para um nível de ruído alto (  $s/r = 2$  ) a probabilidade de não ocorrer detecção é aumentada ( taxa de não-deteção  $\neq 0$  ).

iii) Para uma mesma relação sinal-ruído, o atraso na detecção é diminuído, se o nível de adaptação mínimo é aumentado ( ver figuras 3.8 e 3.10 ). Um nível de adaptação intermediário alto (COMB3) resulta em uma oscilação acentuada do estimador em torno dos valores reais dos parâmetros, embora neste caso obtenha-se



uma menor fase de regulagem ( ver figuras 3.8 e 3.10 ).

iv) A probabilidade de alarme falso é aumentada se os níveis de adaptação são altos ( COMB3 ).

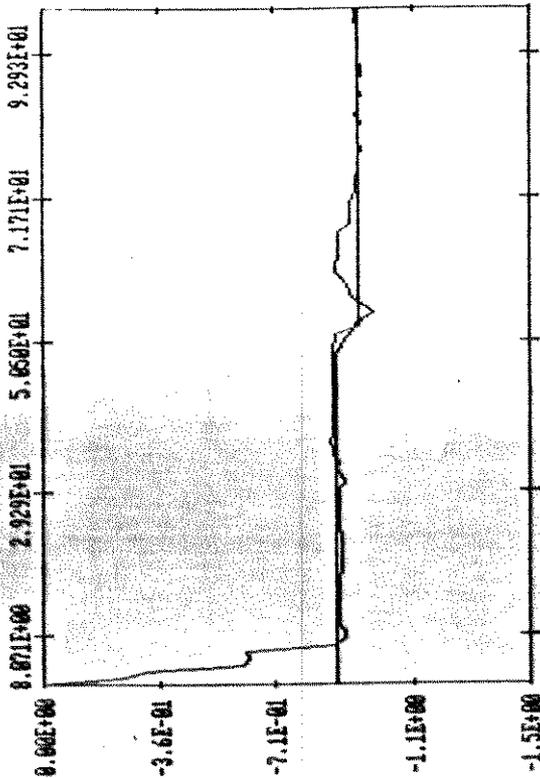
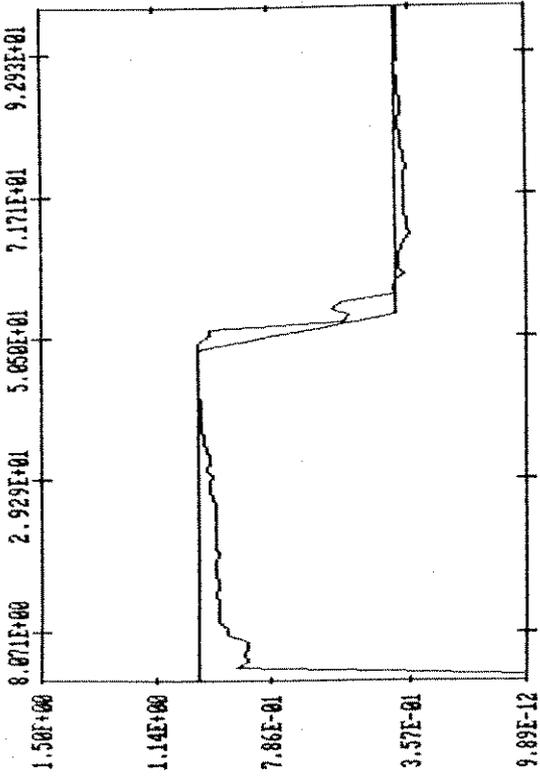
#### 3.4.4 Escolha dos horizontes de cálculo dos critérios para o algoritmo do Filtro de Kalman Adaptativo

Usando a combinação COMB1 de valores de Q, analisa-se a influência dos horizontes  $N_1$ ,  $N_2$  e  $\tau$  sobre o desempenho do algoritmo. Os valores de horizontes de cálculo são descritos na tabela 06 abaixo.

horizonte	$N_2$	$N_1$	$\tau$
JAN1	25	10	5
JAN2	20	5	5
JAN3	15	5	0

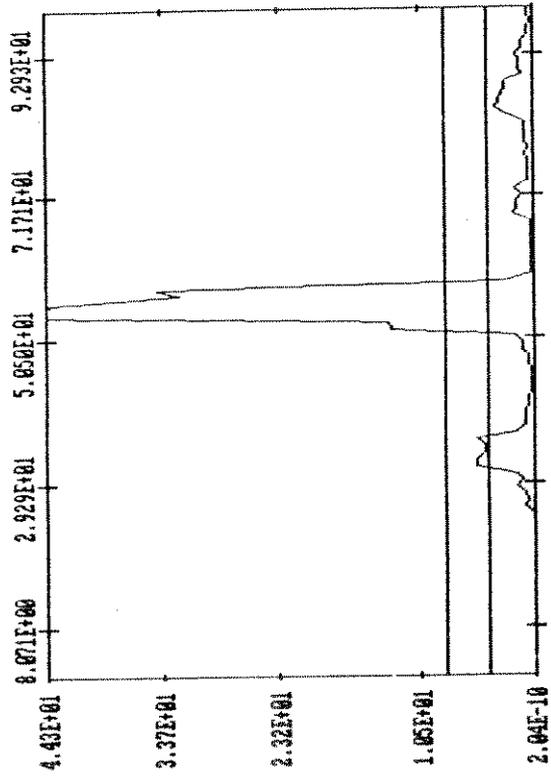
TABELA 06  
Horizontes de Cálculo

A partir das figuras 3.8 e 3.13 que ilustram o comportamento do filtro de Kalman adaptativo para JAN2 e JAN3 respectivamente, verifica-se que a ausência de atraso entre os horizontes curto e longo (  $\tau = 0$  ), resulta em uma atenuação no pico de detecção e em uma menor fase de regulagem, embora os estimadores apresentem uma variância maior em torno dos valores estimados. Para JAN1 a utilização do horizonte curto (  $N_1 = 10$  ) igual ao horizonte longo resulta em um maior atraso na detecção e também, na atenuação do pico de detecção. Neste caso o desempenho do algoritmo é prejudicado por uma maior fase de regulagem e consequentemente, por uma convergência mais lenta do estimador.



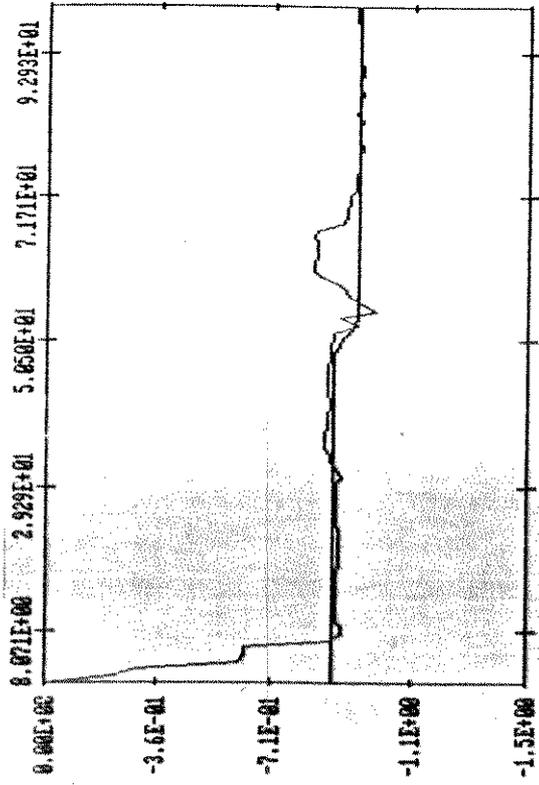
3.2.a - Parametro #1

3.2.b - Parametro #2

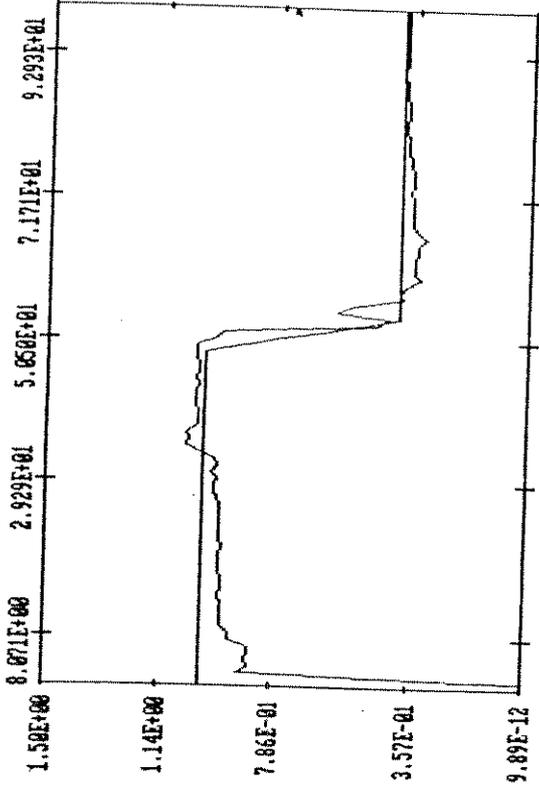


3.2.c - critério do erro previsto

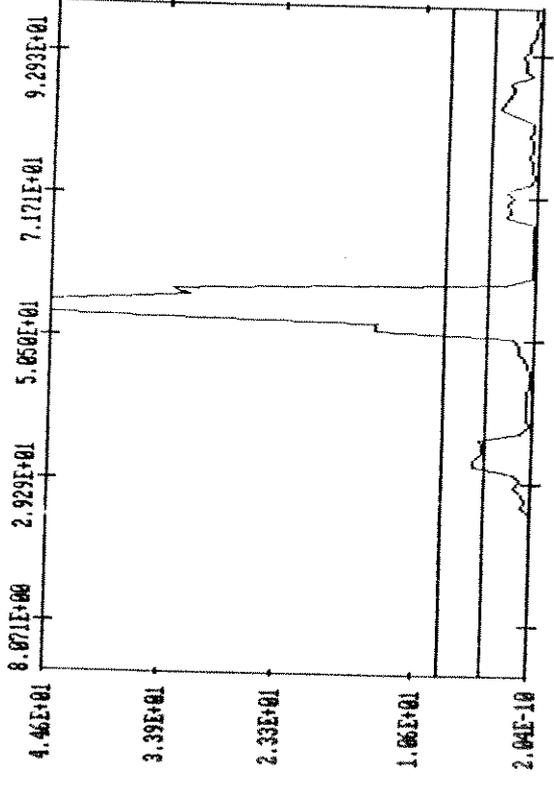
FIGURA 3.2  
 Algoritmo do traço Adaptativo - COMBI - s/r = 10 - lenta  
 mod 1 1/4 mod 2



3.3.a - Parametro a1

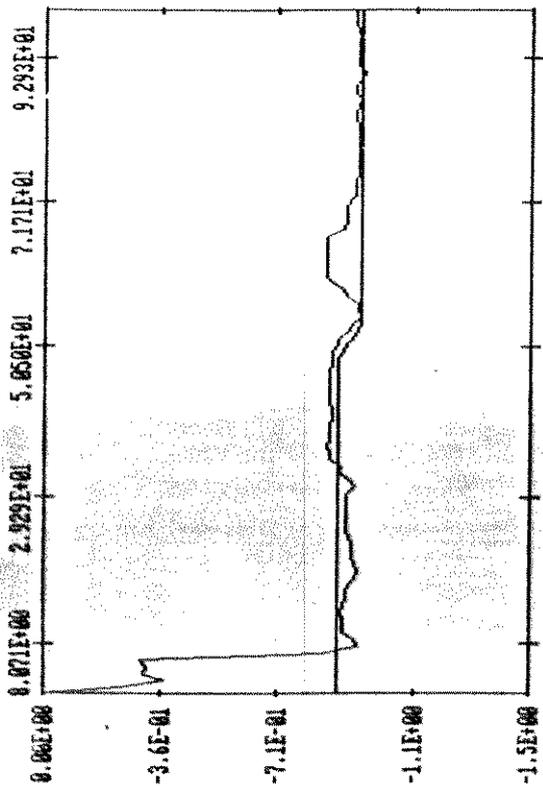


3.3.b - Parametro b0

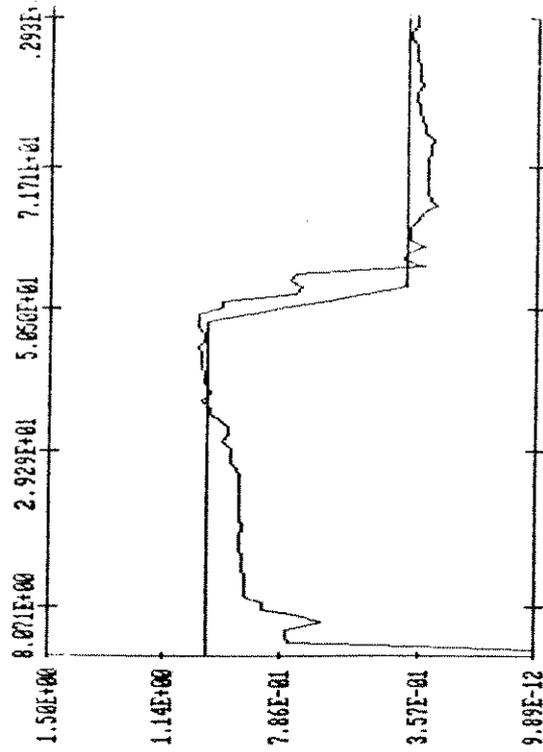


3.3.c - criterio do erro previsto

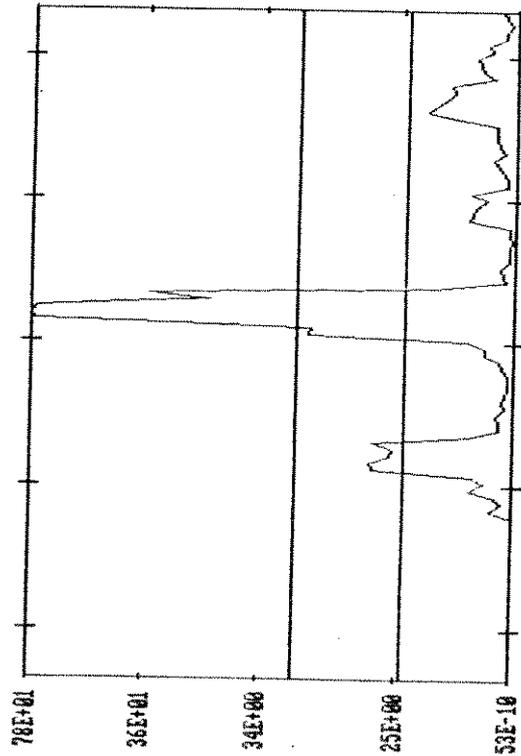
FIGURA 3.3  
 Algoritmo do traço Adaptativo - COMB2 - s/r = 10 - lenta  
 mod 1 34- mod 2



3.4.a - Parametro  $a_1$

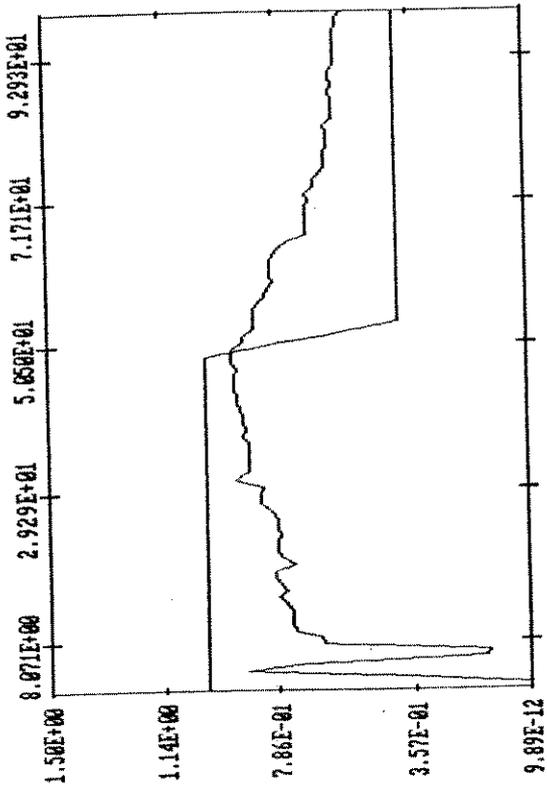


3.4.b - Parametro  $b_0$

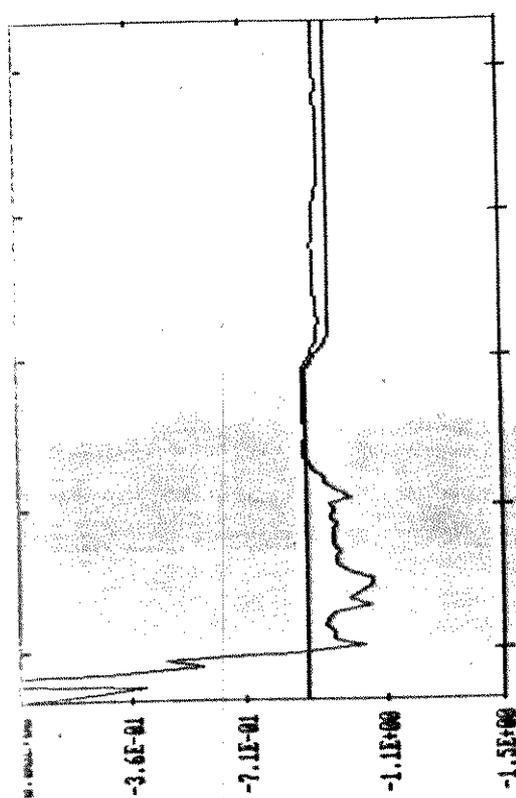


3.4.c - criterio do erro previsto

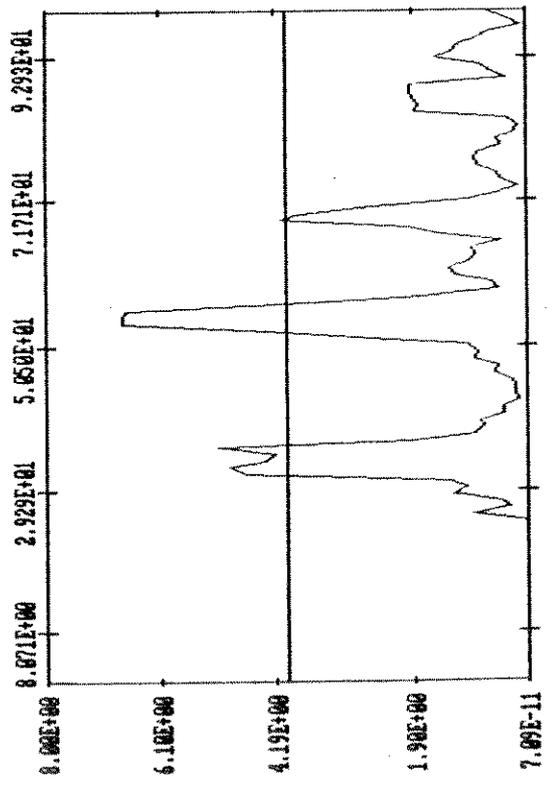
FIGURA 3.4  
Algoritmo do traço Adaptativo - COMB2 -  $\alpha/r = 05$  - lenta  
mod 1



3.5.a - Parametro  $s_1$

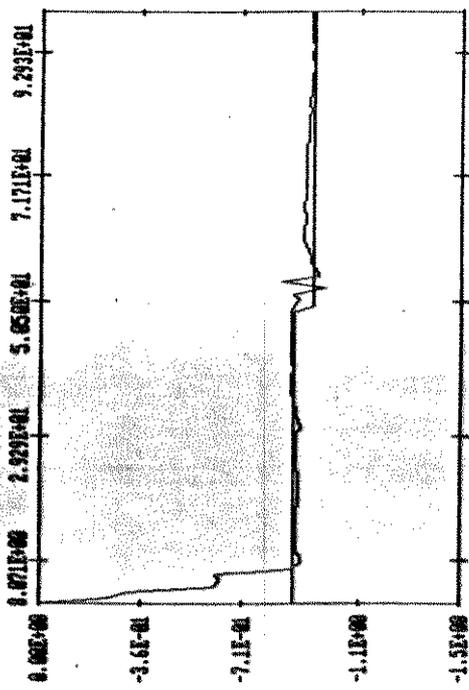


3.5.b - Parametro  $p_0$

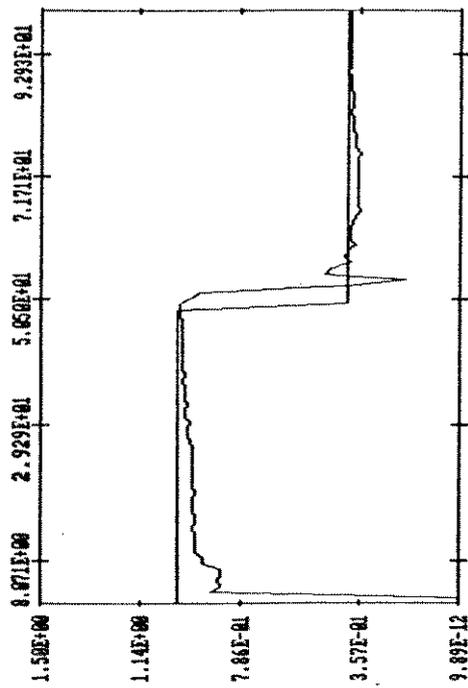


3.5.c - criterio do erro previsto

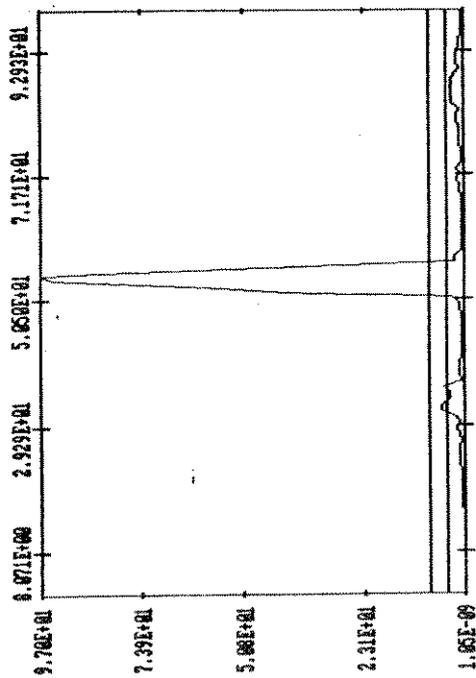
FIGURA 3.5  
 Algoritmo do traço Adaptativo - COMB2 -  $s/r = 0.2$  - lenta  
 mod 1  $\Rightarrow$  mod 2



3.6.a - Parametro a1

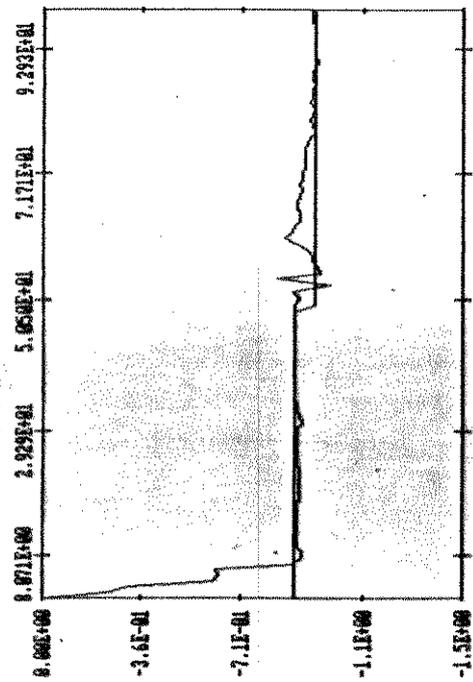


3.6.b - Parametro b0

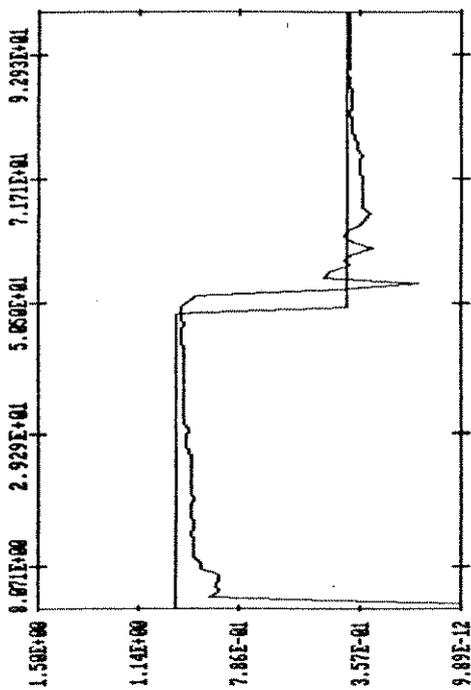


3.6.c - criterio do erro previsto

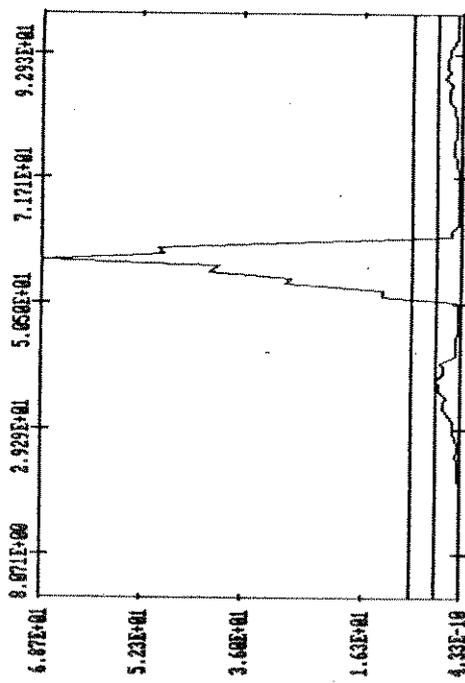
FIGURA 3.6  
 Algoritmo do traço adaptativo - COMB1 - s/r = 10 - rápida  
 mod 1 ←



3.7.a - Parametro a<sub>1</sub>

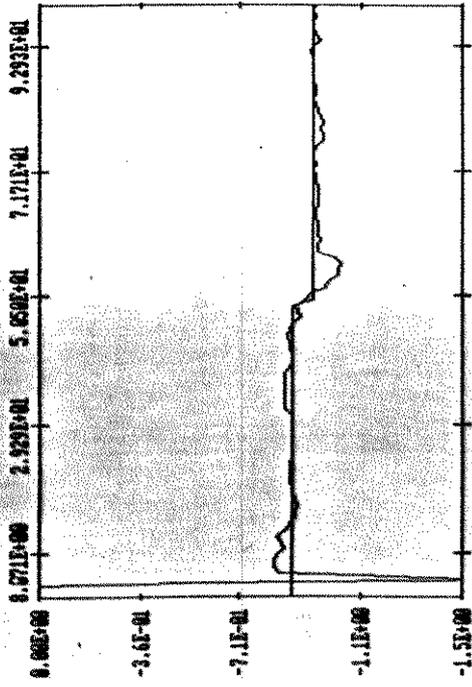


3.7.b - Parametro b<sub>0</sub>

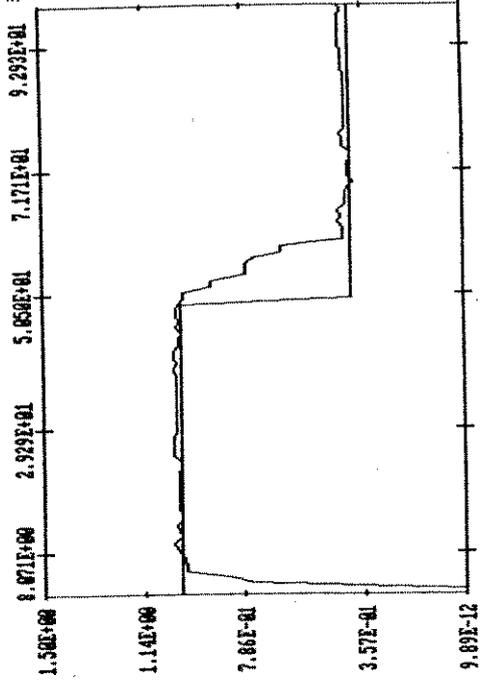


3.7.c - criterio do erro previsto

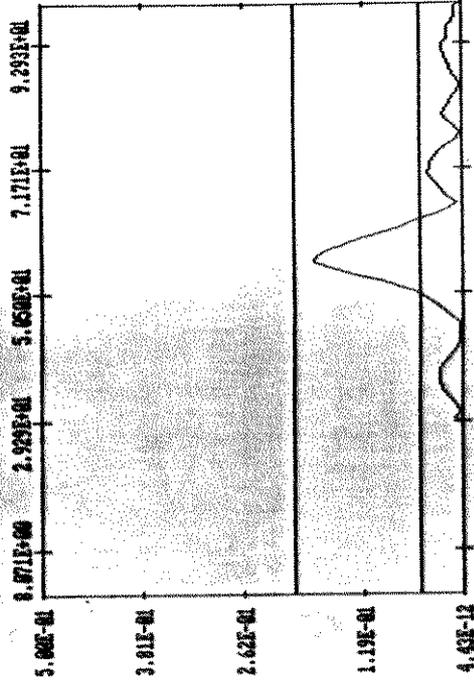
FIGURA 3.7  
 Algoritmo do traço adaptativo - COME1 -  $\epsilon/r = 10$  - rápida - JAN 1  
 mod 1 ~~pa~~ mod 2



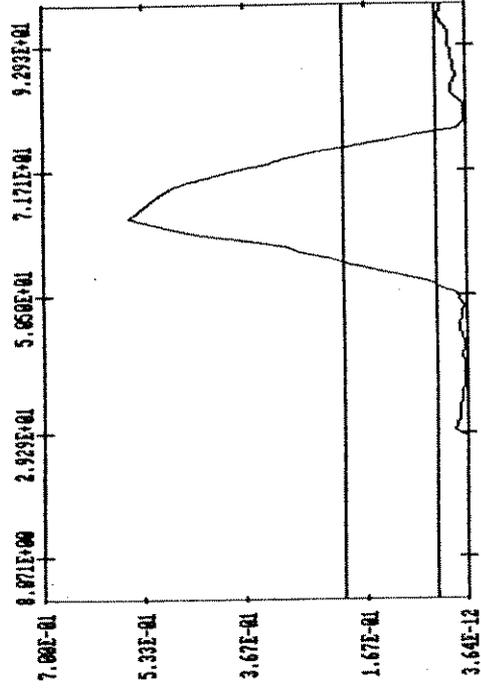
3.8.a - Parametro a1



3.8.b - Parametro b0

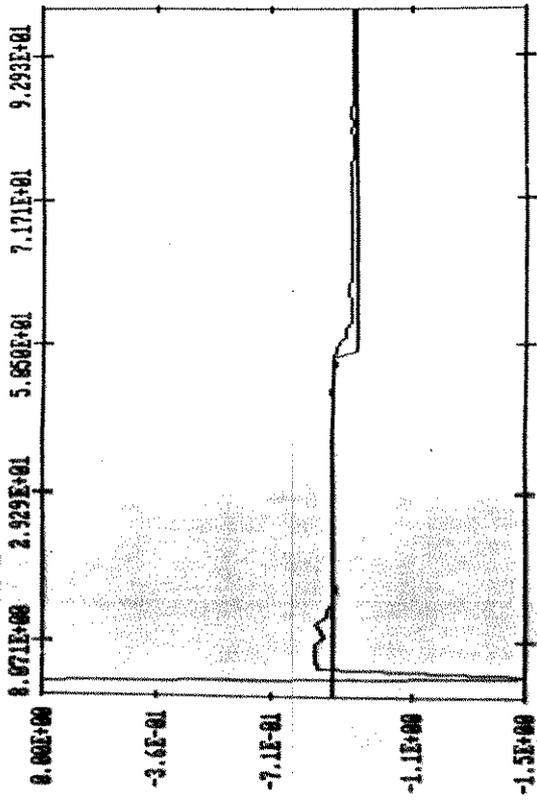


3.8.c - criterio do parametro a1

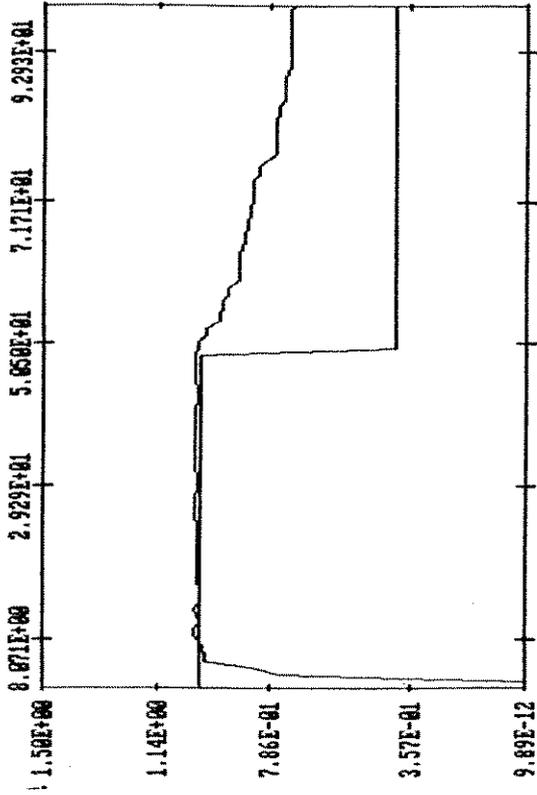


3.8.d - criterio do parametro b0

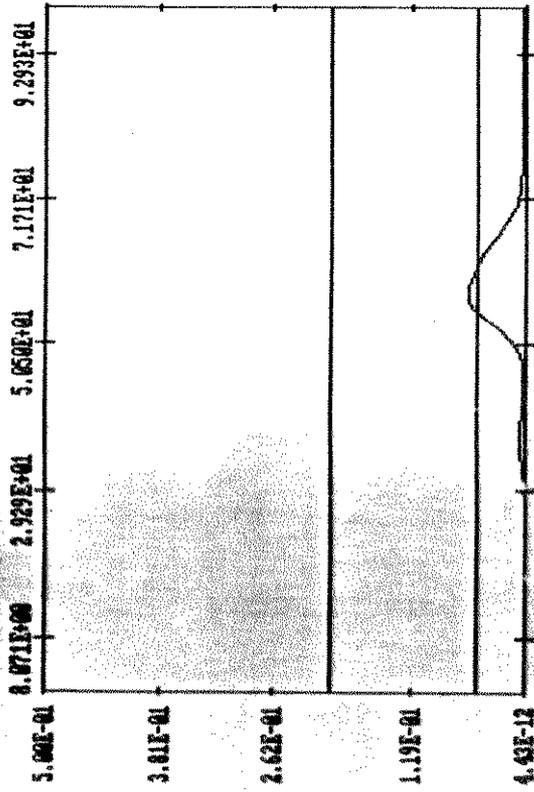
FIGURA 3-8  
Filtro de Kalman Adaptativo - COMB1 - s/r = 10 - rapida  
mod 1 pp mod 2



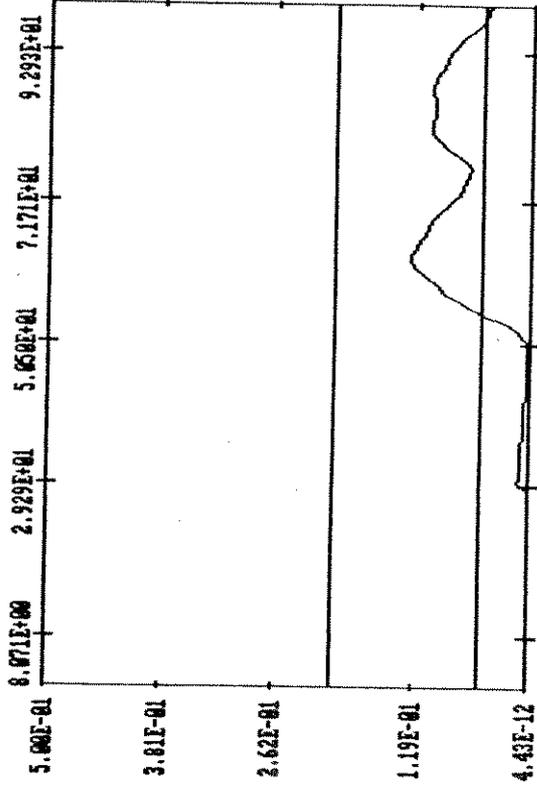
3.9.a - Parametro  $a_1$



3.9.b - Parametro  $b_0$

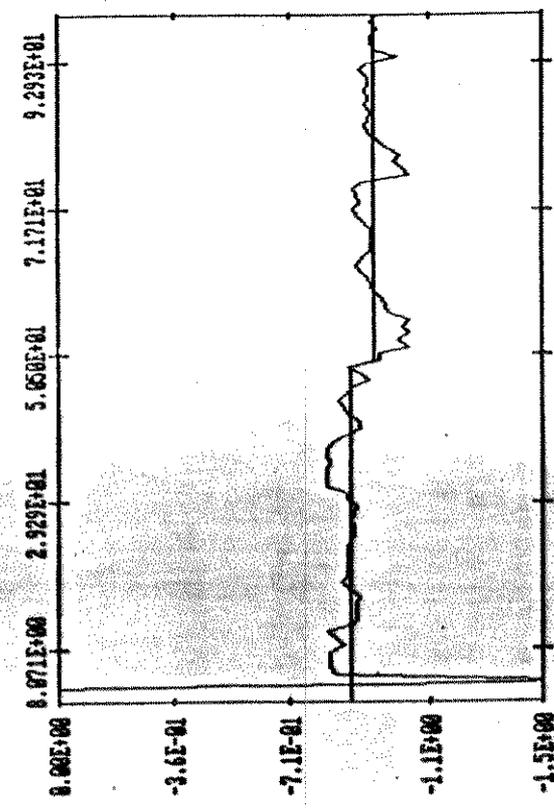


3.9.c - Critério do Parametro  $a_1$

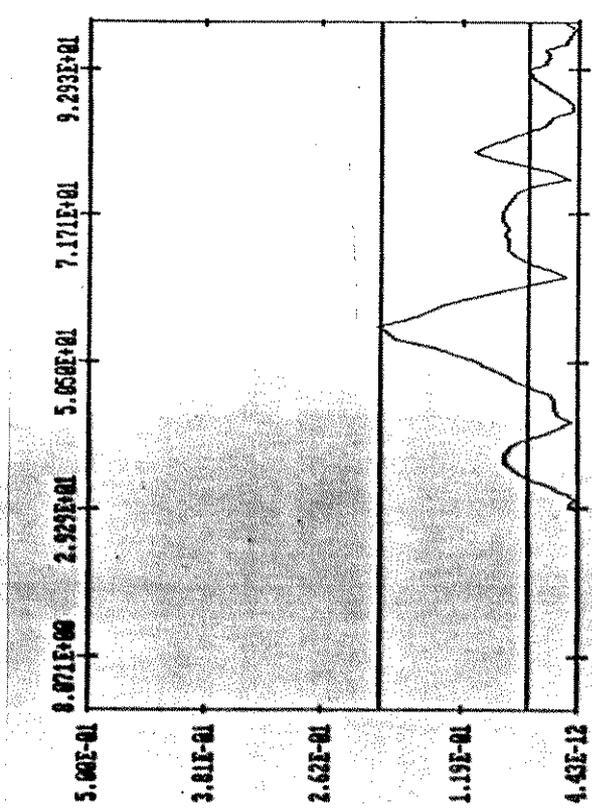


3.9.d - Critério do Parametro  $b_0$

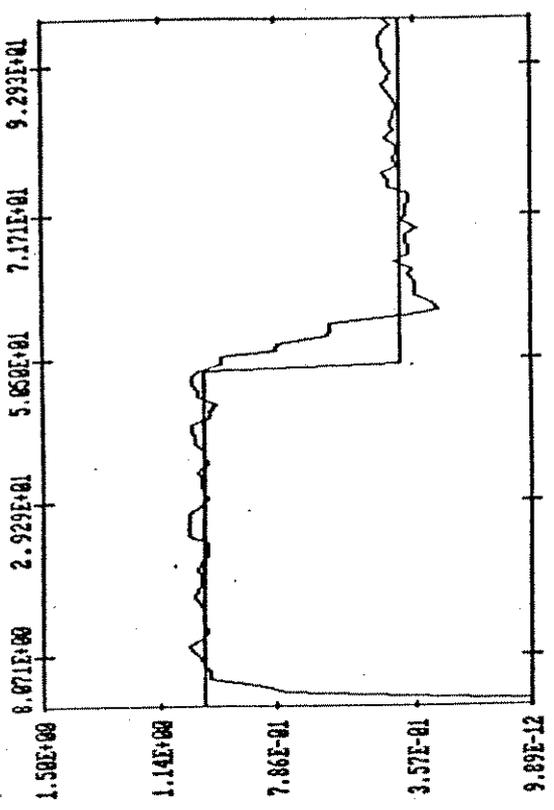
FIGURA 3.9  
Filtro de Kalman Adaptativo - COMB2 -  $s/r = 10$  - rápida  
MOD 1 2A mod 2



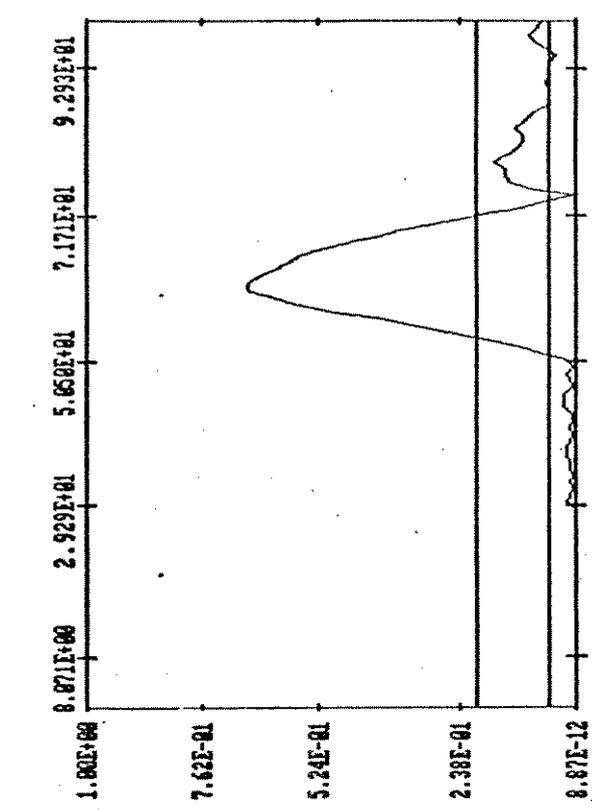
3.10.a - Parametro  $a_1$



3.10.c - Critério do parametro  $a_1$

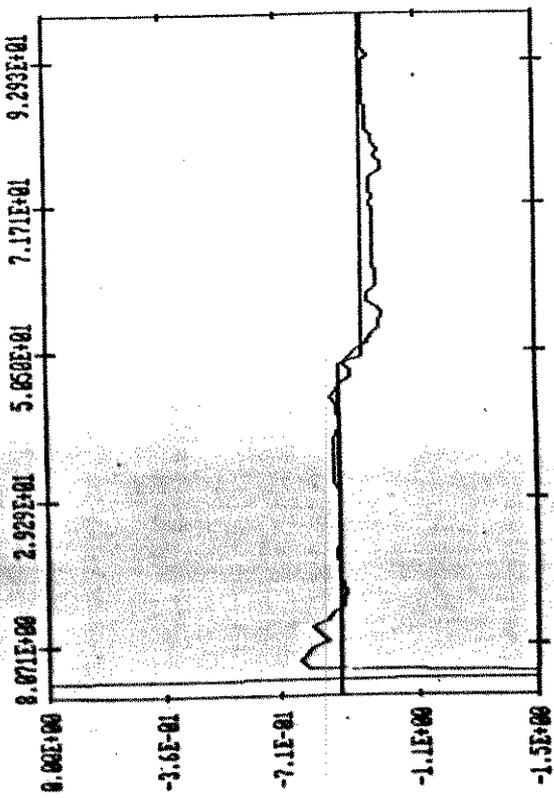


3.10.b - Parametro  $b_0$

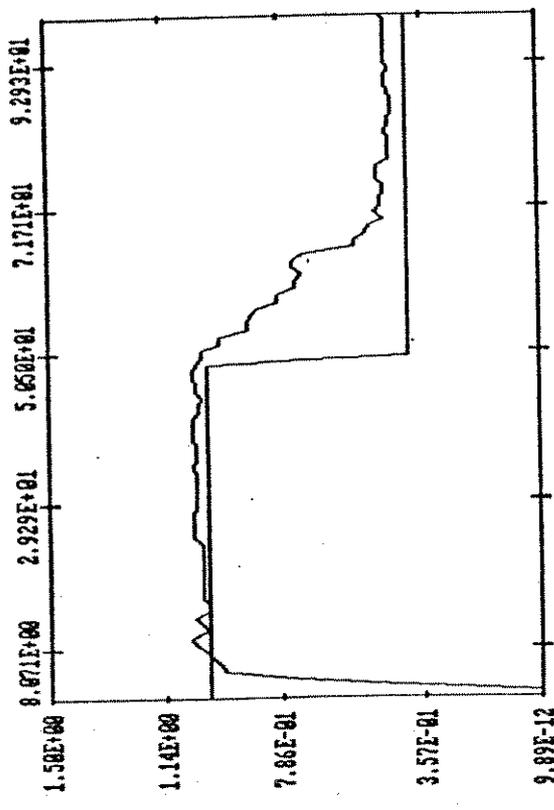


3.10.d - Critério do parametro  $b_0$

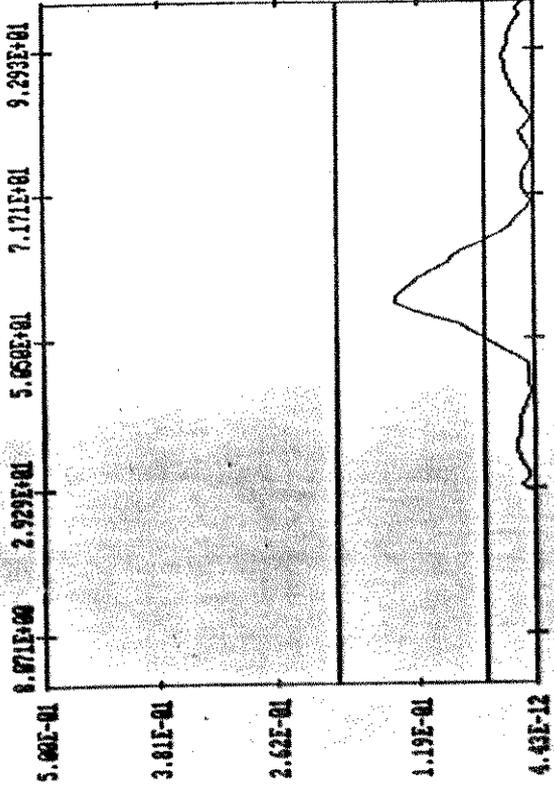
FIGURA 3-10  
 Filtro de Kalman Adaptativo - COMB3 - s/r = 10 - rapida  
 MOD 1 84 mod 2



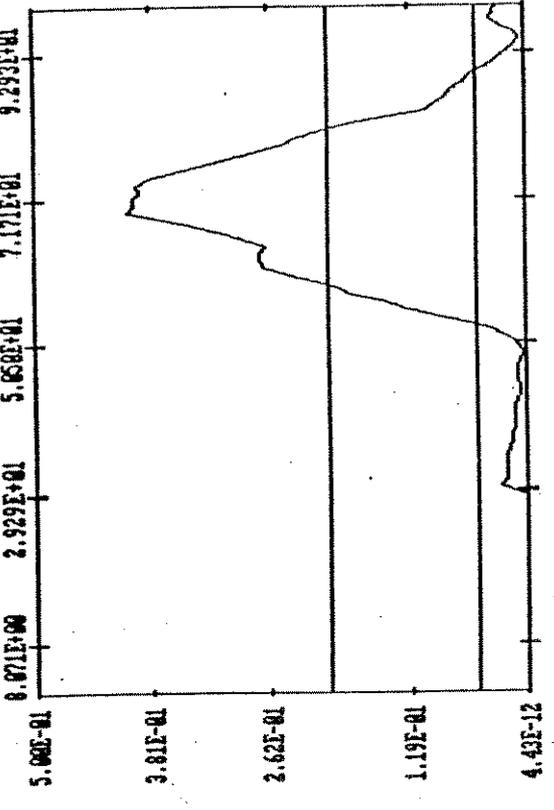
3.11.a - Parametro  $a_1$



3.11.b - Parametro  $b_0$



3.11.c - criterio do parametro  $a_1$



3.11.d - criterio do parametro  $b_0$

FIGURA 3.11  
Filtro de Kalman Adaptativo - COMB1 - s/r = 05 - rapida  
mod 1 de mod 2

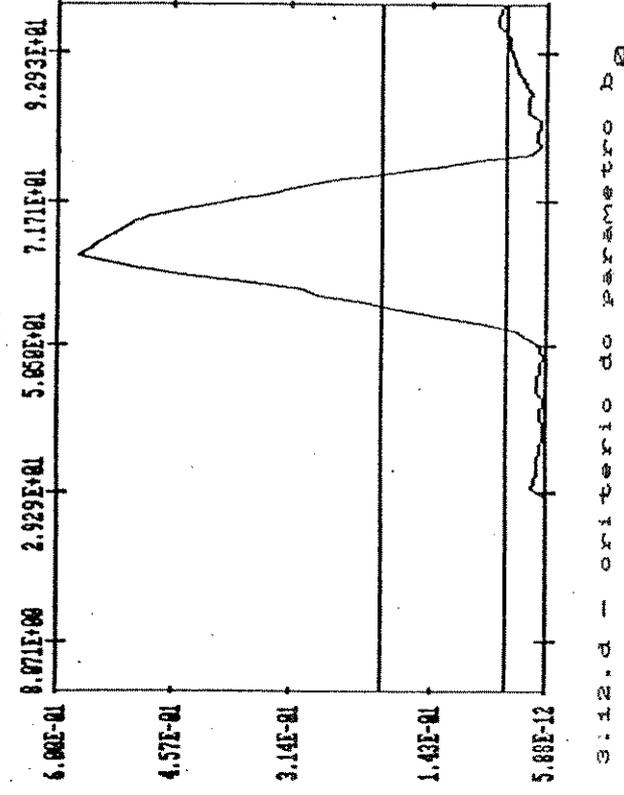
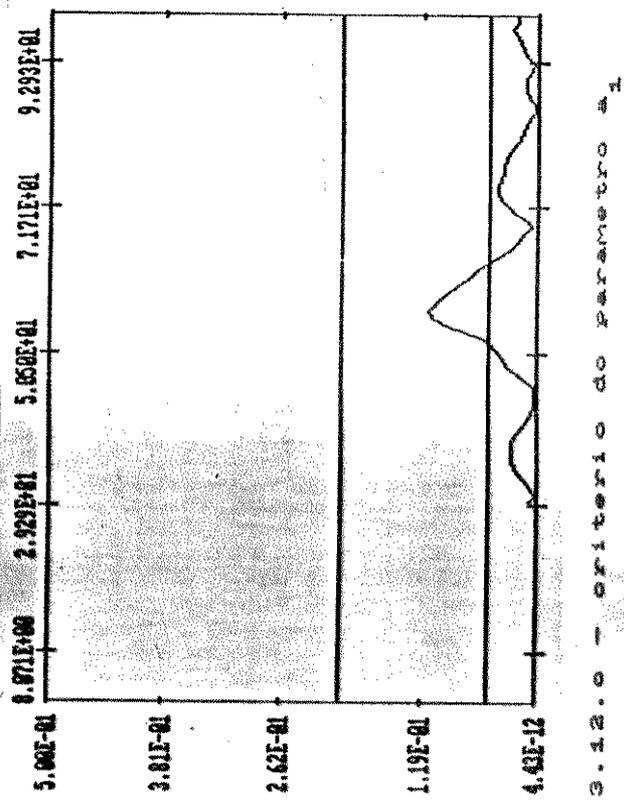
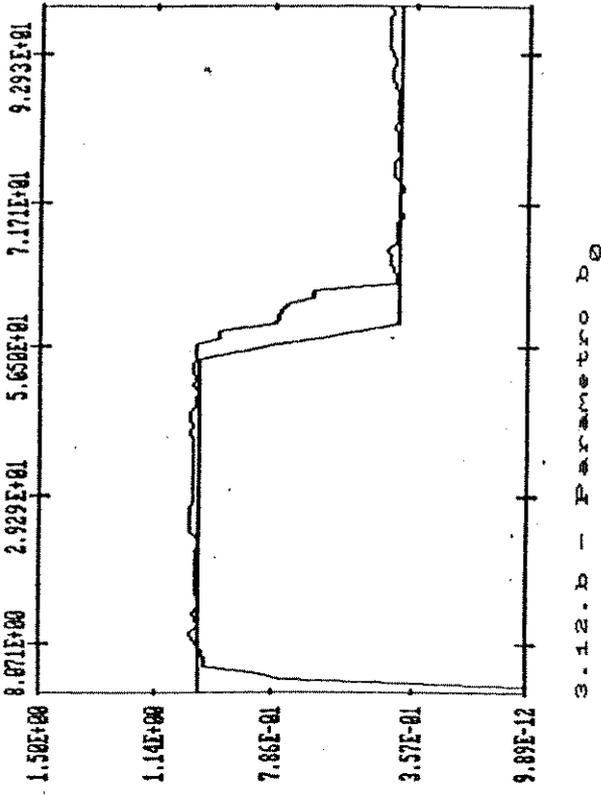
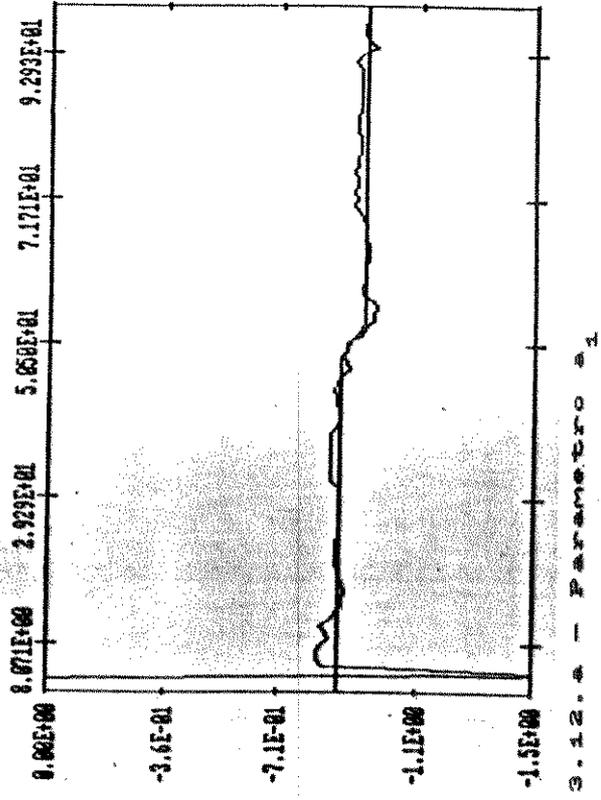
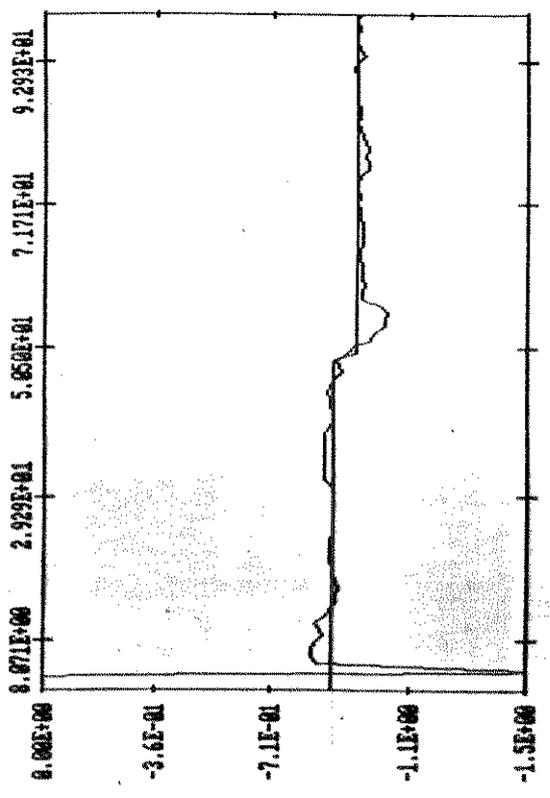
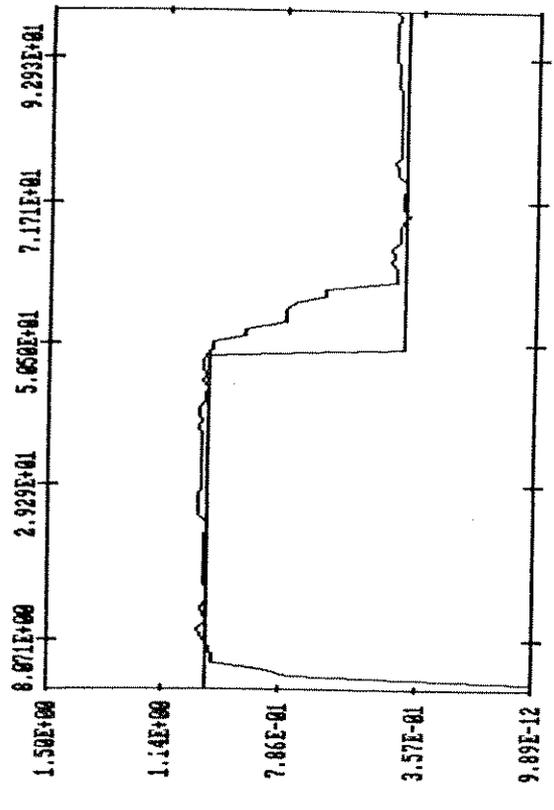


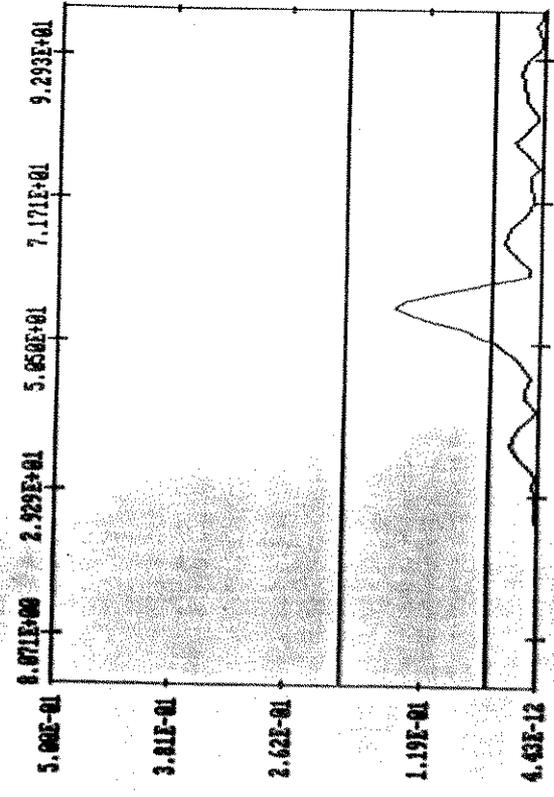
FIGURA 3.12  
 Filtro de Kalman Adaptativo - COMBI - s/r = 10 - lenta  
 mod 1 844 mod 2



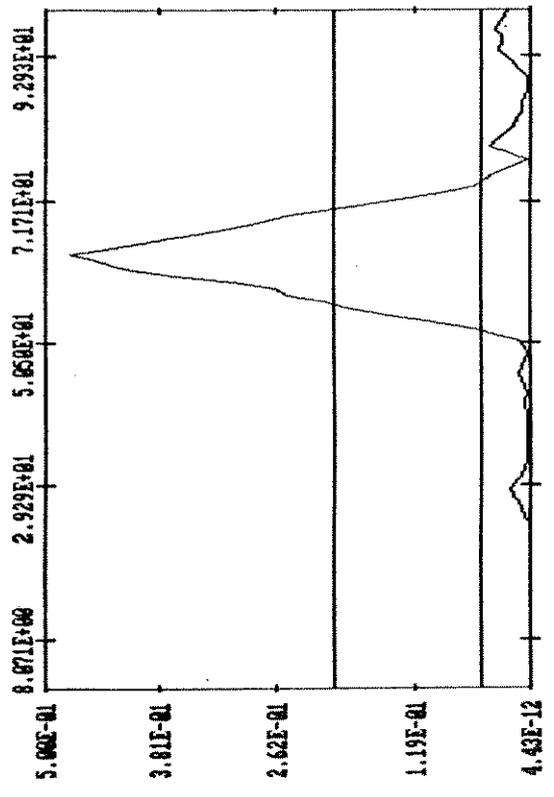
3.13.a - Parametro  $a_1$



3.13.b - Parametro  $b_0$



3.13.c - Critério do parametro  $a_1$



3.13.d - Critério do parametro  $b_0$

FIGURA 3-13  
 Filtro de Kalman Adaptativo - COMB1 -  $s/r = 10$  - rápida - JAN 3  
 mod 1  $\Rightarrow$  mod 2

Dos resultados obtidos para os dois algoritmos conclui-se que o critério dos parâmetros é mais sensível a uma má regulagem dos parâmetros característicos do detetor, principalmente dos níveis de adaptação, que o critério do erro previsto.

### 3.5 CONCLUSÕES SOBRE A REGULAGEM DOS PARÂMETROS DE DETECÇÃO DOS ALGORITMOS ADAPTATIVOS

Da análise dos resultados das várias simulações realizadas, conclui-se que os parâmetros dos testes de detecção ( i)limiares, ii) níveis de adaptação e iii) horizontes de cálculo) devem ser escolhidos em função dos seguintes dados: relação sinal/ruído, tipo de não-estacionaridade, ordem de grandeza dos coeficientes dos polinômios A e B e tipo de excitação.

O limiar  $J_{min}(t)$  para o critério dos parâmetros deve ser utilizado para melhorar a convergência em caso de variações incertas. Das simulações pode-se concluir que ele é constante e independente dos dados acima. Sua ordem de grandeza deve ser igual ao valor máximo do critério em regime estacionário.

Os horizontes ( $N_1$  e  $N_2$ ) devem ser suficientemente grandes para permitir uma estimação satisfatória de  $\sigma_l^2(t)$  ou  $\bar{\theta}_l(t)$ .

O horizonte curto  $N_1$  para o critério dos parâmetros deve ser pequeno a fim de não introduzir um atraso elevado na detecção, no entanto ele deve ser superior ao período de oscilação "curto" do estimador, para evitar que ocorram detecções erradas.

O horizonte curto  $N_0$  para o critério do erro previsto não deve ser escolhido grande pois isso resulta numa atenuação do pico de detecção. Também não deve ser pequeno pois, neste caso

tem-se um estimador muito sensível às perturbações existentes no processo.

A seguir apresentam-se as principais conclusões sobre a influência de cada um dos dados acima especificados na regulação dos parâmetros de detecção.

### 3.5.1 Relação sinal-ruído

O critério do erro previsto é dependente da variância do ruído  $\sigma_e^2(t)$ , pois  $J(t)$  é inversamente proporcional a essa variância. Portanto o limite de detecção  $J_{max}(t)$  deve ser reduzido quando  $\sigma_e^2(t)$  aumenta.

Para o critério dos parâmetros deve-se aumentar  $J_{max}(t)$  quando  $\sigma_e^2(t)$  aumenta para evitar que ocorram detecções falsas, já que os parâmetros estimados apresentam uma variância maior devido a presença de um ruído forte.

Os níveis de adaptação mínimos também devem ser reduzidos quando o ruído é elevado, para que não ocorram detecções falsas, já que os parâmetros estimados são fortemente influenciados pela perturbação existente no processo.

### 3.5.2 Tipo de variação ( amplitude, velocidade )

Para variações grandes em amplitude pode-se aumentar os limites de detecção pois os valores dos critérios de detecção são proporcionais à amplitude da variação. Portanto para variações lentas o nível de adaptação máximo deve ser diminuído.

### 3.5.3 Ordem de grandeza dos polinômios A e B

Os valores do traço da matriz de covariância são proporcionais às ordens de grandeza dos polinômios A e B e a ordem do sistema [23], portanto os níveis de adaptação também são proporcionais a estas características.

### 3.5.4 Tipo de entrada

Se a entrada for pouco excitante a convergência do estimador será lenta e os estimadores terão uma pequena variância. No caso do critério dos parâmetros consegue-se um melhor desempenho diminuindo o limiar de detecção  $J_{\max}(t)$ .

## 3.6 ANÁLISE COMPARATIVA ENTRE OS DOIS MÉTODOS DE DETECÇÃO DE NÃO-ESTACIONARIDADES

Nas simulações apresentadas nesta seção os parâmetros de regulação para o método do erro previsto são

$$\begin{aligned} N_l &= 10, \quad N_c = 5 \quad \text{e} \quad \tau = 0, \\ J_{\min} &= 4.0 \quad \text{e} \quad J_{\max} = 8.0, \\ tr_{\min} &= 0.01, \quad tr_{\text{int}} = 0.1 \quad \text{e} \quad tr_{\max} = 1.0, \end{aligned}$$

e para o critério dos parâmetros tem-se

$$\begin{aligned} N_z &= 20, \quad N_1 = 5 \quad \text{e} \quad \tau = 5, \\ J_{\min} &= 0.05 \quad \text{e} \quad J_{\max} = 0.2, \\ q_{\min} &= 0.01, \quad q_{\text{int}} = 0.01 \quad \text{e} \quad q_{\max} = 2.0. \end{aligned}$$

Utiliza-se o algoritmo do traço adaptativo e o algoritmo do filtro de Kalman adaptativo para a identificação de processos que sofrem variações de pequena amplitude (modelo 1 para modelo 3) e de grande amplitude (modelo 1 para modelo 4), essas variações podem ser lentas ou rápidas. As relações sinal-ruído são 10, 5 e 2.

Os resultados obtidos para o algoritmo do traço adaptativo com o método do erro previsto são mostrados na tabela 07 e os resultados do filtro de Kalman adaptativo com o critério

dos parâmetros estão na tabela 08.

Da comparação entre estas tabelas verifica-se que o critério do erro previsto possui um retardo menor na detecção que o critério dos parâmetros ( ver figuras 3.14 e 3.15 ). A razão deste retardo maior é que o critério dos parâmetros acumula dois atrasos na detecção: um atraso de reação do estimador à variação do parâmetro e um atraso, cerca de  $N_1$ , necessário para se ter como certa a evolução do parâmetro. Estes dois atrasos são mostrados na figura 3.16 a seguir. O primeiro atraso, atraso de reação à variação do parâmetro, pode ser diminuído se o nível de adaptação mínimo for aumentado, diminuindo assim o retardo total na detecção como foi feito na seção 3.4.3. Contudo isto compromete a convergência do estimador já que aumenta sua polarização e variância.

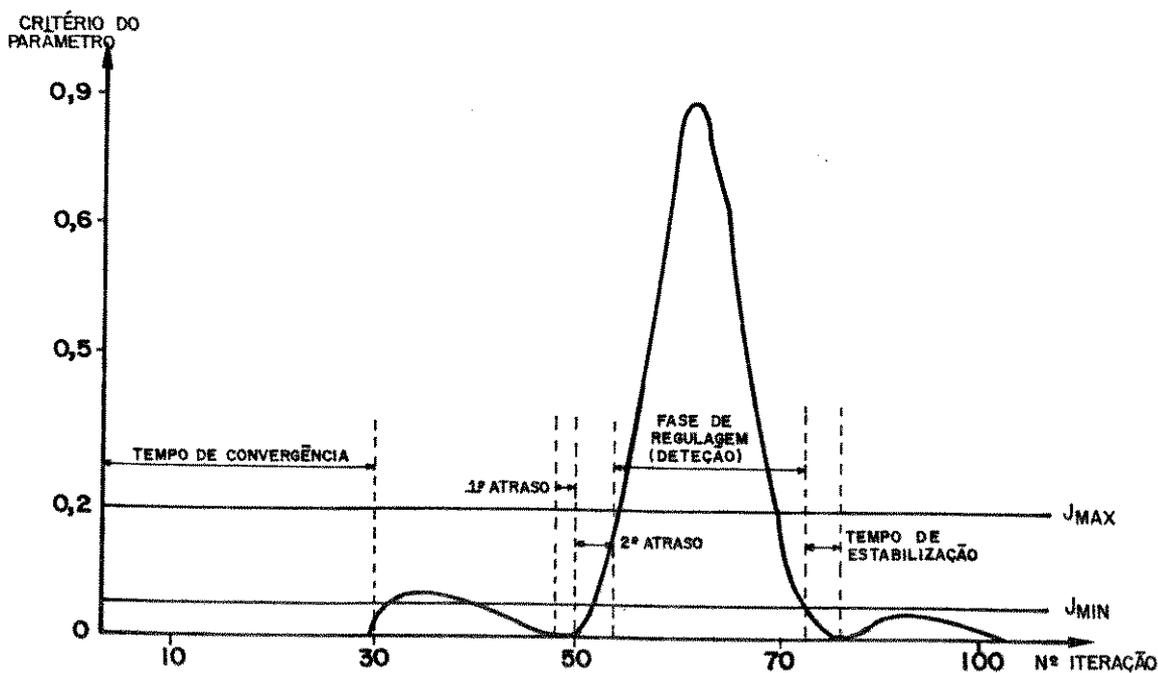


FIGURA 3.16

Comportamento do critério dos parâmetros quando acontece uma não - estacionaridade

Da análise das tabelas 7 e 8 também verifica-se que o critério do erro previsto é mais suscetível a alarmes falsos.

FIGURAS	características das simulações			POL.	POL.	VAR.	VAR.	atraso na detec.	TAF certo	TAF incert	TND certa	TND incert
	valor de traco	razao sinal ruído	tipo de var.	a <sub>1</sub>	b <sub>0</sub>	a <sub>1</sub>	b <sub>0</sub>					
	mod 1	10	RAP.	-0.76E-03	0.84E-03	0.14E-05	0.14E-05	2.2	0.0	0.0131	0.0	0.0
	mod 3		LEN.	-0.88E-03	0.96E-03	0.15E-05	0.13E-05	3.6	0.030	0.0250	0.8	0.0
	MOD 1	5	RAP.	-0.32E-02	-0.14E-01	0.17E-04	0.26E-05	2.0	0.0	0.0166	0.8	0.0
	MOD 3		LEN.	-0.22E-02	-0.13E-01	0.24E-05	0.55E-05	4.1	0.013	0.0155	0.6	0.2
	MOD 1	2	RAP.	-0.29E-02	0.28E-01	0.95E-06	0.20E-05	3.3	0.0	0.0105	0.8	0.5
	MOD 3		LEN.	-0.27E-02	-0.26E-01	0.25E-07	0.85E-05	5.5	0.0	0.0060	0.8	0.6
3 .14	MOD 1	10	RAP.	-0.79E-03	0.12E-02	0.62E-04	0.61E-05	1.6	.0489	0.0582	0.0	0.0
	MOD 4		LEN.	-0.76E-32	0.11E-23	0.48E-06	0.68E-05	1.5	.0436	0.0568	0.0	0.0
	MOD 1	5	RAP.	-0.29E-03	0.45E-02	0.64E-06	0.57E-05	3.3	.0304	0.0532	0.6	0.0
	MOD 4		LEN.	-0.76E-04	0.10E-02	0.42E-06	0.51E-05	4.8	.0066	0.0540	0.6	0.0
	MOD 1	2	RAP.	-0.96E-03	0.66E-01	0.61E-06	0.33E-05	11.0	0.0	0.0111	0.8	0.6
	MOD 4		LEN.	-0.90E-03	0.54E-01	0.65E-06	0.33E-05	10.0	0.0	0.0026	0.8	0.6

TABELA 07

ALGORITMO DO TRAÇO ADAPTATIVO -- CRITERIO DO ERRO PREVISTO

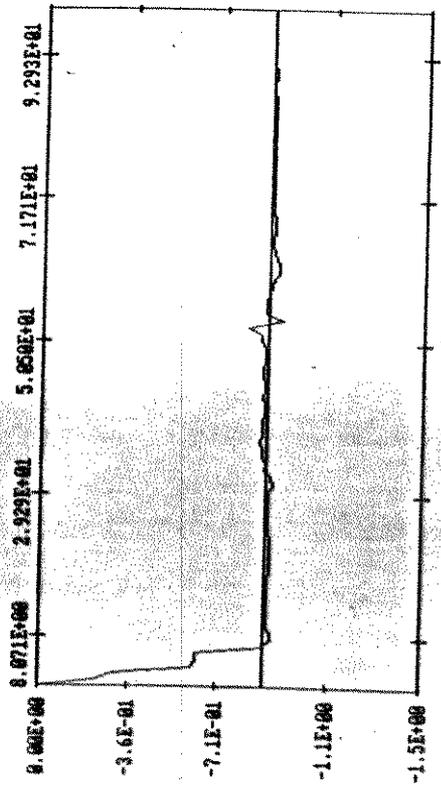
Parametros : N1 = 10      Jmin = 4.0      trmin = 0.01  
                   Nc = 5        Jmax = 8.0      trint = 0.10  
                   Tau = 0         trmax = 1.00

FIGURAS	características das simulações			POL.	POL.	UAR.	UAR.	atraso na detec.	TAF	TAF	TND	TND
	valor de traco	razao sinal ruído	tipo de var.	a <sub>1</sub>	b <sub>0</sub>	a <sub>1</sub>	b <sub>0</sub>		certo	incert	certa	incert
	mod 1	10	RAP.	0.78e-02	-0.10E-01	0.63E-04	0.12E-03	8.4	0.0	0.0	0.0	0.0
	mod 3		LEN.	0.76E-02	-0.10E-01	0.64E-04	0.11E-03	8.5	0.0	0.0	0.0	0.0
	MOD 1	5	RAP.	0.54E-02	-0.20E-01	0.73E-04	0.13E-03	12.2	0.0	0.0	0.0	0.0
	MOD 3		LEN.	0.52E-02	-0.20E-01	0.72E-04	0.14E-03	12.2	0.0	0.0	0.0	0.0
	MOD 1	2	RAP.	-0.12E-01	-0.84E-01	0.81E-04	0.17E-03	16.25	0.0	0.0	0.2	0.2
	MOD 3		LEN.	-0.12E-01	-0.83E-01	0.80E-04	0.18E-03	16.375	0.0	0.0	0.2	0.2
3 .15	MOD 1	10	RAP.	0.38E-02	-0.66E-02	0.43E-04	0.26E-03	6.0	0.0	0.0	0.0	0.0
	MOD 4		LEN.	0.37E-02	-0.64E-02	0.43E-04	0.29E-03	6.2	0.0	0.0	0.0	0.0
	MOD 1	5	RAP.	0.68E-02	-0.20E-03	0.56E-04	0.35E-03	8.2	0.0	0.0	0.0	0.0
	MOD 4		LEN.	0.68E-02	-0.20E-03	0.56E-04	0.37E-03	8.7	0.0	0.0	0.0	0.0
	MOD 1	2	RAP.	0.11E-01	0.74E-01	0.72E-04	0.43E-03	12.4	0.0	0.0	0.0	0.0
	MOD 4		LEN.	0.11E-01	0.75E-01	0.72E-04	0.53E-03	12.5	0.0	0.0	0.0	0.0

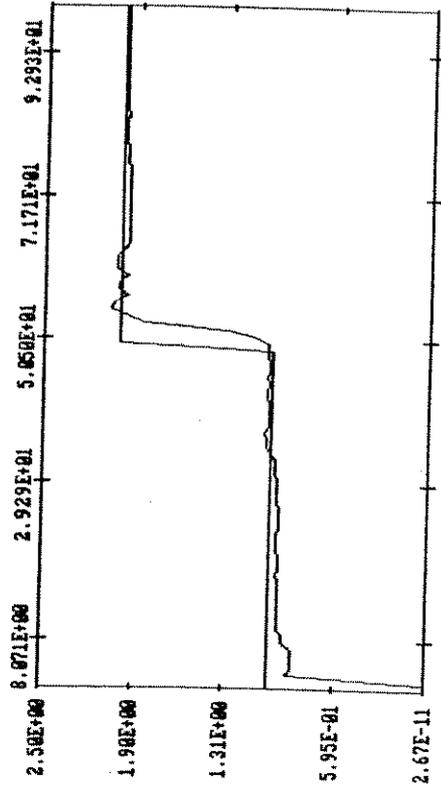
TABELA 08

FILTRO DE KALMAN ADAPTATIVO - CRITERIO DOS PARAMETROS

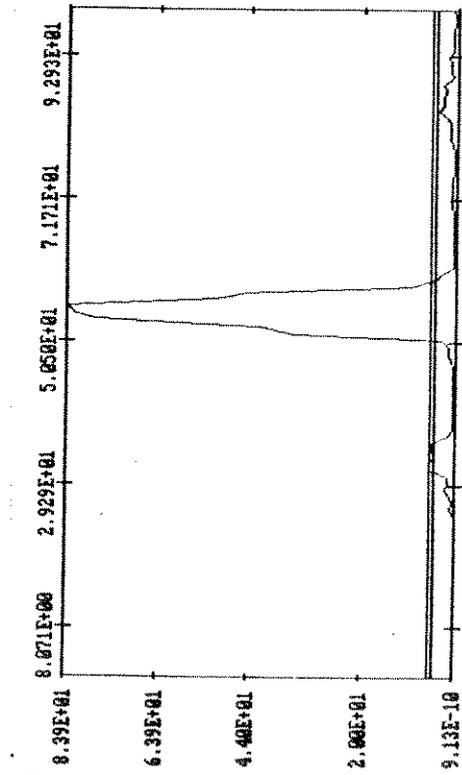
Parametros : N1 = 10      Jmin = 0.05      qmin = 0.01  
                  Nc = 5        Jmax = 0.2        qint = 0.01  
                  Tau = 5                                    qmax = 2.00



3.14.a - Parametro  $p_1$



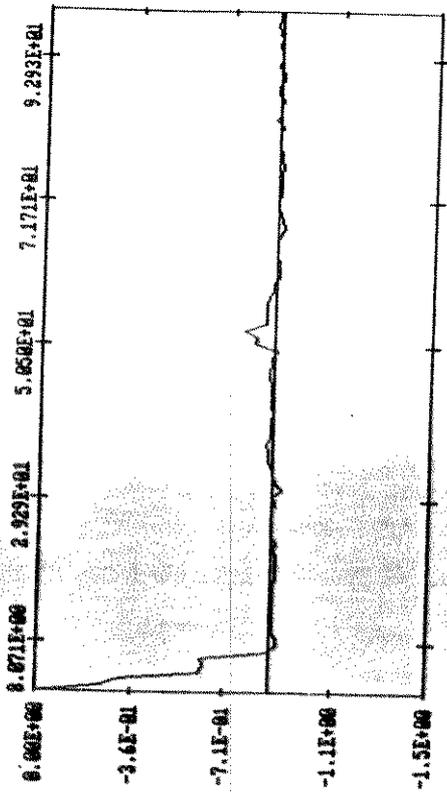
3.14.b - Parametro  $p_0$



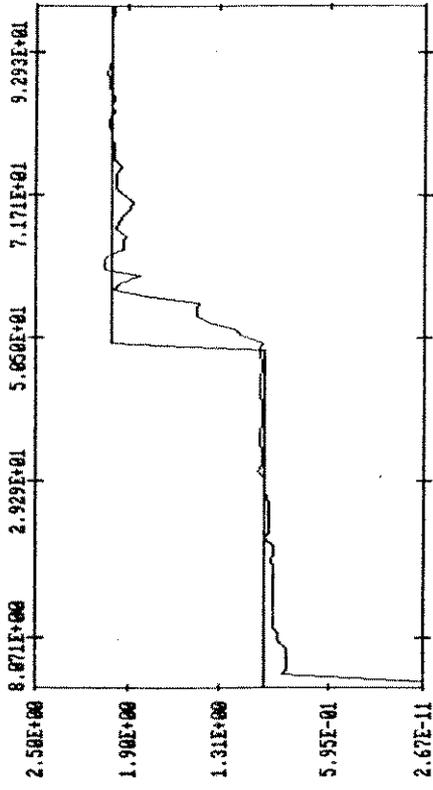
3.14.c - criterio do erro previsto

Algoritmo do traço Adaptativo - COMB1 -  $s/r = 10$  - rápida  
 mod 1 2a mod 4

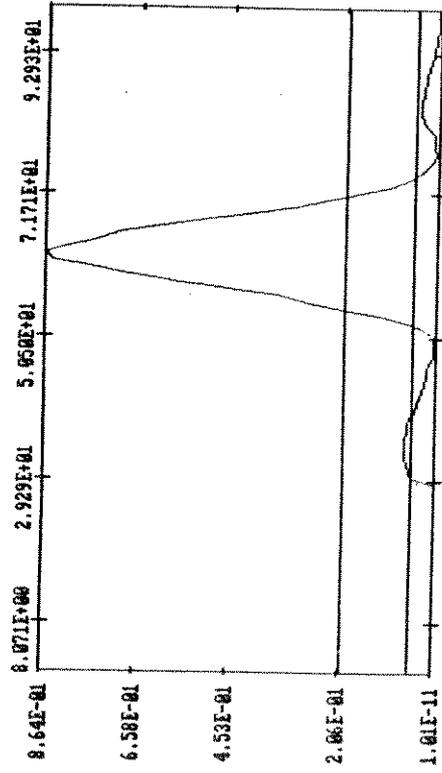
FIGURA 3.14



3.15.a - Parametro a1



3.15.b - Parametro b0



3.15.c - criterio do parametro b0

FIGURA 3.15  
 Filtro de Kalman Adaptativo - COMB1 - s/r = 10 - rapida  
 mod 1 - mod 4

Isto ocorre porque este critério é baseado em estimativas da variância do erro previsto, e se em um determinado instante de amostragem a perturbação no processo é grande, o critério considera que ocorreu uma não-estacionaridade, resultando numa detecção errada e em oscilações no estimador. Esta condição não se verifica com o critério dos parâmetros, que é baseado em valores médios dos parâmetros estimados, pois quando ocorrem valores elevados de ruído, eles têm pouca influência nestas médias.

As taxas de alarme falso podem ser diminuídas para os dois critérios, se o número de iterações de validação é aumentado, no entanto este aumento leva a um maior atraso na detecção.

Quando ocorrem variações lentas ou de pequena amplitude, o critério do erro previsto apresenta taxas de não-deteção elevadas. Isto ocorre porque pequenas variações resultam em pequenos erros previstos e, em consequência a variância  $\sigma_c^2(t)$  não sofre modificações significativas, não favorecendo a detecção.

No critério dos parâmetros ocorre detecção mesmo para variações lentas, porque à medida que o horizonte de cálculo do critério se desloca, as variações, mesmo pequenas, são incorporadas nos valores médios utilizados para cálculo de  $\bar{\theta}(t)$ . Nos casos em que as variações nos parâmetros são elevadas o atraso na detecção é praticamente o mesmo independente da variação ser rápida ou lenta.

Para variações de pequena amplitude, a detecção no método do critério dos parâmetros pode não ocorrer. Se no entanto,  $q_{\min}$  for suficiente para permitir uma adaptação mínima, o estimador consegue atingir os novos valores após a variação.

Um melhor desempenho de um dos algoritmos de identificação quanto a polarização e a variância dos estimadores dependerá das condições em que é aplicado o algoritmo de identificação. Para variações lentas no processo, o critério dos

parâmetros apresenta melhor desempenho ( ver figuras 3.2 e 3.12 ); por outro lado, se o processo sofre várias não-estacionaridades sucessivas, o critério do erro previsto tem um melhor desempenho.

Em termos de tempo de convergência para os novos valores após uma variação, o critério dos parâmetros é mais lento devido aos dois atrasos inerentes ao método que foram citados anteriormente ( ver figuras 3.14 e 3.15 ).

Para relações sinal-ruído baixas ( alto nível de ruído ) ambos os métodos apresentam desempenho inferiores àqueles apresentados com relação sinal-ruído alta ( ver figuras 3.3 e 3.4, 3.8 e 3.11 ).

Com o algoritmo do filtro de Kalman aliado ao critério dos parâmetros é possível ajustar apenas o nível de adaptação relativo ao parâmetro que está variando. Neste caso os outros parâmetros estimados são fracamente influenciados pelo aumento deste nível de adaptação. Para o algoritmo do traço adaptativo e critério do erro previsto, o aumento de "tro" influencia fortemente a estimação de todos os parâmetros do vetor  $\hat{\theta}(t)$ , mesmo os que não sofrem variações.

### 3.7 SELEÇÃO DOS MÉTODOS DE DETEÇÃO

Das conclusões apresentadas na seção 3.6, a escolha entre a utilização de um dos dois métodos de detecção deve ser baseada no comportamento do algoritmo segundo as situações apresentadas.

Para relações sinal-ruído pequenas, variações lentas ou de pequena amplitude deve-se utilizar o critério dos parâmetros que neste caso, apresenta uma probabilidade de detecção maior.

Para relações sinal-ruído elevadas e variações de grande amplitude e rápidas o critério do erro previsto apresenta um melhor desempenho. Pode-se ter um desempenho aceitável com este método se as variações são lentas e a relação sinal-ruído elevada.

Se o processo a ser estimado sofre várias não-estacionaridades sucessivas, o critério do erro previsto deve ser aplicado pois apresenta um menor atraso na detecção e uma convergência mais rápida para os novos valores. Esta escolha é ilustrada na figura 3.17 onde apresenta-se o desempenho dos dois algoritmos na estimação de um parâmetro que sofre variações de 50% em  $t = 35$  e  $t = 70$ .

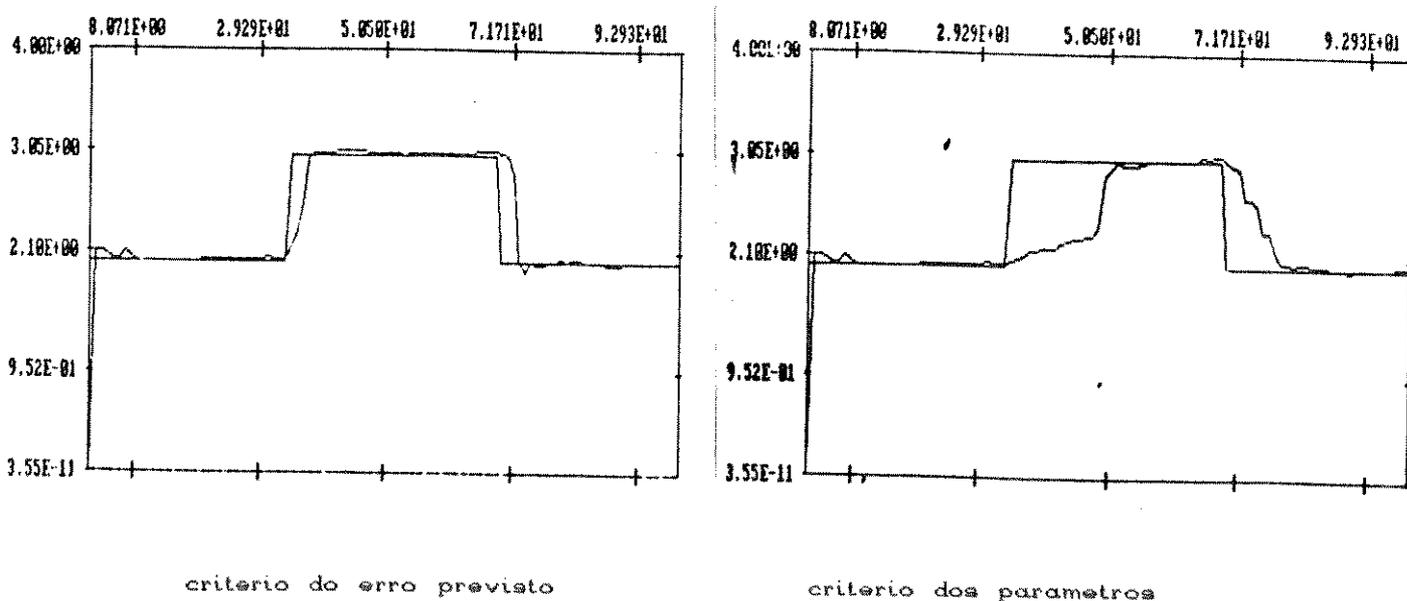


FIGURA 3.17

Desempenho dos algoritmos adaptativos na estimação de um parâmetro que sofre duas variações sucessivas de 50%.

( $s/r = 10$ )

## CAPÍTULO 4

### SISTEMA ESPECIALISTA EM MODELAGEM

#### 4.1 INTRODUÇÃO

O objetivo do sistema especialista em modelagem é realizar um diagnóstico sobre um algoritmo de identificação que está sendo executado. Sua principal função é supervisionar os procedimentos de estimação de parâmetros e verificar se o modelo identificado representa adequadamente o comportamento do processo em estudo.

O desempenho dos algoritmos de identificação é analisado pelo sistema especialista em modelagem tendo como referência critérios relativos à polarização do estimador, à convergência do estimador, à eficiência na detecção de não-estacionaridades e à existência ou não de excitação no processo. Estas informações estão contidas na Base de Conhecimento do sistema especialista e representam um conhecimento factual, heurístico e não-algorítmico obtido diretamente do conhecimento acumulado sobre o desempenho do identificador.

A estrutura lógica do sistema especialista em modelagem é mostrada na figura 4.1. O software do sistema é formado pelos seguintes módulos : base de conhecimento, base de dados, módulo de inferência e explicação, módulo supervisor, módulo de identificação, e interface com o usuário.

O sistema especialista é implementado modularmente. A primeira função executada é a estimação paramétrica realizada por um dos algoritmos de identificação contidos no módulo de

identificação, que gera o modelo do processo. Nesta etapa também são obtidas medidas que permitem determinar a condição do estimador, isto é o seu estado, conforme será descrito na seção 4.3. O modelo e as medidas obtidas são armazenadas na base de dados. A seguir, o módulo de inferência, a partir da base de conhecimento e da base de dados, realiza inferências, gerando um diagnóstico sobre o desempenho do algoritmo de identificação. Os diagnósticos obtidos e uma avaliação do desempenho do algoritmo de identificação são apresentados ao usuário que pode solicitar uma explicação da linha de raciocínio utilizada. Ocorrendo esta solicitação é ativado o módulo de explicação. Por último o módulo supervisor pode realizar as modificações sugeridas, no algoritmo de identificação.

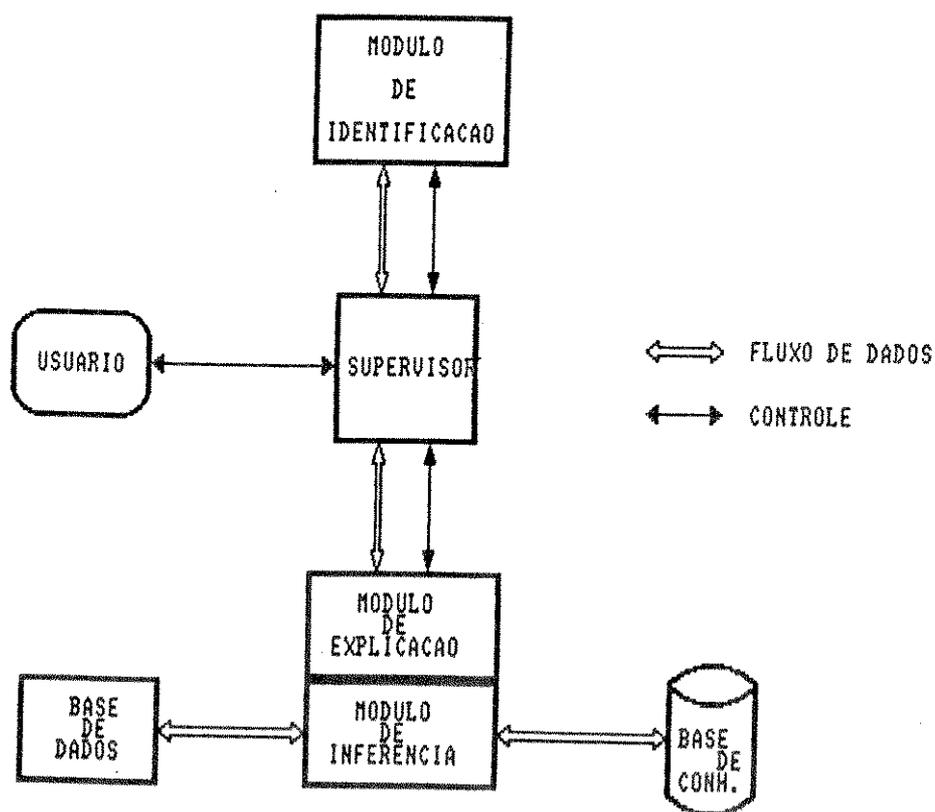


FIGURA 4.1  
Estrutura lógica do Sistema Especialista em Modelagem

Cada um dos módulos que compõem o sistema especialista em modelagem estão descritos nas próximas seções deste capítulo: na seção 4.2 descreve-se a base de dados, na seção 4.3 descreve-se a base de conhecimento e discute-se o conhecimento nela armazenado, o módulo de inferência e explicação é apresentado na seção 4.4, a interface com o usuário é discutida na seção 4.5 e, na seção 4.6, apresenta-se o módulo supervisor e o módulo de identificação.

Na seção 4.7 especificam-se detalhes da implementação computacional do sistema especialista em modelagem e na seção 4.8 apresentam-se os diagnósticos obtidos na identificação de processos simulados.

## 4.2 BASE DE DADOS

Em sistemas de produção, conforme discutido na seção 1.4.1, a representação de fatos e objetos geralmente é feita através de triplas do tipo:

<ATRIBUTO> , <OBJETO> , <VALOR>

Entretanto, esta representação não é muito adequada para representar fatos e objetos em controle de processos, porque neste caso, torna-se difícil identificar objetos específicos que necessariamente compartilhem um atributo comum [16]. Este fato levou à escolha de uma representação por duplas do tipo

<ATRIBUTO> , <VALOR>

Na verdade esta descrição por duplas é equivalente a uma por triplas onde a descrição do <OBJETO> é incluída no nome <ATRIBUTO>. Por exemplo, o fato "a média do erro previsto é zero" pode ser igualmente representada pela tripla

<MÉDIA> , <ERRO> , <ZERO>

ou pela dupla

<MÉDIA DO ERRO> , <ZERO>

Usando a representação por duplas, a base de dados contém mais atributos que em uma representação por triplas associativas [16]. Entretanto sua manipulação pelo módulo de inferência torna-se mais fácil, pois a base de dados passa a ser um vetor, onde cada combinação objeto-atributo corresponde a uma posição no vetor e, o valor desta posição corresponde ao valor associado à combinação objeto-atributo. Também a representação por duplas requer um menor espaço de memória, pois a base de dados é um vetor, e na representação por triplas a base de dados é constituída de diversas matrizes, pois a cada objeto é associado uma matriz cujas colunas representam os diversos atributos que um objeto pode ter [16, 17].

Na representação por duplas proposta, o valor associado a cada atributo pode ser de dois tipos:

i) Um valor qualitativo

<SISTEMA> , <ESTACIONÁRIO>

ii) Ou um valor quantitativo

<TRAÇO DA MATRIZ> , <ZERO>

Estes valores são armazenados na base de dados pelo módulo supervisor, a partir de i) informações recebidas do usuário (por exemplo: valores limites para o teste de detecção), ii) do algoritmo de identificação (por exemplo: valores dos parâmetros estimados), iii) ou de informações obtidas de inferências anteriores (por exemplo: estado do estimador).

Para cada dupla <ATRIBUTO> , <VALOR> existe um sinalizador usado pelo módulo supervisor (programa de controle) para validar o valor associado àquele atributo: se o sinalizador de uma determinada dupla for falso, o módulo supervisor associa a este atributo o valor "nulo" na base de dados; se o sinalizador for verdadeiro então a dupla tem o <VALOR> a ela associado armazenado na base de dados, na posição correspondente ao <ATRIBUTO>.

A base de dados do sistema especialista em modelagem é então formada por um vetor onde cada posição corresponde a uma das variáveis do identificador a ser analisado. Por exemplo, a posição 4 corresponde ao tempo de estabilização para o teste de detecção, a posição 15 corresponde ao traço da matriz de covariância, a posição 19 corresponde ao estado do estimador, etc.

#### 4.3 BASE DE CONHECIMENTO

O conhecimento adquirido sobre o desempenho dos algoritmos de identificação foi organizado na forma de regras de produção simples (conhecimento de primeira ordem). Estas regras possuem apenas operadores conjuntivos do tipo E, pois os operadores disjuntivos do tipo OU podem ser obtidos a partir de operadores E.

A sintaxe das regras é definida por uma gramática na forma normal de Backus estendida [08]. Cada regra possui o seguinte formato:

<regra> :: = SE <CONDIÇÃO> ENTÃO <CONCLUSÃO>  
<CONDIÇÃO> :: = <PRÉ-CONDIÇÃO> | <PRÉ-CONDIÇÃO> E <CONDIÇÃO>  
<PRÉ-CONDIÇÃO> :: = <ATRIBUTO> <OPERADOR> <VALOR>

onde <ATRIBUTO> é o nome da variável analisada, <VALOR> é o valor corrente desta variável e <OPERADOR> é um termo obtido a partir da classificação do valor corrente da variável, em relação aos limites estabelecidos na base de conhecimento. Operadores implementados no sistema especialista em modelagem são maior que, menor que, igual, diferente.

As regras estão classificadas em três níveis hierárquicos, de acordo com o tipo de conclusão obtida com a execução das regras. No nível 1, a partir de dados fornecidos pelo módulo de identificação, classifica-se o comportamento do sistema que está sendo modelado; no nível 2, a partir de dados fornecidos pelo módulo de identificação e de dados inferidos no nível 1, determina-se o comportamento do estimador; no nível 3, a partir de dados dos níveis inferiores e do módulo de identificação realiza-se um diagnóstico sobre o estado global do algoritmo de estimação, e faz-se sugestões para melhorar seu desempenho.

Esta hierarquização da base de conhecimento direciona a máquina de inferência a executar primeiro as regras dos níveis inferiores, facilitando assim a busca e a seleção das regras a executar.

As regras na base de conhecimentos do sistema especialista em modelagem são de dois tipos: regras de definição ou regras de conclusão.

Regras de definição são as utilizadas para estabelecer todas as condições necessárias para determinar o estado do sistema. Como estado do sistema, entende-se a descrição única dos valores das variáveis armazenadas na base de dados, que determinam o comportamento do estimador e do sistema que está sendo identificado. A regra mostrada a seguir é um exemplo de regra de definição:

**REGRA 5:**

SE o sistema é estacionário

E a norma da variação dos parâmetros estimados  $\geq$  limite converg.

ENTÃO a identificação não está convergindo

Regras de conclusão são aquelas cujas conclusões são utilizadas para sugerir modificações de variáveis e parâmetros no algoritmo de identificação. Um exemplo de regra de conclusão é a regra mostrada a seguir:

**REGRA 17:**

SE a identificação não está convergindo

E o sistema está bem excitado

E o traço da matriz  $P(t) \geq$  limite

ENTÃO o estimador não está convergindo porque há problemas na ordem do modelo, sendo necessário alterá-la.

As regras de definição por sua natureza, aparecem em todos os níveis da base de conhecimento do sistema especialista em modelagem, e as regras de conclusão são próprias do nível 3, pois é neste nível que se realizam os diagnósticos sobre o estado global da estimação.

A cada regra foi fornecido um número de prioridade que determina sua classificação no conjunto de regras e, que é utilizado pelo módulo de inferência para selecionar a regra a ser executada durante a operação do sistema especialista.

A base de dados e as regras da base de conhecimento podem ser representadas em um grafo como o mostrado na figura 4.2. Neste grafo, os nós representam as possíveis bases de dados e, os arcos as regras que se aplicadas a uma determinada base de dados transformam-na em outra base de dados.

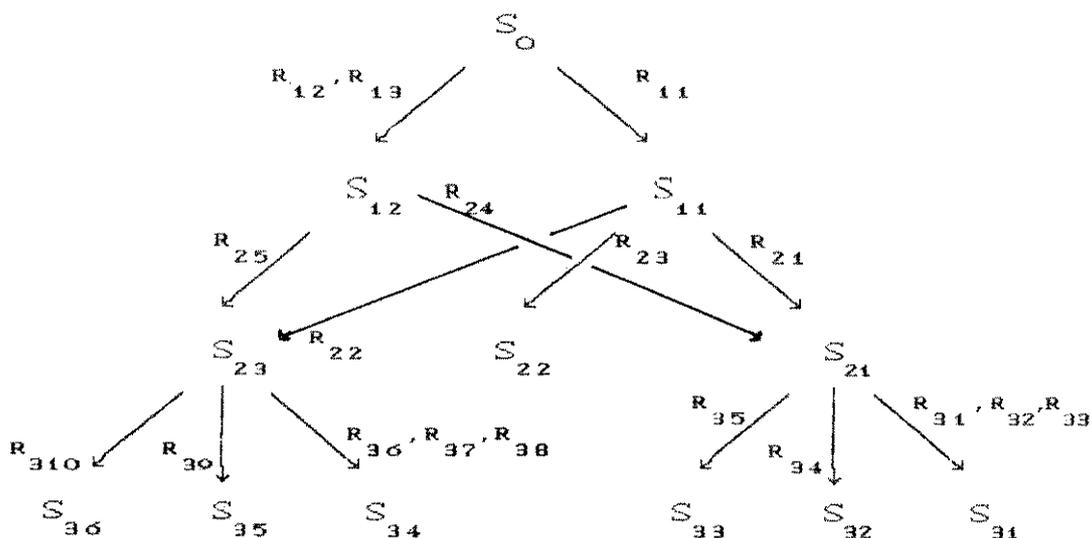


FIGURA 4.2

Representação da base de conhecimento do sistema especialista em modelagem

Neste grafo o nível zero corresponde à base de dados  $S_0$  que contém informações obtidas diretamente do módulo de identificação, a aplicação das regras  $R_{11}$ ,  $R_{12}$  ou  $R_{13}$  (regras do nível 1) a esta base de dados gera uma nova base de dados ( $S_{11}$  ou  $S_{12}$ ) no nível 1, aplicação de regras do nível 2 ( $R_{21}$ ,  $R_{22}$ ,  $R_{23}$ ,  $R_{24}$  ou  $R_{25}$ ) leva a uma das bases do nível 2 ( $S_{21}$ ,  $S_{22}$ ,  $S_{23}$ ) e a aplicação das regras do nível 3 ( $R_{31}$ ,  $R_{32}$ , ...,  $R_{310}$ ) leva a uma das bases do nível 3 ( $S_{31}$ ,  $S_{32}$ , ...,  $S_{36}$ ) que correspondem aos diagnósticos sobre o algoritmo de identificação. Um exemplo de trajetória neste grafo é

$$S_0 \xrightarrow{R_{11}} S_{11} \xrightarrow{R_{22}} S_{23} \xrightarrow{R_{39}} S_{35}$$

A construção da base de conhecimento por níveis hierárquicos facilita a expansão desta base, já que novas regras poderão ser colocadas diretamente no nível 3, ou ainda pode-se

criar novas regras mais específicas, que tenham como antecedentes conclusões obtidas no nível 3. Neste caso insere-se um nível 4. Mais níveis poderão ser então inseridos a medida que novas regras são elaboradas.

A escolha das regras, que compõem a base de conhecimento do sistema especialista em modelagem, considera o objetivo do sistema especialista que é validar, ou não, um modelo de processo obtido por um algoritmo de identificação.

Como um modelo estimado só é considerado representativo de um sistema se satisfizer determinados critérios de convergência e desempenho, algumas regras da base de conhecimento são baseadas nos critérios que estão descritos nos itens 4.3.1 e 4.3.2 a seguir.

O sistema especialista também deve ser capaz de descobrir o que está deteriorando o desempenho do algoritmo de identificação. Para tanto existem regras que analisam os parâmetros de regulagem do teste de detecção. Outras regras analisam o sinal de entrada a fim de determinar se a excitação do processo é persistente ou não.

#### 4.3.1. Critério de convergência

Este critério determina, se existir, o instante de tempo "t", a partir do qual o vetor de estimadores  $\hat{\theta}(t)$  converge para o vetor de parâmetros verdadeiros  $\theta(t)$ . Utiliza-se como critério de convergência o incremento paramétrico global que é calculado como segue.

Mede-se, em cada instante de tempo, a norma do vetor diferença entre o estimador no instante de tempo t e o estimador no instante de tempo t-1. Para tornar o critério independente da ordem de grandeza dos parâmetros divide-se o resultado obtido pela norma do vetor de médias do estimador no instante anterior. O incremento paramétrico global é então dado por

$$Vpg(t) \triangleq \frac{|| \hat{\theta}(t) - \hat{\theta}(t-1) ||}{|| \hat{\theta}(t-1) ||} \quad (4-1)$$

onde  $\hat{\theta}(t)$  é o vetor de parâmetros estimados no instante de tempo  $t$ , e  $\bar{\theta}(t)$  é o vetor de médias dos parâmetros estimados no instante de tempo  $t$ , calculado recursivamente por

$$\bar{\theta}(t) = \alpha \bar{\theta}(t-1) + (1-\alpha) \hat{\theta}(t) \quad (4-2)$$

onde  $\alpha$  é um coeficiente que permite obter uma média recursiva sobre um horizonte cujo comprimento é função de  $\alpha$ . Utiliza-se  $\alpha = 0.6$  que fornece uma média sobre 5 a 10 pontos.

Para tornar o critério independente das variações no vetor  $\hat{\theta}(t)$  devidas às perturbações aleatórias do processo, utiliza-se um valor filtrado obtido por

$$Vpg_f(t) = K Vpg_f(t-1) + (1-K) Vpg(t) \quad (4-3)$$

onde  $K$  é a constante do filtro ( $0 < K < 1$ ) e  $Vpg_f$  é o valor filtrado.

O critério de convergência estabelece que se  $Vpg_f(t) \leq \text{LIMITE}$ , o algoritmo convergiu no instante de tempo  $t$ .

Uma vez satisfeito o critério de convergência, para a validação do modelo identificado é necessário verificar se o vetor de estimadores convergiu para os parâmetros verdadeiros. Para tanto estabelece-se os critérios de desempenho.

#### 4.3.2 Critérios de desempenho

Procura-se determinar através destes critérios se o estimador está ou não polarizado. Como o método utilizado para a identificação é o dos mínimos quadrados, uma condição suficiente

para que o estimador seja não-polarizado é que o resíduo seja ruído branco [12, 15].

Dois critérios simples envolvendo o cálculo da função autocorrelação do erro previsto, e da correlação cruzada entre este erro e a entrada e entre este erro e a saída, são utilizados.

#### 4.3.2.1 Teste da correlação do erro previsto

Com este teste, procura-se determinar se o erro previsto tem uma distribuição normal  $\mathcal{N}(0, \sigma_e^2)$ .

Se a esperança  $\mathbb{E}\{\varepsilon_p(t)\}$  é nula e a autocorrelação  $\phi_{ee}(\tau) = 0$  para  $\tau = 0, 1, 2, \dots$ , o erro previsto é um ruído branco e os parâmetros estimados convergiram para os valores corretos. Do contrário testa-se a correlação entre este erro e a entrada, e entre este erro e a saída.

Para a determinação do valor médio do erro previsto, considera-se a média recursiva sobre os valores deste erro, conforme a equação abaixo:

$$\bar{\varepsilon}_p(t, \hat{\theta}) = \alpha \bar{\varepsilon}_p(t-1, \hat{\theta}) + (1-\alpha) \varepsilon_p(t, \hat{\theta}) \quad (4-4)$$

onde  $\alpha$  como no item 4.3.1.1 determina um horizonte de cálculo ( $\alpha = 0.6$ ).

O teste para correlação branca estabelece que se  $|\bar{\varepsilon}_p(t)| \geq \text{LIMITE}$ , o algoritmo está polarizado, do contrário calcula-se a correlação do erro previsto.

O cálculo da correlação do erro previsto é feito recursivamente através da equação

$$\text{cov}(\tau, t) = \alpha \text{cov}(\tau, t-1) + (1-\alpha) \varepsilon_p(t) \varepsilon_p(t-\tau) \quad (4-5)$$

para  $\tau = 0, \pm 1, \pm 2, \dots, \pm 5$ , onde  $cov(\tau, t)$  é a correlação de ordem  $\tau$  do erro previsto no instante de tempo  $t$ , e  $\alpha$  é o horizonte para cálculo da correlação recursiva. Neste trabalho foi utilizado  $\alpha = 0.9$ .

Os valores de  $cov(\tau)$  são normalizados por  $cov(0)$  e considera-se que o erro é um ruído branco se  $cov(\tau) \leq 0.1$  para  $\tau = \pm 1, \pm 2, \dots, \pm 5$ . Caso contrário o erro previsto é considerado um ruído não-branco e deve-se testar o segundo critério de desempenho.

#### 4.3.2.2 Teste da correlação cruzada

Procura-se determinar se o erro previsto é estatisticamente independente da entrada e da saída.

Se a correlação cruzada  $\phi_{ue}(\tau) = 0$  para todo  $\tau$ , o erro previsto é independente da entrada, e o estimador convergiu para os valores corretos, do contrário ele está polarizado.

O cálculo das correlações  $\phi_{ue}(\tau)$  é dado por

$$cor(\tau, t) = \alpha cor(\tau, t-1) + (1-\alpha) u(t-\tau) \varepsilon_p(t) \quad (4-6)$$

onde  $\tau = 1, 2, \dots, nb$ ,  $\alpha = 0.9$  e  $nb$  é a ordem do polinômio B do modelo descrito por 2-2.b.

O teste é estabelecido como:

i) Se  $cor(\tau) \leq LIMITE$  para todo  $\tau = 1, 2, \dots, nb$  então o erro previsto é independente da entrada e calcula-se a correlação cruzada entre este erro e a saída.

ii) Se  $\exists \tau | cor(\tau) > LIMITE$  com  $\tau = 1, 2, \dots, nb$  então o erro previsto é correlato com a entrada e o estimador está polarizado.

O cálculo da correlação cruzada entre o erro previsto e a saída do processo (  $\phi_{ue}(\tau)$  ) é dado pela equação abaixo:

$$\text{cor}(\tau, t) = \alpha \text{cor}(\tau, t-1) + (1-\alpha) y(t-\tau) \varepsilon_p(t) \quad (4-7)$$

onde  $\tau = 1, 2, \dots, n_a$ ,  $\alpha = 0.9$ , e  $n_a$  é a ordem do polinômio A do modelo descrito por 2-2.a.

O teste é estabelecido como:

i) Se  $\text{cor}(\tau) \leq \text{LIMITE}$  para todo  $\tau = 1, 2, \dots, n_a$  então o erro previsto é independente da saída e o estimador é não-polarizado.

ii) Se  $\exists \tau | \text{cor}(\tau) > \text{LIMITE}$  com  $\tau = 1, 2, \dots, n_b$  então o erro previsto é correlato com a saída e o estimador está polarizado.

#### 4.3.3 Critérios para a análise de excitação persistente

Quando o sinal de entrada não excita todos os modos do sistema, o termo  $\varphi^T(t)P(t)\varphi(t)$  nas equações 2-8 e 2-9 tende a zero [13], prejudicando a convergência do algoritmo de estimação. Nos algoritmos que utilizam fator de esquecimento diferente de um, como é o caso do algoritmo do traço adaptativo, a falta de excitação persistente pode acarretar um aumento excessivo da variância dos parâmetros estimados.

A análise da falta de excitação persistente no processo é realizada através de i) testes sobre os níveis de excitação da entrada e da saída e ii) da monitoração do valor dos elementos da matriz  $P(t)$ .

Em [11] é mostrado que para garantir a excitação do algoritmo dos mínimos quadrados é necessário que as derivadas no tempo, até a ordem  $n_b$  do sinal de entrada  $u(t)$ , e até a ordem  $(n_a-1)$  do sinal de saída  $y(t)$  sejam diferentes de zero. Existem vários métodos para o cálculo destas derivadas a partir dos dados

amostrados de entrada e saída.

Neste trabalho utiliza-se a diferenciação numérica dada pela equação 4-8 para o sinal de entrada

$$\frac{d^i u(t)}{dt} = (1-z^{-1})^i u(t) \text{ com } i \in [1, nb] \quad (4-8)$$

e pela equação 4-9 para o sinal de saída

$$\frac{d^i y(t)}{dt} = (1-z^{-1})^i y(t) \text{ com } i \in [1, na-1] \quad (4-9)$$

Analisa-se o valor dos elementos da matriz de covariância  $P(t)$  monitorando-se o traço desta matriz.

Um teste é realizado de acordo com a relação

$$\text{tr} [P(t)] > \text{LIMITE} \quad (4-10)$$

Se a condição dada por esta equação for satisfeita então pode haver problemas com a excitação no processo. O valor limite é escolhido proporcionalmente ao traço mínimo  $\text{tro}$  do algoritmo do traço adaptativo.

#### 4.4 MÓDULO DE INFERÊNCIA E EXPLICAÇÃO

Como a função do sistema especialista em modelagem é inferir um diagnóstico sobre o estado do algoritmo de identificação e, a partir deste estado, determinar sugestões para aprimorar o processo de modelagem, o módulo de inferência utiliza como estratégia de controle o encadeamento para frente. Conforme descrito no item 1.4.3.1, no encadeamento para frente não é necessário estabelecer-se a priori o objetivo a ser determinado, as regras que forem satisfeitas pela base de dados corrente são

selecionadas para a execução, e suas conclusões são os diagnósticos procurados. Se o módulo de inferência trabalhasse em encadeamento para trás, todos os possíveis diagnósticos teriam de ser testados para saber quais são satisfeitos pela base de dados corrente. Neste caso, tanto o esforço de cálculo como a quantidade de memória requerida seriam maiores que no encadeamento para frente, já que os sub-objetivos teriam que ser armazenados.

De acordo com a hierarquia da base de conhecimento descrita na seção 4.3, o módulo de inferência do sistema especialista em modelagem trabalha por níveis hierárquicos. Inicialmente são analisados os dados fornecidos pelo módulo de identificação. Neste nível as regras são mutuamente exclusivas e portanto, apenas uma regra do nível 1 é satisfeita pela base de dados e executada. Em seguida o módulo de inferência passa a analisar as regras do nível 2.

Neste nível a base de dados é formada por fatos fornecidos pelo módulo de identificação e por fatos inferidos no nível 1. Como no nível anterior, as regras são mutuamente exclusivas e o módulo de inferência seleciona e executa a regra que é satisfeita pela base de dados, passando ao nível seguinte.

No nível 3 o módulo de inferência executa, em ordem de prioridade, todas as regras que são satisfeitas pela base de dados, pois neste nível as conclusões obtidas com a execução das regras são os diagnósticos e sugestões procurados.

Em resumo, o módulo de inferência funciona em encadeamento para frente ("forward chaining") realizando a cada nível, uma busca em largura ("breadth-first-search"), onde a estratégia para a escolha e execução das regras é a descrita pelo fluxograma da figura 2.2.

A fim de criar um histórico de todos os fatos que geram os diagnósticos encontrados, o módulo de inferência, a medida que executa uma regra, armazena esta regra em um arquivo que é

utilizado pelo módulo de explicação para justificar as conclusões obtidas.

As justificativas que formam o módulo de explicação também são escritas em forma de regras de produção Se \_\_ Então. Para cada regra que compõe o módulo de inferência existe uma ou mais regras correspondentes no módulo de explicação que justifica a regra inferida.

Se o usuário questionar alguma das conclusões apresentadas, o módulo de explicação identifica no arquivo das regras executadas qual a regra que gerou a conclusão questionada e apresenta as justificativas associadas a esta regra.

Para mostrar a linha de raciocínio utilizada pelo sistema especialista, o módulo de explicação realiza o caminho inverso ao do módulo de inferência, mostrando quais regras foram executadas em ordem decrescente, isto é, da última para a primeira.

Como alguns antecedentes da última regra passam a ser sub-objetivos (conclusões) para a regra anterior, pode-se dizer que, de uma certa maneira, o módulo de explicação funciona em encadeamento para trás, embora em uma forma bastante simplificada.

#### 4.5 COMUNICAÇÃO COM O USUÁRIO

A comunicação com o usuário é feita através de mensagens onde são apresentados: i) fatos sobre o processo que está sendo modelado (por exemplo: o sistema é estacionário), ii) fatos sobre o estado do algoritmo de identificação (por exemplo: o estado é convergente), iii) diagnósticos sobre o estimador (por exemplo: o estimador está polarizado), e iv) sugestões para aprimorar o processo de identificação (por exemplo: diminua o nível de

adaptação mínimo no procedimento de detecção de não-estacionaridades).

Assim que o módulo de inferência aplica uma regra, o módulo de comunicação apresenta ao usuário a mensagem associada a esta regra, e pergunta se o usuário concorda ou não com a análise que está sendo realizada. Uma comunicação interativa é estabelecida entre o usuário e o sistema especialista a partir da resposta do usuário a esta pergunta.

Se o usuário concorda com a análise (resposta SIM), novos fatos são inferidos através da execução de uma nova regra pelo módulo de inferência. Se o usuário não concorda (resposta NÃO), o procedimento de inferência é suspenso e o módulo supervisor reativa o procedimento de identificação. Se o usuário pedir uma justificativa para esta mensagem (resposta PORQUE), o módulo de explicação é ativado, e as justificativas são apresentadas. Após a apresentação das justificativas, o módulo de comunicação pergunta novamente se o usuário concorda ou não com estas justificativas. As respostas podem ser novamente SIM, NÃO e PORQUE. Para as respostas SIM e NÃO realiza-se, respectivamente, os procedimentos de reativação do módulo de inferência ou de volta ao processo de identificação. Para a resposta PORQUE o módulo de explicação apresenta a linha de raciocínio utilizada para chegar a esta justificativa. Este processo se repete até que todas as regras já executadas sejam justificadas quando então o sistema especialista envia a seguinte comunicação ao usuário: "Não Há Mais Mensagens". O usuário pode decidir continuar o processo de inferência através da resposta SIM, ou voltar ao módulo de identificação através da resposta NÃO.

Exemplos da comunicação com o usuário e do funcionamento do sistema especialista em modelagem são mostrados na seção 4.8.

#### 4.6 MÓDULO SUPERVISOR E MÓDULO DE IDENTIFICAÇÃO

O módulo supervisor gerencia todo o sistema especialista. É responsável pela aquisição e validação dos dados da base de dados, pela ativação do módulo de inferência e explicação ou pela sua interrupção, se o usuário o desejar. A cada instante de tempo, ele reativa o algoritmo de identificação permitindo que haja ou não alterações dos parâmetros deste algoritmo, de acordo com a determinação do usuário. O módulo supervisor controla o fluxo de dados e informação em todo sistema.

No módulo de identificação estão implementados os algoritmos do Traço Adaptativo (ETEP) e o Filtro de Kalman Adaptativo (FKPE), com possibilidades de utilizar ou o algoritmo dos mínimos quadrados ou o algoritmo da variável instrumental, como descrito no capítulo 2. O usuário escolhe um deles para realizar a estimação dos parâmetros do modelo do processo que será identificado, e inicializa-o com o auxílio do sistema especialista. A cada instante de amostragem do algoritmo de estimação, o módulo de identificação fornece os valores dos parâmetros necessários para que o sistema especialista em modelagem realize os diagnósticos. Estes parâmetros são colocados na base de dados e validados pelo módulo supervisor.

Se após a apresentação dos diagnósticos, o usuário desejar fazer alterações no algoritmo de estimação, o processo de identificação é suspenso. Se as alterações forem feitas apenas nos parâmetros de regulagem do teste de detecção, ou nos limites dos testes de convergência, desempenho e excitação, a identificação é retomada no ponto onde foi suspensa. Se as alterações desejadas pelo usuário forem do tipo mudança da ordem do modelo, novo sinal de entrada ou mudança do algoritmo de estimação, o procedimento de identificação é reinicializado a partir do instante  $t = 0$ .

#### 4.7 IMPLEMENTAÇÃO COMPUTACIONAL DO SISTEMA ESPECIALISTA EM MODELAGEM

O software que compõe o sistema especialista é formado por dois programas modulares: IDENT e EXPERT. Estes programas são compilados separadamente, mas carregados juntos na memória e executados alternativamente, de acordo com a função que o sistema especialista deve realizar a cada instante: estimar os parâmetros do modelo ou fazer um diagnóstico sobre a identificação.

No programa IDENT estão implementados os algoritmos adaptativos que realizam a estimação paramétrica de um processo, correspondendo ao módulo de identificação. Este programa é codificado algorítmicamente como um programa convencional.

O programa EXPERT, onde estão implementados os módulos supervisor, base de dados, base de conhecimento, módulo de inferência e explicação e interface para comunicação com o usuário, é o sistema especialista propriamente dito. Este programa é construído como um sistema de produção: cada um dos seus procedimentos corresponde a uma das partes deste tipo de sistema (máquina de inferência, base de dados, base de conhecimentos e interface com o usuário).

Os procedimentos que formam o programa EXPERT são: i) TESTE (procedimento global) que corresponde ao módulo supervisor; ii) LEVEL1 (procedimento de inferência do nível 1); iii) LEVEL2 (procedimento de inferência do nível 2); iv) LEVEL3 (procedimento de inferência do nível 3); v) WAITRESPONSE (módulo de explicação) responsável pelas respostas às perguntas do usuário; vi) MESSAGE que realiza a comunicação com o usuário. Os procedimentos LEVEL1, LEVEL2 e LEVEL3 formam o módulo de inferência.

A seguir analisa-se o funcionamento de cada um destes procedimentos e do módulo de identificação, através de fluxogramas que correspondem à sua implementação computacional. Antes porém, faz-se algumas considerações a respeito das

variáveis e dos tipos de dados manipulados por estes programas, bem como da comunicação entre eles.

Os nomes das variáveis nos programas IDENT e EXPERT são escolhidos o mais informativo possível, assim a variável que contém o valor do traço da matriz de covariância  $P(t)$  é denominado `traço_P` e a que contém o valor do fator de esquecimento é denominada `fat_esq`, o estado do sistema é armazenado na variável `sistema` e o estado do estimador na variável `estado`, etc.

A troca de dados entre o programa de identificação e o módulo supervisor é feita via vetor de dados `DATA`. A cada instante de amostragem o algoritmo de estimação escreve neste vetor, na posição correspondente a cada variável ( por exemplo: em `DATA[11]` é armazenado o valor do limite mínimo para o teste de detecção  $J_{min}(t)$  ), os valores das variáveis que serão transferidos para o módulo supervisor. Neste módulo, o procedimento `TESTE` valida estes dados e os escreve na base de dados.

Outro vetor utilizado para a troca de informação entre procedimentos é o vetor `CHAIN`. Este vetor é uma pilha, e assim que o módulo de inferência executa uma regra, ele armazena na posição endereçada pelo apontador da pilha ( a variável inteira `TOPO` ) o número da regra executada. Os números nesta pilha são lidos depois pelo módulo de explicação, quando é necessário mostrar a linha de raciocínio utilizada para obter a algum diagnóstico.

O vetor que forma a base de dados também é uma estrutura especial de dados e é explicada a seguir.

#### 4.7.1 Implementação e operação da base de dados

Na implementação da base de dados utiliza-se para a definição do vetor `CONTEXT` que forma esta base, dois recursos

próprios da linguagem PASCAL, para a definição de estruturas de dados não convencionais: `declared scalar type` e `record type` [22].

Primeiro se estabelece que qualquer variável do tipo OBJETO contém, dados do tipo `declared scalar`, onde cada valor possível deste dado corresponde a um nome de atributo da base de dados:

```
OBJETO = ( flagn_est, lim_converg, delta_t, dmax, cor_erro,
           cor_cruz, flag_exct, flag_ordem, media_erro, n_iter,
           lim_min, lim_det, inc_pglobal, tro, traço_P, lim_excy,
           lim_excu, diagnóstico, estado, sistema)      (4-11)
```

Com esta estrutura, uma variável que seja do tipo OBJETO pode assumir qualquer um dos atributos acima, por exemplo se "o" é do tipo objeto ele pode conter o valor de `media_erro` ou de `traço_P`.

A base de dados, propriamente dita, é definida como uma estrutura de dados do tipo `record` com dois identificadores, um associado ao valor numérico [`valor`] do atributo e outro associado a sua validade [`valid`], e suas posições são acessadas por variáveis do tipo OBJETO. O vetor `CONTEXT` que corresponde à base de dados é portanto uma variável do tipo `record` definida como:

```
var CONTEXT: array[objetol of record
                    valid:boolean;
                    valor:real;
```

As operações sobre a base de dados são de dois tipos: leitura e escrita.

Para a leitura de um valor no vetor `CONTEXT`, define-se a função `RETRIEVE`, que testa a validade do atributo armazenado na base de dados, e salva seu valor numérico na variável `RETRIEVE`,

se o dado é válido. A implementação desta função é a seguinte:

```
FUNCTION retrieve (s:objeto):Real;  
BEGIN  
  IF context[s].valid THEN retrieve:=context[s].valor  
END;
```

onde a variável s do tipo objeto pode assumir qualquer um dos valores especificados em OBJETO em 4-11.

Para se escrever algum valor no vetor CONTEXT utiliza-se o procedimento ASSERT. A operação de escrita na base de dados corresponde à execução da conclusão de uma determinada regra e assim o procedimento ASSERT é sempre chamado pelo módulo de inferência quando este executa alguma regra. O funcionamento da operação escrita é como segue:

```
PROCEDURE assert (s:objeto; v:real; ruleNo:integer);  
BEGIN  
  Chain[topo]:= ruleNo;          (* armazena na pilha CHAIN o  
                                número da regra executada *)  
  with context[s] do  
    BEGIN  
      valor:=v;  
      valid:=true  
    END  
END;
```

#### 4.7.2 Implementação do módulo supervisor

O supervisor funciona de acordo com o seguinte procedimento: i) recebe do usuário dados para o procedimento de identificação, ii) verifica se estes dados estão corretos, iii) em caso afirmativo ativa o módulo de identificação, iv) em caso negativo requisita novos dados ao usuário, v) após uma iteração do algoritmo de estimação, recebe dados do módulo de identificação e, vi) faz a validação destes dados, colocando-os

na base de dados, vii) ativa em seguida o módulo de inferência; viii) quando cessa o procedimento de inferência, consulta o usuário sobre modificações no algoritmo de estimação, ix) se as modificações desejadas exigirem uma reinicialização do algoritmo de identificação o módulo supervisor volta para o passo inicial, x) se não ele volta para o passo iii.

Este funcionamento é ilustrado no fluxograma da figura 4.3, onde no passo em que ocorrem blocos ligados ao diagrama por linhas pontilhadas, significa que neste passo o módulo supervisor é suspenso, passando o controle ao procedimento que está representado no bloco e que passa a ser executado; ao final desta execução o controle é devolvido ao módulo supervisor no mesmo ponto em que foi suspenso.

#### 4.7.3 Implementação do módulo de identificação

O módulo de identificação é chamado a partir do supervisor e, de acordo com os dados recebidos, é escolhido um dos algoritmos de estimação. Este algoritmo é inicializado e realiza-se a estimação paramétrica do processo através dos passos ilustrados no fluxograma da figura 4.4. Ao final da estimação, os dados obtidos são escritos no vetor DATA, para serem transferidos para a base de dados.

#### 4.7.4 Implementação do módulo de inferência e base de conhecimento

O módulo de inferência é composto de três procedimentos independentes, um para cada nível hierárquico da base de conhecimentos. O procedimento para o nível 1 é denominado LEVEL1, para o nível 2 LEVEL2 e para o nível 3 LEVEL3. O módulo de inferência é chamado a partir do módulo supervisor, que após validar a base de dados e inicializar o apontador da pilha CHAIN (topo), transfere o controle para LEVEL1 (ver fluxograma 4.3).

O fluxograma do procedimento LEVEL1 é mostrado na figura 4.5: inicialmente escolhe-se uma regra da base de conhecimentos no nível 1, lê-se o valor dos seus antecedentes através da função RETRIEVE, e testa-se se estes valores satisfazem as condições da regra escolhida. Em caso negativo volta-se ao início e escolhe-se outra regra. Em caso afirmativo executa-se esta regra através de uma chamada ao procedimento ASSERT; em seguida é chamado o procedimento MESSAGE a fim de enviar-se uma mensagem ao usuário sobre a regra executada e chama-se o procedimento WAITRESPONSE para justificar esta regra. Após a comunicação com o usuário passa-se para o nível 2 da base de conhecimento através da chamada do procedimento LEVEL2.

O funcionamento de LEVEL2 é exatamente igual ao de LEVEL1, exceto que no último passo do fluxograma da figura 4.5, passa-se para o nível 3 da base de conhecimento através da chamada do procedimento LEVEL3.

O procedimento de inferência para o terceiro nível é diferente dos procedimentos para os níveis 1 e 2, já que a base de conhecimento neste nível não é formada apenas de regras mutuamente exclusivas como nos níveis anteriores, e todas as regras que forem satisfeitas pela base de dados corrente devem ser executadas o que não acontece em LEVEL1 e LEVEL2, onde executa-se apenas uma regra por vez: a de maior prioridade que é satisfeita pela base de dados corrente.

O fluxograma deste procedimento é mostrado na figura 4.6 e os passos de escolha, teste e execução das regras e a chamada de MESSAGE e WAITRESPONSE é como em LEVEL1 e LEVEL2. Após a execução destes passos é verificado se todas as regras deste nível foram testadas; em caso afirmativo retorna-se ao módulo supervisor no ponto subsequente à chamada de LEVEL1 (label 99), em caso negativo reinicia-se o processo de busca e seleção de uma regra a ser executada (volta-se ao início do procedimento).

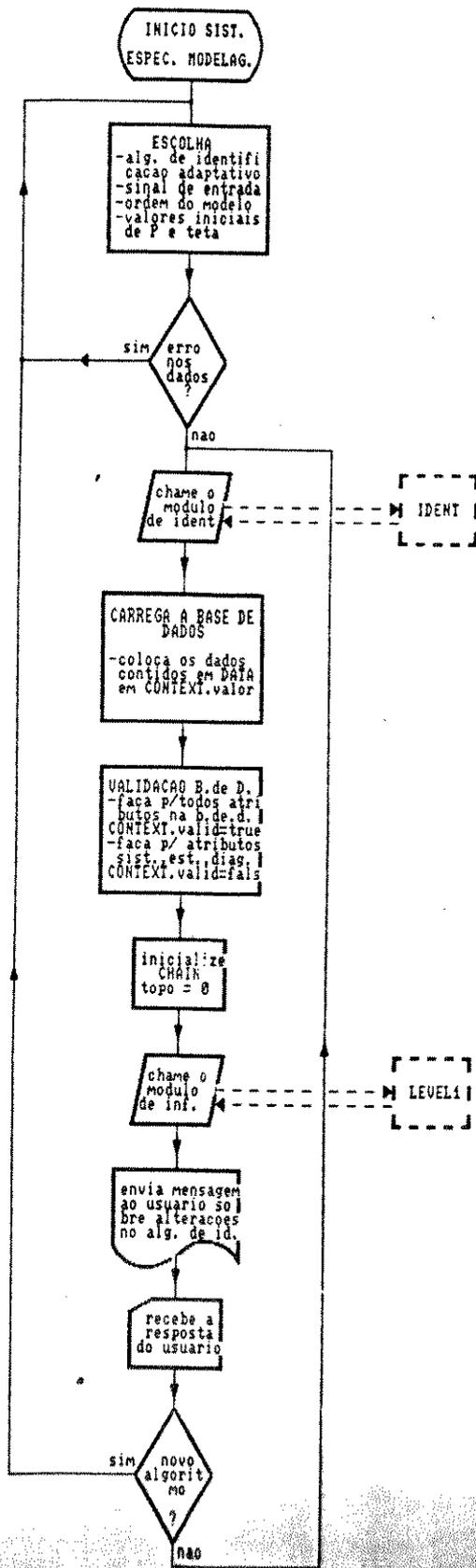


FIGURA 4.3  
Modulo Supervisor

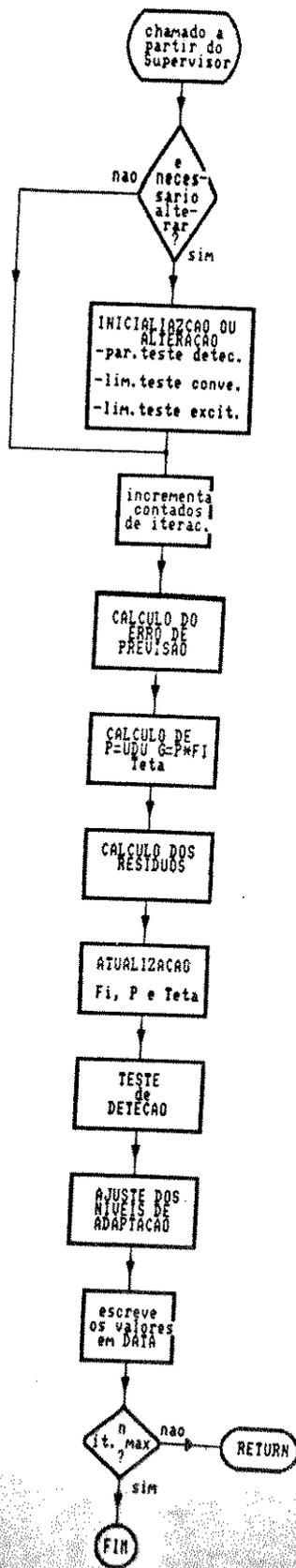


FIGURA 4.4  
Módulo de Identificação



FIGURA 4.5.  
Módulo de Inferência  
(Nível 1 ou Nível 2)

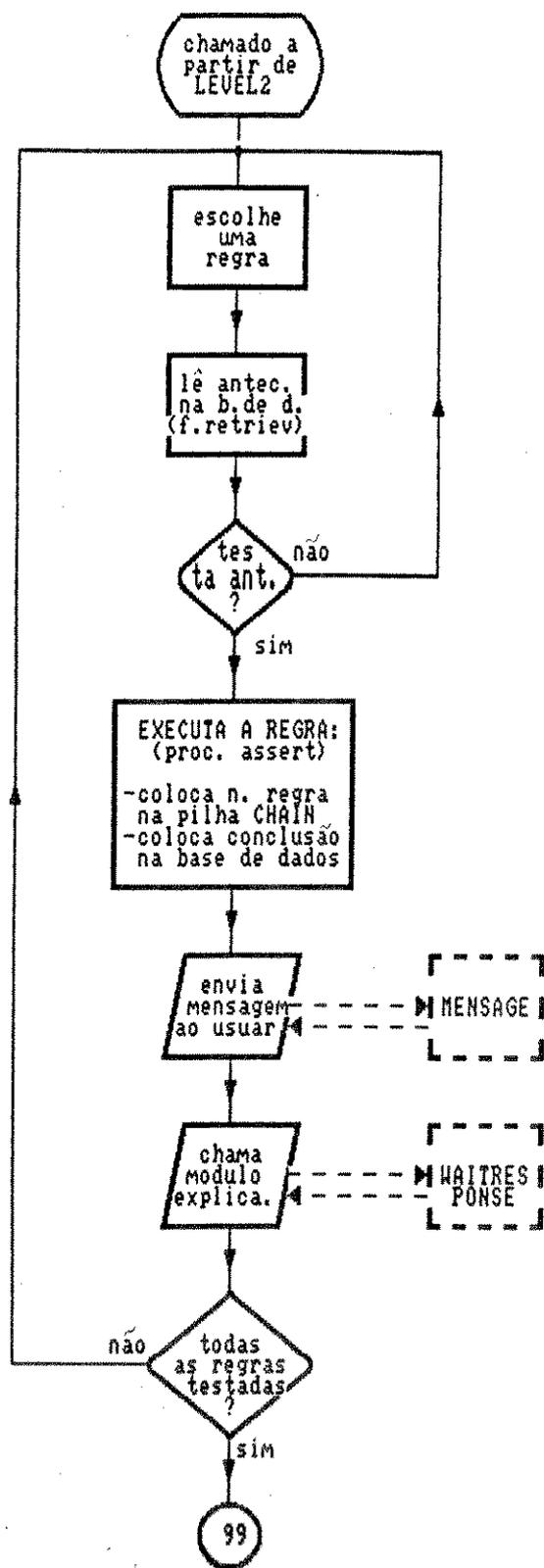


FIGURA 4.6  
Módulo de Inferência  
(Nível 3)

#### 4.7.5 Implementação do módulo de explicação e comunicação com o usuário

O procedimento MESSAGE funciona segundo o fluxograma da figura 4.7 onde o número da regra executada é lido na pilha CHAIN e a mensagem associada a este número é enviada ao usuário.

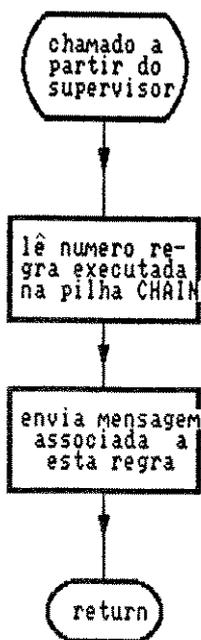
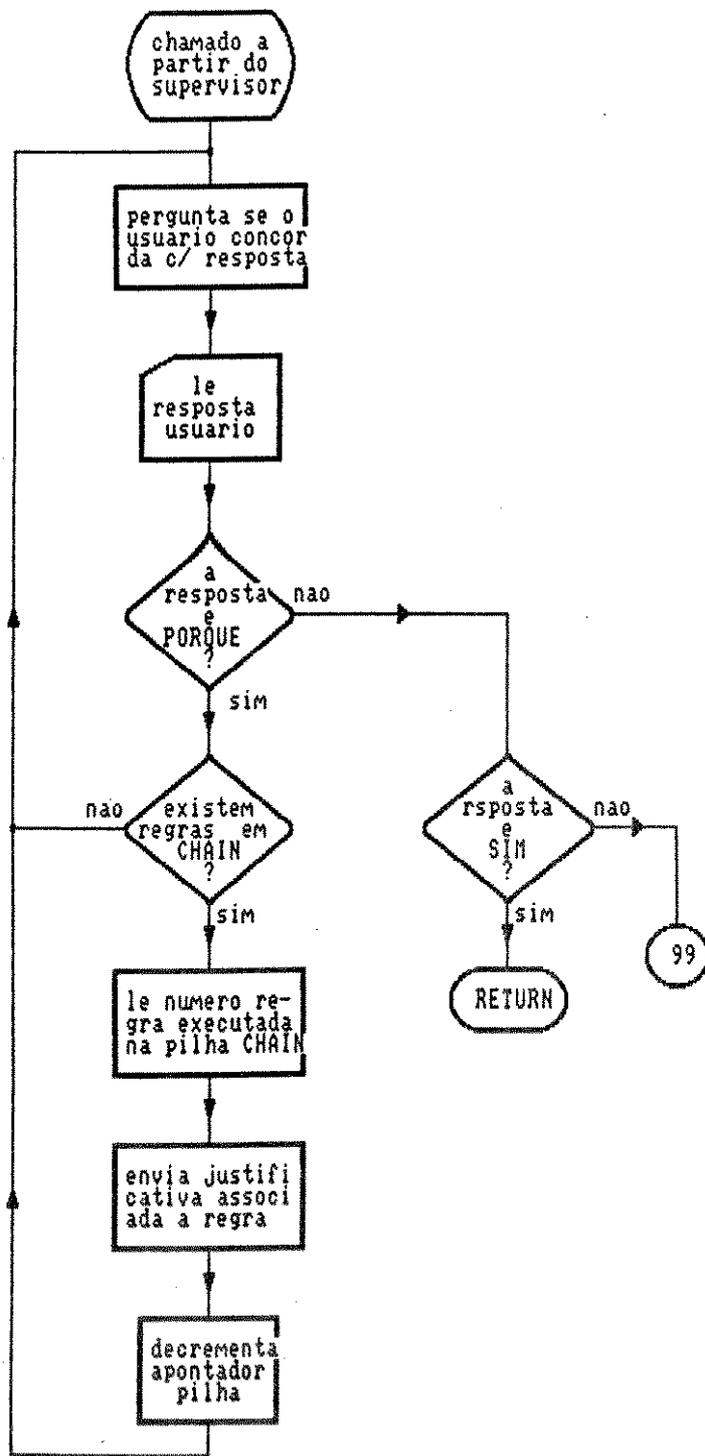


FIGURA 4.7  
Procedimento MESSAGE

Os procedimentos para explicação das regras executadas são feitos por WAITRESPONSE que também estabelece uma comunicação interativa com o usuário.

O funcionamento deste procedimento é descrito pelo fluxograma da figura 4.8, onde inicialmente é perguntado se o



**FIGURA 4.8**  
Modulo de Explicacao

usuário concorda ou não com a mensagem apresentada, em seguida a resposta do usuário é lida e verifica-se o tipo de resposta recebida: se a resposta for NÃO retorna-se ao módulo supervisor no ponto subsequente à chamada do procedimento LEVEL1 ( label 99), se a resposta for SIM volta-se ao módulo de inferência e se a resposta for PORQUE verifica-se se existem regras na pilha CHAIN. Em caso afirmativo lê-se o número da regra que está armazenado no topo da pilha, envia-se a justificativa associada a esta regra e decrementa-se o apontador da pilha CHAIN (topo), voltando-se ao início, onde é perguntado se o usuário concorda ou não com a mensagem enviada. Se não houver mais regras na pilha CHAIN é porque todas as regras executadas já foram justificadas e envia-se uma mensagem ao usuário avisando-o que ele deve ou parar o processo de inferência ( resposta NÃO ) ou continuá-lo (resposta SIM ), e volta-se ao início.

#### 4.8 EXEMPLOS DA UTILIZAÇÃO DO SISTEMA ESPECIALISTA EM MODELAGEM NA IDENTIFICAÇÃO DE PROCESSOS SIMULADOS

O sistema especialista em modelagem é utilizado na identificação de um processo simulado de acordo com a equação 4-12.

$$y(t) + a_1(t) y(t-1) = b_0(t) u(t-2) + e(t) \quad (4-12)$$

O processo como descrito no capítulo 3, sofre uma variação brusca do modelo 1 para o modelo 2 ( MOD 1 + MOD 2 ), no instante  $t_v = 50$ , isto é, os parâmetros  $a_1$  e  $b_0$  na equação 4-12, assumem os seguintes valores:

$$a_1(t) = - 0.905; \quad b_0(t) = 1.0 \quad t \in [0,50] \quad (4-13)$$

$$a_1(t) = - 0.980; \quad b_0(t) = 0.4 \quad t \in [51,100] \quad (4-14)$$

A entrada  $u(t)$  e a perturbação  $e(t)$  são seqüências brancas de variáveis aleatórias com média zero e desvios padrão  $\sigma_u = 1.0$  e  $\sigma_e = 0.1$ , respectivamente.

As condições iniciais para o algoritmo de identificação são  $\hat{\theta}(0) = 0$  e  $P(0) = 100 \times I$ .

O algoritmo utilizado foi o do traço adaptativo. Os parâmetros de regulagem do teste de detecção são ajustados, segundo as conclusões do capítulo 3:

Horizontes de cálculo:  $N_1 = 10$ ,  $N_c = 5$  e  $\tau = 0$ ;

Níveis de adaptação:  $tr_{\min} = 0.01$ ,  $tr_{\text{int}} = 0.1$  e  $tr_{\max} = 1.0$ ;

Limites de detecção:  $J_{\max} = 8.0$  e  $J_{\min} = 4.0$ .

As figuras 4.9 e 4.10 mostram a evolução dos parâmetros estimados. Os diagnósticos obtidos com o sistema especialista em modelagem, nos instantes marcados ( iterações 25, 42, 56 e 98 ) são apresentados a seguir para ilustrar o funcionamento do sistema.

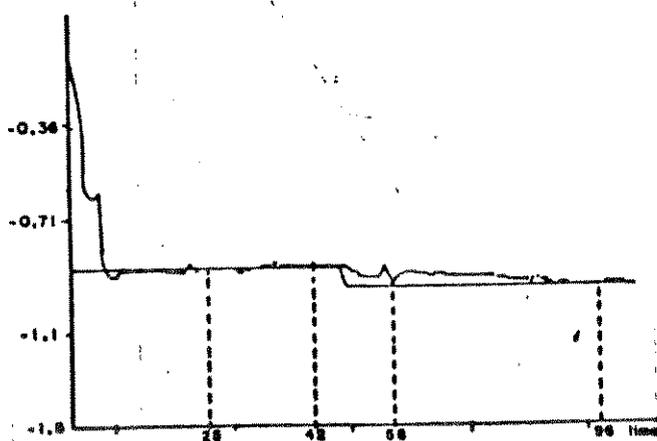


FIGURA 4.9  
Parâmetro  $\hat{a}_1(t)$

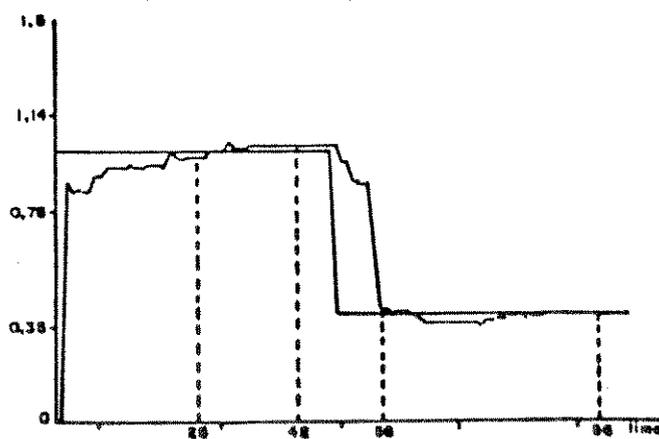


FIGURA 4.10  
parâmetro  $\hat{b}_0(t)$

SISTEMA SUPERVISOR EXPERIMENTAL (IT = 25)

-- Conclusão : Pela regra 3 descobrimos que:

O SISTEMA É ESTACIONÁRIO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 3 diz que :

SE O CRITÉRIO DE DETECÇÃO < 4.00

ENTÃO O SISTEMA É ESTACIONÁRIO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- S

-- Conclusão : Pela regra 5 descobrimos que:

O ESTADO É NÃO CONVERGENTE

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 5 diz que :

SE O SISTEMA É ESTACIONÁRIO

E A NORMA DA VARIACÃO DOS PARÂMETROS > 0.010

ENTÃO A IDENTIFICAÇÃO NÃO ESTÁ CONVERGINDO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?) --

-- S

Deseja alterar parâmetros no algoritmo de estimação?

digite a sua escolha:

(-1) - não deseja alterar

(00) - parâmetros do teste de deteção

(01) - limites p/os critérios de convergência ou excitação

-- -1

SISTEMA SUPERVISOR EXPERIMENTAL (IT = 42)

-- Conclusão : Pela regra 3 descobrimos que:

O SISTEMA É ESTACIONÁRIO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 3 diz que :

SE O CRITÉRIO DE DETECÇÃO < 4.00

ENTÃO O SISTEMA É ESTACIONÁRIO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- S

-- Conclusão : Pela regra 4 descobrimos que:

O ESTADO É CONVERGENTE

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 4 diz que :

SE O SISTEMA É ESTACIONÁRIO

E A NORMA DA VARIACÃO DOS PARÂMETROS < 0.010

ENTÃO A IDENTIFICAÇÃO ESTÁ CONVERGINDO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?) --

-- S

-- Conclusão : Pela regra 12 descobrimos que:

OS PARÂMETROS ESTIMADOS ESTÃO CONVERGINDO

MAS NÃO ATINGIRAM AINDA OS VALORES CORRETOS

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 12 diz que :

SE A IDENTIFICAÇÃO ESTÁ CONVERGINDO  
E O ERRO PREVISTO É NÃO CORRELATO COM A ENTRADA  
E A MÉDIA DO ERRO PREVISTO > 0.010  
E A MÉDIA DO ERRO PREVISTO < 0.100  
ENTÃO O ESTIMADOR ESTÁ CONVERGINDO MAS NÃO ATINGIU  
OS VALORES CORRETOS

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?) --

-- S

Deseja alterar parâmetros no algoritmo de estimação?

digite a sua escolha:

(-1) - não deseja alterar

(00) - parâmetros do teste de detecção

(01) - limites p/os critérios de convergência ou excitação

-- -1

#### SISTEMA SUPERVISOR EXPERIMENTAL (IT = 58)

-- Conclusão : Pela regra 1 descobrimos que:

O SISTEMA É NÃO ESTACIONÁRIO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 1 diz que :

SE O CRITÉRIO DE DETECÇÃO > 8.00  
ENTÃO O SISTEMA É NÃO ESTACIONÁRIO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- S

-- Conclusão : Pela regra 6 descobrimos que:

O ESTADO É INDETERMINADO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 6 diz que :

SE O SISTEMA É NÃO ESTACIONARIO

E  $\Delta T = IT_{ATUAL} - IT_{VARIAC\tilde{A}O} < 10$

ENTAO O ESTIMADOR NÃO ATINGIU AINDA UM REGIME  
ESTÁVEL APÓS A VARIACÃO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?) --

-- S

Deseja alterar parâmetros no algoritmo de estimação?

digite a sua escolha:

(-1) - não deseja alterar

(00) - parâmetros do teste de detecção

(01) - limites p/os critérios de convergência ou excitação

-- -1

SISTEMA SUPERVISOR EXPERIMENTAL (IT = 96)

-- Conclusão : Pela regra 3 descobrimos que:

O SISTEMA É ESTACIONÁRIO

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 3 diz que :

SE O CRITÉRIO DE DETECÇÃO < 4.00  
ENTÃO O SISTEMA É ESTACIONÁRIO

-- VOCÊ CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- S

-- Conclusão : Pela regra 4 descobrimos que:

O ESTADO É CONVERGENTE

-- VOCÊ CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 4 diz que :

SE O SISTEMA É ESTACIONÁRIO  
E A NORMA DA VARIACÃO DOS PARÂMETROS < 0.010  
ENTÃO A IDENTIFICAÇÃO ESTÁ CONVERGINDO

-- VOCÊ CONCORDA ? ([S]im, [N]ão, [P]orque?) --

-- S

-- Conclusão : Pela regra 10 descobrimos que:

OS PARÂMETROS ESTIMADOS CONVERGIRAM  
PARA OS VALORES CORRETOS

-- VOCÊ CONCORDA ? ([S]im, [N]ão, [P]orque?)

-- P

Porque a regra 10 diz que :

SE A IDENTIFICAÇÃO ESTÁ CONVERGINDO  
E O ERRO PREVISTO É NÃO CORRELATO COM A ENTRADA  
E A MÉDIA DO ERRO PREVISTO < 0.010  
ENTÃO O ESTIMADOR CONVERGIU PARA OS VALORES CORRETOS

-- VOCE CONCORDA ? ([S]im, [N]ão, [P]orque?) --

-- S

Deseja alterar parâmetros no algoritmo de estimação?

digite a sua escolha:

(-1) - não deseja alterar

(00) - parâmetros do teste de detecção

(01) - limites p/os critérios de convergência ou excitação

-- -1

Da comparação entre o desempenho ilustrado nas figuras 4.9 e 4.10 e os diagnósticos apresentados pelo sistema especialista em modelagem, pode-se observar que:

i) No instante de tempo  $t=25$ , o parâmetro  $\hat{a}_1$  (figura 4.9) já havia convergido para o seu valor real, no entanto o parâmetro  $\hat{b}_0$  ainda apresentava uma variância considerável (figura 4.10). Neste caso o Sistema Supervisor concluiu que o estado era não convergente uma vez que a norma da variação dos parâmetros não estava dentro dos limites aceitáveis.

ii) No instante  $t=42$ , de acordo com as figuras 4.9 e 4.10, os parâmetros  $\hat{a}_1$  e  $\hat{b}_0$  já não apresentam variações, no entanto existe um erro entre o valor estimado e o valor real para o parâmetro  $b_0$  (polarização de  $b_0$ ). O diagnóstico do Sistema Supervisor para este instante, foi: o estimador é convergente mas os valores estimados ainda não são corretos. Este diagnóstico é baseado na análise de que o erro previsto é não correlato com a entrada mas sua média não pode ser considerada nula, embora apresente valores pequenos. Se, ao contrário, a média do erro previsto estivesse acima de um limite máximo e houvesse correlação entre este erro e a entrada, o diagnóstico seria que o estimador estaria polarizado e, o Sistema Supervisor recomendaria a utilização do método da variável instrumental.

iii) No instante  $t=58$ , os estimadores  $\hat{a}_1$  e  $\hat{b}_0$  apresentam uma variância alta pois sofreram uma variação no instante  $t=50$ , e o

algoritmo de estimação ainda se encontra em sua fase de regulagem. Neste caso, o Sistema Supervisor diagnosticou que o processo é não-estacionário, e que o estado do estimador é indeterminado pois ainda não decorreu um tempo igual ao tempo de estabilização, desde que foi detetada a não-estacionaridade.

iv) No instante  $t=98$ , pelas figuras 4.9 e 4.10, os parâmetros estimados convergiram para os parâmetros verdadeiros. Neste caso o Sistema Supervisor diagnosticou que o processo é novamente estacionário, e o estimador é convergente. A partir do valor da média do erro previsto e da análise da correlação cruzada entre este erro e a entrada e saída também foi concluído que os parâmetros estimados estão corretos.

Outros exemplos, onde utilizam-se uma perturbação não-branca no processo, entrada pouco excitante ou relação sinal-ruído pequena, têm sido simulados e os diagnósticos apresentados pelo Sistema Supervisor têm sido corretos.

#### 4.9 CONCLUSÕES

Neste capítulo desenvolveu-se a estrutura do sistema especialista em modelagem. Os diversos módulos que formam este sistema : base de conhecimento, base de dados, módulo de inferência e explicação, módulo supervisor, módulo de identificação, e interface com o usuário foram descritos. A implementação computacional destes módulos foi apresentada em forma de fluxogramas.

Os fatos na base de dados são representados através de duplas associativas do tipo <ATRIBUTO> , <VALOR> porque esta forma de representação simplifica a descrição dos fatos e facilita a implementação e operação do vetor que forma a base de dados.

A base de conhecimento é composta por regras do tipo SE  
— ENTÃO, organizadas hierarquicamente em três níveis. O  
conhecimento representado por estas regras foi obtido a partir de  
simulações dos algoritmos de identificação.

O módulo de inferência utiliza como estratégia de  
controle o encadeamento para frente ,e realiza uma busca em  
largura, para selecionar as regras que devem ser executadas em  
cada passo. Esta estratégia requer menor esforço computacional e  
menor quantidade de memória que as outras estratégias de  
controle.

Finalizando um exemplo do funcionamento do sistema  
especialista foi apresentado. Mostraram-se interações típicas  
entre o sistema especialista e o usuário, para a modelagem  
correta de um processo simulado.

## CAPÍTULO 5

### CONCLUSÕES

#### 5.1 SÍNTESE DO TRABALHO

O sistema especialista proposto neste trabalho realiza a supervisão de algoritmos de modelagem para sistemas variantes no tempo. O sistema pode auxiliar inicialmente, o usuário na seleção do algoritmo de identificação mais adequado para cada tipo de aplicação considerada. O sistema também analisa o desempenho do algoritmo escolhido e, caso necessário, são apresentadas ao usuário sugestões para modificações nos parâmetros do estimador, a fim de melhorar o desempenho ou obter-se um modelo que represente adequadamente o processo em estudo.

O sistema especialista proposto e implementado é um sistema de produção, possuindo base de dados, base de conhecimentos, com o conhecimento representado na forma de regras de produção do tipo SE \_\_ ENTÃO, e uma máquina de inferência que trabalha em modo de encadeamento para frente e realiza uma busca em largura para seleção das regras a executar. O sistema especialista contém um módulo supervisor que administra o sistema, uma interface homem - máquina que permite a comunicação com o usuário e um módulo com os algoritmos de identificação.

Para facilidade de expansão, manutenção, correção de erros e utilização o sistema foi implementado modularmente. Cada módulo que constitui o software do sistema especialista realiza uma tarefa distinta, não interferindo na tarefa dos demais módulos.

Além do desenvolvimento e implementação do sistema especialista propriamente dito, neste trabalho também foram propostos dois algoritmos de estimação capazes de detetar e rastrear variações nos parâmetros do processo em estudo: i) o algoritmo do traço adaptativo, que utiliza o estimador com traço limitado, e o critério do erro previsto para deteção de não-estacionaridades (ETEP); ii) o filtro de Kalman adaptativo, que utiliza o método de busca aleatória para estimar os parâmetros e o critério dos parâmetros estimados como método de deteção (FKPE).

## 5.2 CONCLUSÕES E RECOMENDAÇÕES

A partir dos resultados das simulações realizadas, foram estabelecidas condições para a utilização dos algoritmos adaptativos:

i) Para não-estacionaridades lentas ou de pequena amplitudes e relações sinal-ruído pequenas deve-se utilizar o algoritmo do filtro de Kalman adaptativo (FKPE) que neste caso, apresenta uma maior probabilidade de deteção de não-estacionaridades.

ii) Para relações sinal-ruído elevadas e não-estacionaridades de grande amplitude e rápidas deve-se utilizar o algoritmo do traço adaptativo (EETEP), que apresenta um melhor desempenho.

iii) O algoritmo do traço adaptativo (EETEP) também deve ser utilizado em caso de variações lentas nos parâmetros se a relação sinal-ruído for elevada.

iv) Se o processo identificado sofre não-estacionaridades sucessivas, deve-se utilizar o algoritmo do traço adaptativo (EETEP) pois este algoritmo apresenta um menor atraso na deteção e

uma convergência mais rápida para os novos valores dos parâmetros.

Além da natural utilização do sistema especialista em modelagem de processos, uma outra aplicação deste sistema seria na detecção de falhas em processos industriais. A falha em um processo está normalmente associada à variação de seus parâmetros. Assim, o Sistema Supervisor detectaria os parâmetros que variam e a partir desta informação poderia emitir um diagnóstico sobre as causas da falha.

Como futuros trabalhos sugere-se:

i) A aplicação deste sistema em tempo real, que pode ser conseguida transformando os diagnósticos gerados em um conjunto de regras de produção, que são executadas automaticamente para a identificação de um processo real.

ii) A expansão deste sistema para fornecer também diagnósticos sobre o estado de um controlador atuando no processo. Os diagnósticos estariam relacionados com o modelo do sistema, com o modelo do controlador, com a análise do seu desempenho, etc.

## APÊNDICE A

### ALGORITMO COM FATORIZAÇÃO U-D

Neste apêndice é demonstrado a obtenção do algoritmo recursivo de fatorização U-D de Bierman. A demonstração está de acordo com a encontrada em [05] e [13].

Considerando a equação A-1 para atualização da matriz de covariância  $P(t)$  dos mínimos quadrados e  $P(t) \in \mathbb{R}^{n \times n}$ , onde  $n$  é a ordem do estimador  $\hat{\theta}(t)$ .

$$P(t) = \left[ \begin{array}{c} P(t-1) - \frac{P(t-1) \varphi(t) \varphi^T(t) P(t-1)}{\lambda(t) + \varphi^T(t) P(t-1) \varphi(t)} \\ \lambda(t) \end{array} \right]^{-1} \lambda(t) \quad (A-1)$$

Suponha que  $P(t-1)$  é fatorada na forma U-D como em A-2

$$P(t-1) = U(t-1) D(t-1) U^T(t-1) \quad (A-2)$$

onde  $U(t-1)$  é uma matriz triangular superior com todos os elementos da diagonal unitários e  $D(t-1)$  é uma matriz diagonal ( $U(t-1), D(t-1) \in \mathbb{R}^{n \times n}$ ).

A matriz  $P(t)$  deve ser fatorada de forma semelhante

$$P(t) = U(t) D(t) U^T(t) \quad (A-3)$$

Substituindo estas formas fatoradas na equação A-1, tem-se

$$U(t) D(t) U^T(t) = \frac{1}{\lambda(t)} \left[ \begin{array}{c} U(t-1) D(t-1) U^T(t-1) - \frac{U(t-1) D(t-1)}{\lambda(t) +} \\ \frac{U^T(t-1) \varphi(t) \varphi^T(t) U(t-1) D(t-1) U^T(t-1)}{\varphi^T(t) U(t-1) D(t-1) U^T(t-1) \varphi(t)} \end{array} \right] \quad (A-4)$$

Formando-se os seguintes vetores auxiliares

$$f(t) = U^T(t-1) \varphi(t) \quad (A-5.a)$$

$$h(t) = D(t-1) f(t) \quad (A-5.b)$$

$$s(t) = \lambda(t) + \varphi^T(t) P(t-1) \varphi(t) = \lambda(t) + f^T(t) h(t) \quad (A-5.c)$$

e substituindo-os na equação A-4, obtém-se A-6

$$U(t) D(t) U^T(t) = \left[ U(t-1) D(t-1) U^T(t-1) - \frac{U(t-1) h(t) h^T(t) U^T(t-1)}{s(t)} \right] * \\ * 1/\lambda(t) \quad (A-6)$$

que pode ser rescrita como em A-7.

$$U(t) D(t) U^T(t) = U(t-1) \left[ D(t-1) - \frac{h(t) h^T(t)}{s(t)} \right] U^T(t-1) * 1/\lambda(t) \quad (A-7)$$

O termo entre colchetes na equação A-7 pode ser fatorado como na equação A-8,

$$D(t-1) - \frac{h(t) h^T(t)}{s(t)} = \bar{U}(t) \bar{D}(t) \bar{U}^T(t) \quad (A-8)$$

e substituindo a equação A-8 em A-7, obtém-se a equação A-9.

$$U(t) D(t) U^T(t) = U(t-1) \bar{U}(t) \bar{D}(t) \bar{U}^T(t) U^T(t-1) * 1/\lambda(t) \quad (A-9)$$

De A-9 tem-se que

$$U(t) = U(t-1) \bar{U}(t) \quad (A-10.1)$$

$$D(t) = \bar{D}(t) / \lambda(t) \quad (A-10.2)$$

E assim a fatorização  $P(t) = U(t) D(t) U^T(t)$  está agora reduzida à fatorização dada na equação A-8.

Para simplificar a notação, reescreve-se as matrizes  $\bar{U}(t)$ ,  $\bar{D}(t)$  e  $D(t)$  como:

$$\bar{U}(t) = [ \bar{U}_1 \dots \bar{U}_i \dots \bar{U}_n ] = \begin{bmatrix} 1 & \bar{u}_{12} & \dots & \bar{u}_{1i} & \dots & \bar{u}_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 1 & \dots & u_{in} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & \dots & \dots & 1 \end{bmatrix}$$

$$\bar{D}(t) = \begin{bmatrix} \bar{d}_1 & & & & & \\ & \cdot & & & & \\ & & \bar{d}_i & & & \\ & & & \cdot & & \\ & 0 & & & \cdot & \\ & & & & & \bar{d}_n \end{bmatrix} \quad \text{e} \quad D(t-1) = \begin{bmatrix} d_1 & & & & & \\ & \cdot & & & & \\ & & d_i & & & \\ & & & \cdot & & \\ & 0 & & & \cdot & \\ & & & & & d_n \end{bmatrix}$$

onde  $u_{ij}$ ,  $\bar{d}_i$  e  $d_i \in \mathbb{R}$ , e os vetores  $f(t)$  e  $h(t)$  como

$$f^T(t) = [ f_1 \dots f_i \dots f_n ]$$

$$h^T(t) = [ h_1 \dots h_i \dots h_n ]$$

$$\text{e } s(t) = s = s_n.$$

Com esta notação a equação A-8 é reescrita na equação A-9,

$$\sum_{i=1}^n \bar{U}_i \bar{U}_i^T \bar{d}_i = \sum_{i=1}^n d_i e_i e_i^T - [h h^T * 1/s] \quad (\text{A-11})$$

onde  $e_i$  é o  $i$ -ésimo vetor unitário.

Para resolver a fatorização A-8 é preciso determinar a partir de A-11,  $\bar{d}_i$  e  $\bar{U}_i$ , dados  $d_i$ ,  $f$ ,  $h$  e  $s$ . Para tanto considera-se a matriz  $M_n$  definida em A-12,

$$M_n = \bar{d}_n \bar{U}_n \bar{U}_n^T - d_n e_n e_n^T + 1/s_n h h^T \quad (\text{A-12})$$

e introduz-se o vetor

$$C_n^T = [ c_{n+1,1} \dots c_{n+1,i} \dots c_{n+1,n-1} \ 0 ] = [ h_1 \dots h_i \dots h_n ] \quad (\text{A-13})$$

Escolhendo

$$\bar{d}_n = d_n - c_{n,n}^2 / s_n \quad (\text{A-14. a})$$

$$\bar{u}_{nn} = 1 \quad (\text{A-14. b})$$

$$\bar{u}_{ij} = -c_{n,n} c_{n,i} / \bar{d}_n s_n \text{ com } i = 1, \dots, n-1 \quad (\text{A-14. c})$$

é fácil verificar que a última linha e a última coluna de  $M_n$  serão nulas, que o produto  $\bar{U}_n \bar{U}_n^T$  exceto pela última linha e última coluna valerá

$$\begin{aligned} \bar{U}_n \bar{U}_n^T &= c_{n,n}^2 / \bar{d}_n^2 s_n^2 \begin{bmatrix} c_{n,1} \\ \vdots \\ c_{n,n-1} \\ \bar{d}_n s_n / c_{n,n} \end{bmatrix} \begin{bmatrix} c_{n,1}^2 \dots c_{n,n-1}^2 & d_n s_n / c_{n,n} \end{bmatrix} \\ &= \begin{bmatrix} c_{n,n}^2 / \bar{d}_n^2 s_n^2 \end{bmatrix} C_{n-1} C_{n-1}^T \end{aligned} \quad (\text{A-15})$$

pois de acordo com A-13,  $C_{n-1}^T = [c_{n,1} \dots c_{n,i} \dots c_{n,n-1} \ 0]$ , e o produto  $C_n C_n^T$  também excetuando a última linha e última coluna valerá

$$C_n C_n^T = \begin{bmatrix} c_{n,1} \\ \vdots \\ c_{n,n} \end{bmatrix} \begin{bmatrix} c_{n,1} & \dots & c_{n,n} \end{bmatrix} = C_{n-1} C_{n-1}^T \quad (\text{A-16})$$

Como o produto  $e_n e_n^T$  exceto pela última linha e última coluna é a matriz nula, a matriz  $M_n$  dada em A-12 pode ser reescrita de acordo com as equações A-14, A-15 e A-16 como

$$M_n = \bar{d}_n - \frac{c_{n,n}^2}{\bar{d}_n^2 s_n^2} C_{n-1} C_{n-1}^T + C_{n-1} C_{n-1}^T / s_n$$

$$M_n = \left[ \frac{c_{n,n}^2}{\bar{d}_n^2 s_n^2} + \frac{1}{s_n} \right] C_{n-1} C_{n-1}^T \quad (\text{A-17})$$

Reescrevendo a equação A-5 como em A-18

$$s(t) = s_n = \lambda(t) + \sum_{j=1}^n f_j h_j \quad (\text{A-18})$$

vê-se que é possível calcular  $s_k$  para  $k = 1, \dots, n$  através da equação A-19

$$s_k = \lambda(t) + \sum_{i=1}^k f_i h_i \quad (\text{A-19})$$

e a equação A-18 será dada recursivamente por A-20.

$$s_n = s_{n-1} + \frac{h_{nn}^2}{d_n} = s_{n-1} + \frac{c_{n,n}^2}{d_n} \quad (\text{A-20})$$

Dividindo a equação A-20 por  $s_{n-1}$  obtem-se a equação A-21:

$$1 / s_{n-1} = 1 / s_n + c_{n,n}^2 / d_n s_n s_{n-1} \quad (\text{A-21})$$

Desenvolvendo-se a equação A-21, obtem-se A-22

$$\frac{1}{s_n} = \frac{1}{s_{n-1}} \left[ 1 - \frac{c_{n,n}^2}{d_n s_n} \right]$$

$$s_{n-1} / s_n = 1 - c_{n,n}^2 / d_n s_n \quad (\text{A-22})$$

Desenvolvendo-se A-14.b obtem-se a equação A-23

$$\frac{1}{d_n} = d_n - c_{n,n}^2 / s_n$$

$$\frac{d_n}{d_n} = 1 - \frac{c_{n,n}^2}{d_n s_n} \quad (\text{A-23})$$

Da comparação entre A-22 e A-23 obtem-se a equação A-24

$$\bar{d}_n = \frac{s_{n-1}}{s_n} d_n \quad (\text{A-24})$$

onde  $\bar{d}_n$  é obtido a partir de  $s_n$ ,  $s_{n-1}$  e  $d_n$ .

Substituindo A-24 em A-14.c tem-se a equação A-25 a seguir

$$\bar{u}_{in} = \frac{-c_{n,n}}{d_n \frac{s_{n-1}}{s_n}} c_{n,i} = -\frac{h_n}{d_n s_{n-1}} h_i = \frac{d_n f_n}{d_n s_{n-1}} h_i$$

$$\bar{u}_{in} = \frac{f_n}{s_{n-1}} h_i \quad \text{para } i = 1, \dots, n-1 \quad (\text{A-25})$$

onde  $\bar{u}_{in}$  é obtido a partir a partir de  $s_{n-1}$ ,  $f$  e  $h$ .

Substituindo a equação A-24 para  $\bar{d}_n$  e a equação A-21 para  $1/s_{n-1}$  em A-17, a matriz  $M_n$  passa a ser dada por A-26

$$M_n = \left[ \begin{array}{c} c_{n,n}^2 \\ \frac{s_{n-1}}{s_n} d_n s_n^2 + \frac{1}{s_n} \end{array} \right] C_{n-1} C_{n-1}^T = \left[ \begin{array}{c} c_{n,n}^2 \\ \frac{s_{n-1}}{s_n} d_n s_n \end{array} + \frac{1}{s_n} \right] C_{n-1} C_{n-1}^T$$

$$M_n = \frac{1}{s_{n-1}} C_{n-1} C_{n-1}^T \quad (\text{A-26})$$

Substituindo a equação A-13 em A-11 obtem-se a equação A-27 abaixo,

$$\sum_{i=1}^n \bar{u}_i \bar{u}_i^T \bar{d}_i = \sum_{i=1}^n d_i e_i e_i^T - \frac{1}{s_n} C_n C_n^T \quad (\text{A-27})$$

que pode ser escrita recursivamente como na equação A-28.

$$\sum_{i=1}^n \bar{u}_i \bar{u}_i^T \bar{d}_i = \sum_{i=1}^{n-1} \bar{d}_i \bar{u}_i \bar{u}_i^T + \bar{d}_n \bar{u}_n \bar{u}_n^T =$$

$$= \sum_{i=1}^{n-1} d_i e_i e_i^T + \left[ d_n e_n e_n^T - \frac{1}{s_n} C_n C_n^T \right] \quad (\text{A-28})$$

O termo entre colchetes na equação A-28 vale, de acordo com a equação A-12 para  $M_n$ ,

$$\left[ d_n e_n e_n^T - \frac{1}{s_n} C_n C_n^T \right] = \bar{d}_n \bar{U}_n \bar{U}_n^T - M_n \quad (\text{A-29})$$

Substituindo A-29 na equação A-28, vê-se que

$$\sum_{i=1}^{n-1} d_i \bar{U}_i \bar{U}_i^T = \sum_{i=1}^{n-1} d_i e_i e_i^T - M_n \quad (\text{A-30})$$

pois

$$\sum_{i=1}^{n-1} d_i \bar{U}_i \bar{U}_i^T + \bar{d}_n \bar{U}_n \bar{U}_n^T = \sum_{i=1}^{n-1} d_i e_i e_i^T + \bar{d}_n \bar{U}_n \bar{U}_n^T - M_n$$

e substituindo o valor de  $M_n$  dado em A-26 obtém-se a equação A-21 abaixo.

$$\sum_{i=1}^{n-1} d_i \bar{U}_i \bar{U}_i^T = \sum_{i=1}^{n-1} d_i e_i e_i^T - \frac{1}{s_{n-1}} C_{n-1} C_{n-1}^T \quad (\text{A-31})$$

A expressão A-31 tem exatamente a mesma forma que a equação A-27, desde que  $\bar{U}_n$  e  $\bar{d}_n$  sejam dados respectivamente por A-25 e A-24, exceto que a ordem diminui de  $n$  para  $n-1$ . Os mesmos procedimentos podem ser utilizados para calcular  $d_{n-1}$ ,  $U_{n-1}$ , etc.

O algoritmo para encontrar  $U(t)$  e  $D(t)$  recursivamente é assim obtido através da resolução das equações A-25 e A-27 e das atualizações dadas em A-10. Este algoritmo é conhecido na literatura como algoritmo de fatorização de Bierman.

Se o ganho dos mínimos quadrado dado por A-32

$$G(t) = \frac{P(t-1)\varphi(t)}{1/\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)} = \frac{P(t-1)\varphi(t)}{s(t)} \quad (\text{A-32})$$

for calculado na forma fatorizada

$$G(t) = \frac{U(t-1)D(t-1)U^T(t-1)\phi(t)}{s(t)} = \frac{U(t-1)h(t)}{s(t)} \quad (A-33)$$

obtem-se o algoritmo dos mínimos quadrados ponderados recursivo fatorizado que está descrito na seção 2.6.1.

## APÊNDICE B

### ALGORITMO COM FATORIZAÇÃO U-D PARA $P(t) = \tilde{P}(t) + Q(t)$

Neste apêndice discute-se a fatorização  $UDU^T$  da matriz  $P(t)$  dos mínimos quadrados quando utiliza-se neste algoritmo, a estratégia de busca aleatória para manter a capacidade de rastreamento. Esta fatorização está de acordo com [13] e [21].

A matriz  $P(t)$  é calculada através da equação B-1,

$$P(t) = \tilde{P}(t) + Q(t) \quad (B-1)$$

onde  $\tilde{P}(t)$  é calculada através da equação A-1 do apêndice A.

O valor de  $\tilde{P}(t) = \tilde{U}(t) \tilde{D}(t) \tilde{U}(t)^T$  é obtido usando o algoritmo do apêndice A. Para cálculo de  $Q(t)$ , deve-se realizar a seguinte fatorização

$$Q(t) = V(t) V(t)^T \quad (B-2)$$

onde  $V(t)$  é uma matriz cujo posto é o número de elementos não nulos ( $s$ ) da diagonal de  $Q(t)$  e  $V(t) \in \mathbb{R}^{n \times s}$ .

Formando a matriz  $W \in \mathbb{R}^{n+s \times n}$  como

$$W = [ W_1 \dots W_i \dots W_n ] = ( \tilde{U}(t) \quad V(t) )$$

e a matriz

$$D = \begin{bmatrix} D(t) & 0 \\ 0 & I_s \end{bmatrix}$$

onde  $I_s$  é a matriz identidade de ordem  $s$ , a equação B-1 pode ser reescrita como

$$P(t) = \tilde{U}(t) \tilde{D}(t) \tilde{U}^T(t) + V(t) V(t)^T = WDW^T \quad (B-3)$$

Para obter a fatorização U-D de P(t) é preciso que

$$WDW^T = U(t) D(t) U^T(t) \quad (B-4)$$

onde a matriz D(t) seja diagonal e a matriz U(t) seja triangular superior com todos os elementos da diagonal unitários.

Utilizando um procedimento de ortogonalização de Gram-Schmidt calcula-se os vetores  $W_1, \dots, W_n$  através do produto escalar  $\langle W_i, W_j \rangle = W_i^T D W_j$  e obtem-se a matriz  $\tilde{W} = T W$  formada por vetores coluna ortogonais (D-normal), onde T é uma matriz triangular.

Aplicando a transformação T à equação B-4, tem-se B-5.

$$\begin{aligned} T W D W^T T^T &= T U D U^T T^T \\ \tilde{W} \tilde{W}^T &= [T U] D [T U]^T \end{aligned} \quad (B-5)$$

O produto  $\tilde{W} \tilde{W}^T$  é diagonal e portanto escolhe-se na equação B-5

$$D(t) = \tilde{W} D \tilde{W}^T \quad (B-6)$$

e em consequência U(t) passa a ser dado pela equação B-7.

$$U(t) = T^{-1} \quad (B-7)$$

As equações B-6 e B-7 correspondem a decomposição desejada em B-4.

O algoritmo dos mínimos quadrados fatorizado com busca aleatória é obtido usando o algoritmo fatorizado de Bierman, para cálculo de  $\tilde{U}(t)$ ,  $\tilde{D}(t)$ ,  $G(t)$  e  $\hat{\theta}(t)$ , e as equações B-6 e B-7 para atualização de U(t) e D(t). A implementação deste algoritmo está descrita na seção 2.6.2.

## REFERÊNCIAS BIBLIOGRÁFICAS

[01] - Amaral, W.C. - "Identificação de Sistemas Multivariáveis"  
Tese de Doutorado, FEC - UNICAMP, 1981.

[02] - Aström, K.J. - Introduction To Stochastic Control Theory;  
Academic Press, New York, 1970.

[03] - Aström, K.J.; Anton, J.J., Arzen, K.-E. - "Expert Control"  
Automatica, 22, 277-286, 1986.

[04] - Aström, K.J. - "Adaptive Feedback Control", Proceedings of  
the IEEE, vol 75, n° 2, feb. 1987.

[05] - Bierman, G.J. - Factorization Methods For Discrete  
Sequential Estimation, Academic Press, New York, 1977.

[06] - Eykoff, P. - System Identification Parameter and State  
Estimation, London, Wiley, 1974.

[07] - Favier, G., Rougerie C., Bariani J.P., Amaral W., Gimeno  
L., Arruda, L.V.R. - "A Comparison of Fault Detection Methods and  
Adaptive Identification Algorithms", 1988-IFAC on  
Identification, China, 1988.

[08] - Gomide, F., Amaral W., Arruda L.V.R., Favier G., Bárbara  
A.S., Fontanini, W. - "Expert System Identification: The  
Supervisory Approach", 1988-IFAC on Identification, China, 1988.

[09] - Goodwin, G.C.; Sin, K.S. - Adaptive Filtering  
Prediction and Control, Prentice-hall, 1984.

[10] - Iserman, R. - "Practical Aspects of Process  
Identification", Automatica, 16, 575-587, 1980.

- [11] - Iserman, R. - "Process Fault Detection Based on Modeling and Estimation Methods -A Survey", Automatica, 20, 387-404, 1984.
- [12] - Iserman, R.; Lachmann, K.-H. - "Parameter - Adaptive Control with Configuration Aids and Supervision Functions", Automatica, 21, n° 6, 625-638, 1985.
- [13] - Ljung, L.; Söderström, T. - Theory and Practice of Recursive Identification, MIT Press, 1983.
- [14] - Nilson, N. - Principles of Artificial Intelligence, Toga Publishing Co., 1980.
- [15] - Payne, R.L.; Goodwin, G.C. - Dynamic System Identification: Experiment Design and Data Analysis. Academic Press, 1977.
- [16] - Redford, S.J. - "Development of an Expert System for Configuring Self-Tuning Controllers", Msc. Thesis, University of Manchester, 1985.
- [17] - Rich, E. - Artificial Intelligence, Mac Grow-Hill, 1983.
- [18] - Samaan, R.S. - "Station de Travail et Systèmes Experts en Supervision ", Rapport de D.E.A d'Automatique, Grenoble, 1984.
- [19] - Seborg, D.E.; Edgar, T.F.; Sham, S.L. - "Adaptive Control Strategies for Process Control: A Survey", AIChE Journal, vol 32, n° 6, 881-913, june 1986.
- [20] - Shirley, R.S. - "Some Lessons Learned Using Expert Systems for Process Control", IEEE Control Systems Magazine, 11-15, December, 1987.
- [21] - Thornton, C.L.; Bierman, G.J. - "UDU<sup>T</sup> Covariance Factorization for Kalman Filtering", In C.T. Leondes ed.: Control and Dynamics Systems, vol 16, 177-248, Academic Press, 1980.

[22] - Turbo Pascal, version 3.0 - Reference Manual, Borland International Inc., 1985.

[23] - Valdman, B. - "Etude et Developpement de Systemes de Commande pour une Plateforme de Visée en Poursuite de Cible", Thèse Docteur - Ingenieur, Université de Nice, 1985.

[24] - Vax/Vms - Primer Digital Software, Digital Equipment Corporation, 1982.

[25] - Waterman, D.A. - A Guide to Expert Systems, Addison - Wesley, 1986.

[26] - Wellstead, P.; Zanker, P. - "Techniques of Self-Tuning", Optimal Control Applications & Methods, vol 3, 305-322, 1982.